# Data-Centric Debugging: mitigating model failures via targeted image retrieval

Sahil Singla*
Google Research
sasingla@google.com

Atoosa Malemir Chegini, Mazda Moayeri, Soheil Feizi
University of Maryland
{atoocheg,mmoayeri,sfeizi}@umd.edu


Figure 1. We introduce a method for debugging model failures by discovering "visually similar" images from the web. Left: sample images on which 20 ImageNet trained models with near state-of-the-art accuracy make incorrect predictions (label below each image). Our proposed framework can fix such failure modes. Right: "visually similar" images from the web.

## Abstract

*Deep neural networks can be unreliable in the real world when the training set does not adequately cover all the settings where they are deployed. Focusing on image classification, we consider the setting where we have an error distribution $\mathcal{E}$ representing a deployment scenario where the model fails. We have access to a small set of samples $\mathcal{E}_{sample}$ from $\mathcal{E}$ and it can be expensive to obtain additional samples. In the traditional model development framework, mitigating failures of the model in $\mathcal{E}$ can be challenging and is often done in an ad hoc manner. In this paper, we propose a general methodology for model debugging that can systemically improve model performance on $\mathcal{E}$ while maintaining its performance on the original test set. Our key assumption is that we have access to a large pool of weakly (noisily) labeled data $\mathcal{F}$. However, naively adding $\mathcal{F}$ to the training would hurt model performance due to the large extent of label noise. Our Data-Centric Debugging (DCD) framework carefully creates a debug-train set by selecting images from $\mathcal{F}$ that are visually similar to the images in $\mathcal{E}_{sample}$. To do this, we use the $\ell_2$ distance in the feature space (penultimate layer activations) of various models including ResNet, Robust ResNet and DINO where we observe DINO ViTs are significantly better at discovering similar images compared to Resnets. Compared to the baselines that maintain model performance on the test set, we achieve significantly (+9.45%) improved results on the debug-heldout sets.*

*Work carried out while at the University of Maryland

## 1. Introduction

As machine learning systems are increasingly being deployed in the real world, understanding and mitigating their failure modes becomes critical to ensure that models work reliably in different deployment settings. For example, in medical applications, it is common to train a model using data from a few hospitals, and then deploy it more broadly to hospitals outside the training set [86]. In such cases, we may want to identify the hospital systems on which the model fails and feed more training data from those systems into the model to improve its performance.

Most of the prior works in the literature focus on mitigating a small set of failure modes identified by a human-in-the-loop [45, 60]. This can involve collecting new datasets

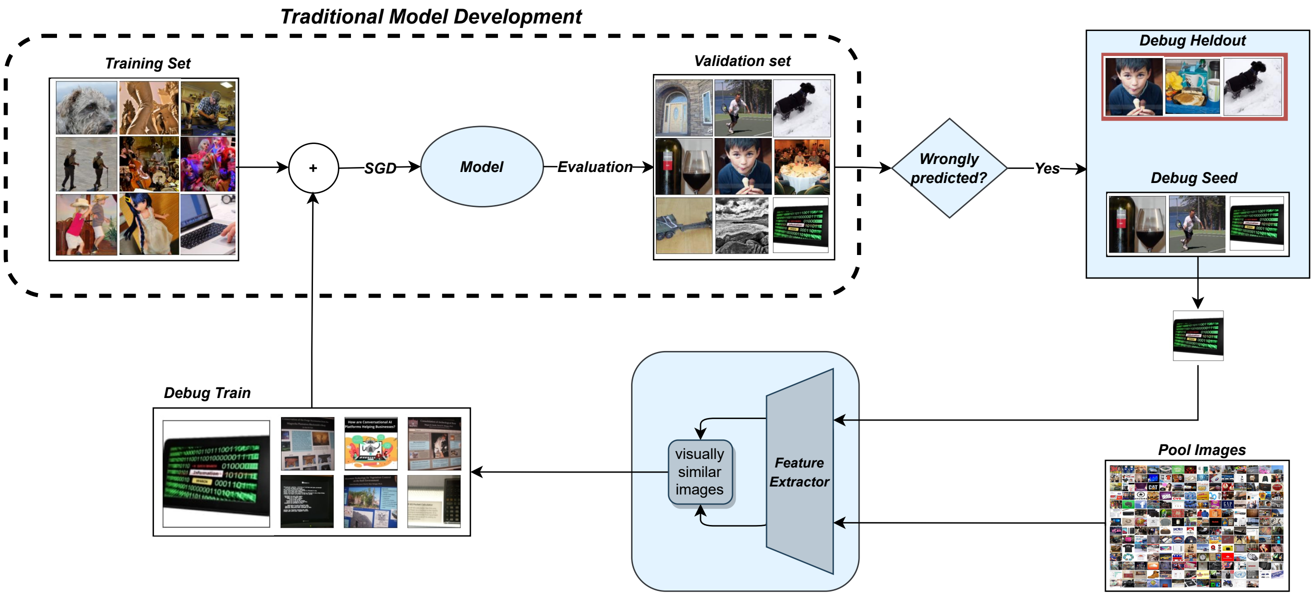



Figure 2. Our framework (Data-Centric Debugging) for model debugging. We want to improve model performance on an error distribution $\mathcal{E}$ (while maintaining accuracy on the test sets) using $\mathcal{E}_{sample}$ (from $\mathcal{E}$) and a weakly labeled dataset $\mathcal{F}$. We divide the set $\mathcal{E}_{sample}$ into two disjoint sets: $\mathcal{E}_{seed}$ and $\mathcal{E}_{heldout}$. Using the samples in $\mathcal{E}_{seed}$, we want to add several "visually similar" images (selected from $\mathcal{F}$) to the training set, such that the model performance improves on $\mathcal{E}_{heldout}$.

with objects in uncommon settings [4, 27–29, 33] (e.g. frog in snow, ship indoors) which can be time consuming and expensive. Moreover, in several cases, humans might not be adequately aware of the undesirable failure modes and even if they are, collecting large number of images in the desired deployment scenarios might not be feasible.

In this work, we focus on the image classification problem $\mathcal{X} \rightarrow \mathcal{Y}$ where the goal is to predict the ground truth label $y \in \mathcal{Y}$ for input $\mathbf{x} \in \mathcal{X}$. In the traditional model development framework, we have a training set, a validation set and a test set. The training set is used to train the model, the validation set is used to evaluate and improve the model performance during development and the test set is used to report a final metric for the model performance.

In this setting, we consider an error distribution $\mathcal{E}$ representing a deployment scenario where a trained model fails i.e. the model makes incorrect predictions on every sample from $\mathcal{E}$. We have access to a small set of images $\mathcal{E}_{sample}$ from $\mathcal{E}$ and it is prohibitively expensive to obtain more samples from $\mathcal{E}$. Our goal is to improve model performance on $\mathcal{E}$ while maintaining performance on the existing test set(s). One naive approach could be to add $\mathcal{E}_{sample}$ to the training set. However, the model might overfit to $\mathcal{E}_{sample}$ and fail to generalize to novel samples from $\mathcal{E}$. Thus, the traditional development framework can be ineffective when sampling a large amount of data from $\mathcal{E}$ is infeasible.

In this work, we propose a new formulation where in addition to the training data and $\mathcal{E}_{sample}$, we have access to a large weakly-labeled (i.e., very noisily labeled) pool of images denoted by $\mathcal{F}$. Here, $\mathcal{F}$ could be obtained from Flickr, Commoncrawl [1] or any suitable data source. We collect $\mathcal{F}$ using Flickr search (Section 3.2) and carry out a filtration step to ensure that the images in $\mathcal{F}$ are "significantly different" from the test set (Section 3.3). Because of the noisy labeling, we find that naively adding the complete set $\mathcal{F}$ to the training set can hurt model performance. For example, we observe $-13.67\%$ drop in accuracy (Table 3). Thus, we want to select a few samples from $\mathcal{F}$ *without human supervision* to improve model performance on $\mathcal{E}$.

Intuitively, by selecting several images from $\mathcal{F}$ that are "visually similar" to the images in $\mathcal{E}_{sample}$, we would expect a broader coverage of $\mathcal{E}$ resulting in improved model performance compared to say, only adding $\mathcal{E}_{sample}$. However, because $\mathcal{F}$ can be large, identifying such similar images can be difficult. Moreover, even if we discover the similar images, we may achieve improved results on $\mathcal{E}_{sample}$ due to some patterns specific to the $\mathcal{E}_{sample}$ images. For example, an image in $\mathcal{E}_{sample}$ contains some pattern that is similar to the patterns of some different class and similarity matching may discover images from the other class. As a result, the model may achieve improvements on $\mathcal{E}_{sample}$ and still fail to generalize to new samples from $\mathcal{E}$. Thus, to ensure that a model revision improves performance over the distribution $\mathcal{E}$ as opposed to simply the observed instances

| Debug-Train | Accuracy on different sets | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **incorrectly classified** | | **subset of 160 classes** | | | **all 1000 classes** | | |
| | Seed | Heldout | MFreq | Compl. | INet | MFreq | Compl. | INet |
| original | 0% | 0% | 35.56% | 56.89% | 62.89% | 63.70% | 76.12% | 76.47% |
| semi supervised | 16.78% | 21.97% | 39.62% | 56.29% | 51.22% | 61.30% | 72.73% | 75.25% |
| Complete ($\mathcal{F}$) | 18.53% | 23.09% | 37.56% | 53.84% | 49.22% | 61.70% | 72.93% | 75.07% |
| Random ($\mathcal{F}$) | 16.78% | 20.17% | 41.56% | 58.81% | 63.91% | 63.77% | 75.21% | 76.37% |
| DCD-DINO | 36.28% | **29.62%** | 54.06% | **63.85%** | **64.62%** | 65.28% | **76.42%** | 76.54% |

Table 1. Results using DINO ViT-S/8 in our framework. "INet" denotes ImageNet test set, "MFreq" denotes the ImageNet-V2 [56] MatchedFrequency set, "Compl." (Complement) denotes the set of all ImageNet-V2 images excluding "MFreq".

$\mathcal{E}_{sample}$, a framework for model debugging is required.

To address these challenges, we introduce **Data-Centric Debugging (DCD)**, illustrated in Figure 2: a framework for targeted image retrieval to mitigate failure modes of deep models and faithfully assess model performance on the error distribution. To retrieve visually similar images, we use the $\ell_2$ distance in the feature space (penultimate layer activations) of a deep network. To faithfully evaluate model performance, we divide the set $\mathcal{E}_{sample}$ into two disjoint sets namely, $\mathcal{E}_{seed}$ and $\mathcal{E}_{heldout}$. We refer to them as debug-seed set and debug-heldout sets respectively. We want to use the set $\mathcal{E}_{seed}$ for discovering visually similar images such that the model performance improves on $\mathcal{E}_{heldout}$. That is, we only use $\mathcal{E}_{seed}$ (not $\mathcal{E}_{heldout}$) for visual similarity matching and evaluate model performance on $\mathcal{E}_{heldout}$. Because $\mathcal{E}_{heldout}$ is disjoint from $\mathcal{E}_{seed}$ but from the same distribution $\mathcal{E}$, an improved performance would suggest that the model is not overfitting to the images in $\mathcal{E}_{seed}$ and can generalize to novel samples from $\mathcal{E}$. Thus, model performance on $\mathcal{E}_{heldout}$ is a more faithful evaluation metric for $\mathcal{E}$.

We apply our proposed framework on the ImageNet [16] classification task. We first select 160 ImageNet classes on which 20 highly accurate ImageNet trained models achieve low accuracy (details in Appendix A). From these classes, we select the incorrectly classified samples from the ImageNet-V2 dataset as the $\mathcal{E}_{sample}$ set. Next, we divide $\mathcal{E}_{sample}$ into the $\mathcal{E}_{seed}$ and $\mathcal{E}_{heldout}$ sets (Section 3.4). For an image $\mathbf{x} \in \mathcal{E}_{seed}$ with label $i$, we can either select visually similar images from the subset of $\mathcal{F}$ with weak label $i$ (denoted by $\mathcal{F}(i)$) or from the complete set $\mathcal{F}$ thereby discarding the weak labels. In the latter case, we can assign the label $i$ to selected images. We find that selecting from the complete set often leads to images that are similar to $\mathbf{x}$, but from a different class, thereby contaminating the dataset with wrongly labeled images. Thus, we select similar images from the subset $\mathcal{F}(i)$ (examples in Appendix J).

In Section 4, we experiment with several different models for extracting image embeddings for visual similarity matching namely, Standard Resnet-50, Robust Resnet-50, DINO ViT-S/8 and DINO ViT-S/16 [10]. Our experiments (Table 3) suggest that DINO models are significantly better at discovering visually similar images compared to Resnets.

In Table 1, we compare our method against the "original" model trained using standard ImageNet, "semi-supervised" model trained via semi-supervised learning (on $\mathcal{F}$), "Random ($\mathcal{F}$)": trained using randomly selected images from subsets $\mathcal{F}(i)$, "Complete ($\mathcal{F}$)": trained on the full $\mathcal{F}$ dataset (with class re-weighting so that weights assigned to classes are same as in ImageNet). For our results ("DCD-DINO"), we used DINO ViT S/8 for similarity matching. We observe that our method achieves the best results on Heldout set: (29.62%), significantly outperforming semi-supervised (19.47%), complete (23.07%) and random (20.17%). We also achieve the best results on several 160 class ImageNet subsets. Moreover, "Complete ($\mathcal{F}$)" results in large accuracy drop on INet (160 classes) from 62.89% to 49.22% (-13.67%) whereas with our method, the accuracy improves to 64.62% (+1.73%). These results highlight that our proposed framework is effective in mitigating model failures.

In summary, we make the following contributions:

1. We proposed **DCD**, a framework for mitigating model failures via data-centric debugging. In contrast to the traditional model development using training/validation/test splits, we construct debug seed/train/heldout datasets to systematically improve failure modes of the model.

2. Using the $\ell_2$ distance in the penultimate layer for retrieving visually similar images, we conduct experiments using different models for extracting penultimate layer features. We find that DINO models are significantly better at discovering similar images compared to Resnet-50 models (Results in Figure 3).

3. Using our framework, we achieve 29.62% accuracy on the debug-heldout set, compared to 0%, 21.97%, 23.09%, 20.17% for the baselines. We also achieve significant improvements: 63.85% vs 58.81% for runner-up (+5.04%) on ImageNet-V2 subset ("Compl." in Table 1).

## 2. Notation

Let set $\mathcal{A}$ consist of (image, label) pairs: $(\mathbf{x}, y) \in \mathcal{A}$, and $\mathcal{A}(i)$ denote the images with label $i$:

$$\mathcal{A}(i) = \{\mathbf{x} : (\mathbf{x}, i) \in \mathcal{A}\}$$

Given two sets: $\mathcal{A}$ and $\mathcal{B}$, we use: $(\mathcal{A} - \mathcal{B})(i)$ to denote $\mathcal{A}(i) - \mathcal{B}(i)$. We use $|\mathcal{A}|$ to denote the cardinality (number of elements) of $\mathcal{A}$, $[n]$ to denote the set: $[0, 1, \ldots, n-1]$ and $\|\mathbf{z}\|$ for the $l_2$ norm of vector $\mathbf{z}$. For image $\mathbf{x}$, $\Phi(\mathbf{x})$ denotes the penultimate layer output of the model $\Phi$. We use $\mathcal{Y}$ to denote the set of all 1000 ImageNet classes.

## 3. Framework for model debugging

Consider the image classification problem $\mathcal{X} \rightarrow \mathcal{Y}$ where we want to predict the ground truth label $y \in \mathcal{Y}$ for input $\mathbf{x} \in \mathcal{X}$. Given a trained model, we have an *error distribution* $\mathcal{E}$ of incorrectly classified images i.e. every image sampled from $\mathcal{E}$ is misclassified by the model. We have access to a set of samples $\mathcal{E}_{sample}$ from $\mathcal{E}$ and it is very expensive to draw more samples. Here, $\mathcal{E}$ represents the deployment scenario where we we want to improve model performance. For example, we may be interested in images with people of color, specific gender or distribution shift (e.g., people wearing masks during COVID-19), etc.

We also assume that we have access to a large pool of weakly labeled images (noisy labels) denoted by $\mathcal{F}$. Here, $\mathcal{F}$ can be obtained using any suitable data source depending on the problem. Using $\mathcal{F}$, we want to improve model performance on $\mathcal{E}$ while maintaining on the desired test set(s).

One naive method could be to add the complete set $\mathcal{F}$ to the training set. However, because the labels in $\mathcal{F}$ can be very noisy, this can reduce the quality of the dataset and hurt model performance. Thus, we want to improve performance on $\mathcal{E}$ by selecting new training images from $\mathcal{F}$ while maintaining model performance on the desired test set(s).

Intuitively, we would expect an improved performance on $\mathcal{E}$ by selecting several images from $\mathcal{F}$ that are "visually similar" to $\mathcal{E}_{sample}$ images. However, identifying similar images from a large dataset can be difficult. Moreover, even if we successfully discover such images, we may see an improved performance on $\mathcal{E}_{sample}$ due to some patterns specific to $\mathcal{E}_{sample}$ images that may fail to generalize to new images from $\mathcal{E}$. For example, the pattern in one image may be a strong match for another image from different class. Thus, we want an evaluation procedure that reflects

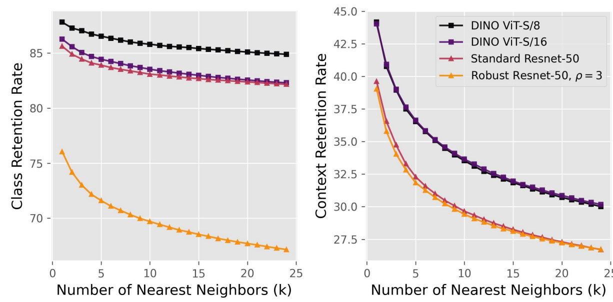



Figure 3. Class and context retention for nearest-neigbhor retrieval on the FOCUS dataset, using different models $\Phi$. DINO ViTs are superior for both class and context retention.

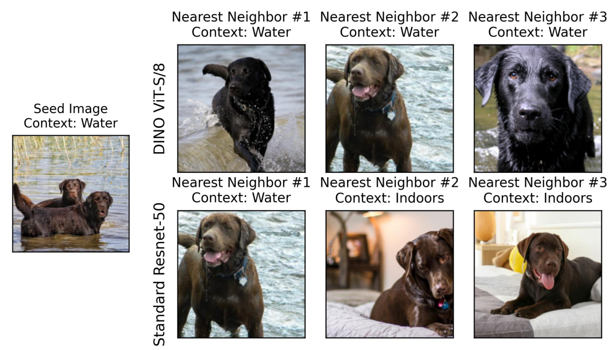



Figure 4. Example where nearest neighbor ($k = 3$) search on FOCUS dataset using DINO ViT-S/8 features retains context, while Standard Resnet-50 does so in $1/3$ cases.

the true model performance on $\mathcal{E}$ because the performance on $\mathcal{E}_{sample}$ may be an overestimate of the same.

To address these challenges, we introduce **Data-Centric Debugging (DCD)**, illustrated in Figure 2: a framework for targeted image retrieval to mitigate failure modes of deep models and faithfully assess model performance on the error distribution. In the next subsections, we discuss the building blocks of our framework.

### 3.1. Models for visual similarity matching

From the set $\mathcal{F}$, we would like to identify images that are "visually similar" to the images in $\mathcal{E}_{sample}$ and add to the training set as we would expect such images to be most effective for improving model performance. In this work, we use the penultimate layer output of a deep model $\Phi$ as the image embedding for visual similarity matching. Given two images $\mathbf{x}, \mathbf{z}$, we use the squared $l_2$ distance ($\|\Phi(\mathbf{x}) - \Phi(\mathbf{z})\|^2$) in this space as the visual similarity distance to discover similar images [10, 22, 57, 59, 74, 76, 77, 89, 90].

We experiment with four pretrained models $\Phi$ for computing these distances: Standard Resnet-50, Robust Resnet-50, DINO ViT-S/16 and DINO ViT-S/8 [10] (Appendix C).

To better inform the choice of the model $\Phi$, we conduct an experiment on the FOCUS dataset [33]. FOCUS consists of common objects in various settings, leading to *two* labels per sample: one label denoting class (bird, plane, etc), and the other denoting context (snow, night, indoors, etc). We obtain features for every sample in FOCUS using various backbones, and then obtain $k = 25$ nearest neighbors per sample in the feature space of each model. We then compute the percent of neighbors amongst the top $k' \leq k$ that retain (i) object class and (ii) image context.

Figure 3 visualizes the results. We observe that the retention rate for class is far higher than for context. Retaining both class and context is important for our use case as the set $\mathcal{E}_{sample}$ may consist of instances of a class in an uncommon context and we would want to select more examples of that object in that same context. We find that DINO vision transformers are superior in *both* class and context retention compared to Resnet models. Specifically, DINO ViT-S/8 achieves the best class and context retention. This unique property makes DINO transformers a prime candidate for visual similarity computation and targeted image retrieval. In Figure 4, we show an example where the 3 nearest neighbors using DINO ViT-S/8 features retain context, but Standard Resnet-50 features do not.

### 3.2. Collecting large pool of images from the web

We want to collect a large pool of images from the web and identify images that are visually similar to the images in $\mathcal{E}_{sample}$. Since collecting a large number of images for all 1000 ImageNet classes can be time consuming, we first select 160 classes (denoted by $\mathcal{T}$) on which 20 highly accurate ImageNet trained models achieve low accuracy (see Appendix A). For each class $i \in \mathcal{T}$, we obtain their synset (set of synonyms). For example, in the synset {'junco', 'snowbird'}, 'junco' and 'snowbird' are synonyms. For each synonym in the synset, we perform a Flickr search and collect the URLs of the first 30,000 images in the search results. After collecting URLs for all classes in $\mathcal{T}$, we remove URLs that were common across multiple classes. This results in a weakly labeled dataset (denoted by $\bar{\mathcal{F}}$) consisting of 952,951 images across 160 classes. Note that $\bar{\mathcal{F}}$ is weakly labeled because all images in the search results may not contain the relevant object in the search term.

### 3.3. Removing overlap with the test sets

Since the model performance on test set can be trivially improved by adding images from the test set to the training set, it is critically important to ensure that the new images added to the training set are "sufficiently different" from the test set images. To this end, we introduce a filtration step based on the criteria that the newly added images should be at least as different from test set images as they are between the ImageNet train/test sets (see Appendix D). Thus for each class $i \in \mathcal{T}$, we first compute a threshold visual similarity distance $\tau(i)$ using the ImageNet dataset.

| **Dataset** | **Class subset** | **# of images** |
|---|---|---|
| Seed<br>Heldout | 160 classes ($\mathcal{T}$) | 1,031<br>719 |
| MFreq | 160 classes ($\mathcal{T}$)<br>1000 classes ($\mathcal{Y}$) | 1,600<br>10,000 |
| Complement | 160 classes ($\mathcal{T}$)<br>1000 classes ($\mathcal{Y}$) | 1,668<br>10,683 |

Table 2. Dataset sizes.

Let $\mathcal{U}$ denote the union of all test sets that we want to evaluate our model on. This includes the seed set, heldout set, and all other test sets. We select the images $\mathbf{x} \in \bar{\mathcal{F}}(i)$ that have visual similarity distance $> \tau(i)$, from all images $\mathbf{z} \in \mathcal{U}(i)$. The new dataset constructed by selecting such images is denoted by $\mathcal{F}$. We add images from $\mathcal{F}$ to the training set (instead of $\bar{\mathcal{F}}$) to prevent images identical (or highly similar) to the test set images from being selected. We note that similar overlap removal procedures have been used in several prior works [37, 42, 52, 55].

### 3.4. Debug-seed and -heldout sets

We divide the set $\mathcal{E}_{sample}$ into two disjoint sets: $\mathcal{E}_{seed}$ and $\mathcal{E}_{heldout}$. Using $\mathcal{E}_{seed}$, we want to add images from $\mathcal{F}$ to the training set that result in improved model performance on $\mathcal{E}_{heldout}$. We stress that $\mathcal{E}_{heldout}$ is never used for image retrieval. The intuition here is that since we are only using $\mathcal{E}_{seed}$ to collect new images and $\mathcal{E}_{heldout}$ is from the same distribution $\mathcal{E}$, an improved performance would suggest that the model is learning relevant concepts (not overfitting to $\mathcal{E}_{seed}$). This leads to the following definitions:

- **Debug-Seed Set**: set of images ($\mathcal{E}_{seed}$) sampled from $\mathcal{E}$ used to collect new training data to improve performance
- **Debug-Heldout Set**: set of images ($\mathcal{E}_{heldout}$) sampled from $\mathcal{E}$ disjoint from $\mathcal{E}_{seed}$ that is used to evaluate performance of model trained on images collected using $\mathcal{E}_{seed}$.

We remark that this is similar to the validation/test setup in model development. We construct the debug-seed and heldout sets by selecting images incorrectly classified by a Standard Resnet-50. We use the 160 class subset ($\mathcal{T}$) for which we obtained Flickr images (Section 3.2).

We use the ImageNet-V2 dataset [56] to sample the seed $\mathcal{E}_{seed}$ and heldout $\mathcal{E}_{heldout}$ sets. ImageNet-V2 consists of three (non-disjoint) sets namely, (a) MatchedFrequency, (b) Threshold0.7, (c) TopImages. We observe that models achieve the lowest accuracy on MatchedFrequency (or "MFreq") set [56]. Thus, we use the incorrectly classified images from this set to construct the set $\mathcal{E}_{seed}$. This ensures that $\mathcal{E}_{seed}$ has a large size. We want the heldout set $\mathcal{E}_{heldout}$ to be disjoint from $\mathcal{E}_{seed}$. Thus, we take the union of all the three sets and remove the "MFreq" images from the union to define the "Complement" set.

We construct the seed set ($\mathcal{E}_{seed}$) by selecting images from "MFreq" with labels in $\mathcal{T}$ that were incorrectly classified by the Resnet-50 model. For the heldout set ($\mathcal{E}_{heldout}$), we select the incorrectly classified images from the "Complement" set, again from the 160 classes in $\mathcal{T}$. The sizes of these datasets are in Table 2.

### 3.5. Debug-train and -validation sets

We use the images in $\mathcal{E}_{seed}$ to select new images from $\mathcal{F}$ and add them to the training set. We may also want to validate that upon training the model on these new images, the performance improves on images visually similar to $\mathcal{E}_{seed}$. Thus, we define the debug-train and debug-validation sets:

- **Debug-Train Set**: set of images selected from $\mathcal{F}$ and added to the training set to improve model performance.
- **Debug-Validation (De-Val) Set**: set of images selected from $\mathcal{F}$ (and disjoint from debug-train set) to validate that the model performance improves on images visually similar to the images in $\mathcal{E}_{seed}$.

For each image $\mathbf{x} \in \mathcal{E}_{seed}(i)$, the de-val should contain a set of few (say $k$) images visually similar to $\mathbf{x}$ with label $i$. We may construct this set by selecting $k$ images with the smallest visual similarity distance to $\mathbf{x}$ from $\mathcal{F}(i)$. However, for two images $\mathbf{x}, \mathbf{z} \in \mathcal{E}_{seed}(i)$, the sets of $k$ images may overlap. Thus, some seed images may have fewer (than $k$) images included and not be well represented. To address this limitation, we use an algorithm (in Appendix E) that removes the selected images on the fly and avoids overlaps. The resulting de-val set is denoted by $\mathcal{V}$. We construct the debug-train set using a similar procedure. Because we want debug-train set to be disjoint from the debug-val set ($\mathcal{V}$), we find visually similar images from the subset: $(\mathcal{F} - \mathcal{V})(i)$. The procedure is same as the de-val set except that we use: $(\mathcal{F} - \mathcal{V})$ instead of $\mathcal{F}$. We first construct the de-val set using $k = 4$ followed by the debug-train set using $k = 46$.

## 4. Experiments

In this Section, we discuss results using the seed and heldout sets (discussed in Section 3.4). We evaluate our proposed method on two criteria: the improvement in accuracy on the heldout-debug set and the accuracy drop on the ImageNet, MFreq and Complement test sets. For each of these test sets, we evaluate on both 160 class subset and all 1000 classes. We use the Resnet-50 architecture for training all models. Each model was trained for 90 epochs over eight GPUs (RTX 2080 Ti). We use the composer library for training all models to reduce training time [72].

### 4.1. Baseline models

We compare against several baseline models:

- **Original**: trained on the ImageNet training set (no additional training images are added)
- **Semi-supervised**: two different models trained using semi-supervised learning. One trained using YFCC-100M [73] as described in [81], other using the $\mathcal{F}$ dataset.
- **Semi-weakly supervised**: trained using IG-1B dataset [42] again using [81]. The key difference between semi-supervised and semi-weakly supervised is that in the latter, model is pre-trained on weakly-supervised data using social media hash tags.
- **Finetuning**: fine-tuned on the $\mathcal{E}_{seed}$ set starting from the "original" model.
- **Only seed**: trained on ImageNet along with only the seed set $\mathcal{E}_{seed}$ (no training images from $\mathcal{F}$).
- **Auto augment**: again trained on ImageNet along with only the seed set $\mathcal{E}_{seed}$ but using the Auto Augment data augmentation strategy [15].
- **Feat match**: again trained on ImageNet along with the seed set $\mathcal{E}_{seed}$ but using feature augmentation as in [39].
- **Complete** ($\mathcal{F}$): trained using the complete dataset $\mathcal{F}$ of size $952,022$ (Section 3.3). Since this leads to a disproportionately large number of images from the 160 classes ($\mathcal{T}$) in $\mathcal{F}$, we assign weights $(w(i))$ to these classes so that the total weight for each class is the same as in ImageNet:

$$\forall\, i \in \mathcal{T}, \;\; w(i) \times |\mathcal{I}_{train}(i) \cup \mathcal{F}(i)| = |\mathcal{I}_{train}(i)|$$

- **Random** ($\mathcal{F}$): For each class $i \in \mathcal{T}$, we randomly select $46 \times |\mathcal{E}_{seed}(i)|$ images (without replacement) from $\mathcal{F}(i)$. Note that this set has the same size as our debug-train set. This ensures a fair comparison across models. From Table 2, the dataset has size $1031 \times 46 = 47426$.

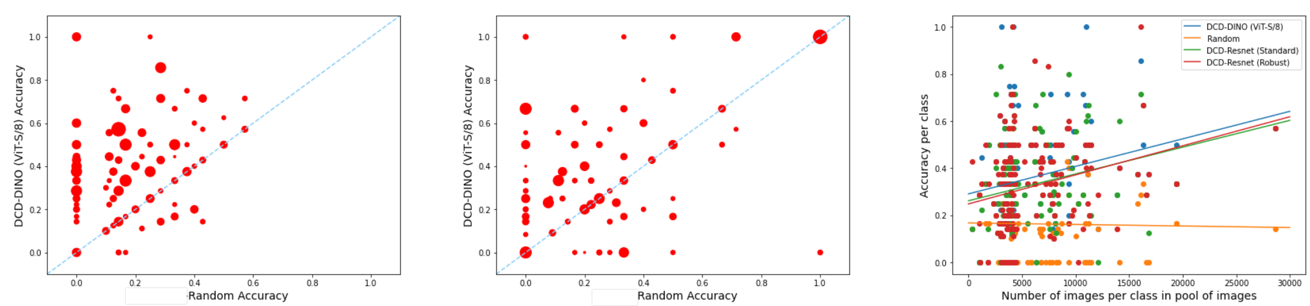
### 4.2. Table details

In Table 3, we show results on different test sets namely,
**MFreq**: only the ImageNet-V2 MatchedFrequency set
**Complement**: all ImageNet-V2 images except Mfreq
**INet**: standard ImageNet test set.
Details for "MFreq" and "Complement" are in Section 3.4.

In "Accuracy on different sets", we show the model accuracy on various subsets of the test sets. In "160 classes", we show the accuracy on images from classes in $\mathcal{T}$ and "1000 classes": from all 1000 classes in ImageNet. In "incorrectly

| Debug-Train method | Accuracy on different sets | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | **incorrectly classified** | | **subset of 160 classes** | | | **all 1000 classes** | | |
| | Seed | Heldout | MFreq | Compl. | INet | MFreq | Compl. | INet |
| original | 0% | 0% | 35.56% | 56.89% | 62.89% | 63.70% | 76.12% | 76.47% |
| semi-supervised (YFCC) | 12.51% | 19.47% | 36.13% | 54.08% | 60.09% | 63.65% | 74.28% | 74.44% |
| semi-supervised ($\mathcal{F}$) | 16.78% | 21.97% | 39.62% | 56.29% | 51.22% | 61.30% | 72.73% | 75.25% |
| semi-weakly supervised | 17.46% | 24.20% | 40.88% | 58.51% | 63.96% | **66.99%** | **77.50%** | **77.18%** |
| finetuning | 32.61% | 24.05% | 47.55% | 56.63% | 59.66% | 62.85% | 71.46% | 75.51% |
| only seed | **44.42%** | 15.58% | 59.31% | 56.47% | 63.78% | 68.05% | 76.28% | 76.86% |
| auto augment | 34.04% | 17.80% | 52.12% | 56.65% | 63.80% | 67.56% | 76.92% | 76.79% |
| feat match | 15.23% | 21.00% | 24.19% | 33.75% | 35.64% | - | - | - |
| Complete ($\mathcal{F}$) | 18.53% | 23.09% | 37.56% | 53.84% | 49.22% | 61.70% | 72.93% | 75.07% |
| Random ($\mathcal{F}$) | 16.78% | 20.17% | 41.56% | 58.81% | 63.91% | 63.77% | 75.21% | 76.37% |
| DCD-DINO (ViT-S/8) | 36.28% | **29.62%** | 54.06% | **63.85%** | **64.62%** | 65.28% | 76.42% | 76.54% |
| DCD-DINO (ViT-S/16) | 36.76% | 26.98% | **55.00%** | 62.83% | 64.11% | 65.69% | 75.38% | 76.41% |
| DCD-Resnet (Standard) | 32.39% | 28.09% | 51.75% | 63.31% | 64.57% | 65.62% | 75.96% | **76.70%** |
| DCD-Resnet (Robust) | 33.07% | 26.84% | 51.5% | 62.29% | 62.27% | 65.04% | 75.70% | 76.50% |

Table 3. Results comparing DCD variants. "INet" denotes the ImageNet test set, "MFreq" denotes the ImageNet-V2 [56] MatchedFrequency set, "Compl." denotes the Complement set (all ImageNet-V2 images excluding "MFreq").




(a) accuracy for 160 classes for DCD-DINO (Y-axis) and Random ($\mathcal{F}$) (X-axis) on seed

(b) accuracy for 160 classes for DCD-DINO (Y-) and Random ($\mathcal{F}$) (X-) on heldout

(c) heldout set accuracy for each class $i \in \mathcal{T}$ (Y-axis) as $|\mathcal{F}(i)|$ increases (X-axis)

Figure 5. In (a) and (b), we plot DCD-DINO accuracy (Y-axis) and Random ($\mathcal{F}$) (X-axis) for all 160 classes. The size of each dot is proportional to the number of images in $\mathcal{F}$ for that class ($|\mathcal{F}(i)|$) and dashed line represents $Y = X$. In (c), we plot the accuracy per class on the heldout set as $|\mathcal{F}(i)|$ increases. We also show the corresponding linear regression lines.

classified", accuracy on images (from 160 class subset) that are incorrectly classified by the "original" model.

In the last four rows, "Debug-Train method" denotes the model $\Phi$ used for computing visual similarity distances. DCD-DINO (ViT-S/8 and ViT-S/16) denote the models trained using DINO ViT-S/8 and ViT-S/16 features. DCD-Resnet (Standard and Robust) denote the models trained using Standard and Robust Resnet-50 features.

### 4.3. Discussion

**Adding the complete set**: While one may believe that noisy data is better than no data, we observe that naively adding noisy data ($\mathcal{F}$) has diminishing returns. In "Complete ($\mathcal{F}$)", 952,022 extra images are added while in "Random ($\mathcal{F}$)" only 47,426. Even though we add 20 $\times$ more images in "Complete", accuracy on seed, heldout are only marginally better. In fact, under other metrics, such as "160 classes" (INet), accuracy for "Complete ($\mathcal{F}$)" is 49.22% significantly below "original" $62.89\%$ ($-13.67\%$) and "Random ($\mathcal{F}$)" $63.91\%$ ($-14.69\%$). This suggests that naively adding large amounts of noisy data can hurt model performance.

**Comparing models**: We want to compare the quality of visual similarity matching using different models. We observe that similarity matching using both DINO models achieves significantly higher accuracy on "Seed" compared to using Resnet-50: DCD-DINO (ViT-S/8) achieves $36.28\%$ compared to $33.07\%$ for Robust Resnet-50. This provides evidence that DINO models are better for similarity matching. Similar trend is also observed for the Heldout set. However, between Standard and Robust Resnet-50, the results on "Seed" are comparable suggesting that adversarial robustness is not critical for similarity matching.

**Comparison with semi-(weakly) supervised methods**: We observe that DCD-DINO (ViT-S/8) significantly outperforms semi-supervised and semi-weakly supervised methods for both the incorrectly classified and subset of 160 classes benchmarks. The performance of semi-weakly supervised methods on all 1000 classes is only marginally better. Since the semi-weakly supervised model was trained using 1 billion additional images spanning all 1000 classes while DCD models are trained using 47426 additional images spanning only 160 classes, these results suggest that using DCD for all 1000 ImageNet classes can further boost model performance on all 1000 classes benchmarks.

**Comparing Complete, Random and DCD**: We observe that DINO ViT-S/8 achieves significantly improved results compared to both "Complete ($\mathcal{F}$)" and "Random ($\mathcal{F}$)". On "Heldout", we achieve 29.62% compared to 23.09% for the next best i.e. gain of 6.53%. On the "Compl." set (160 classes), we achieve 63.85%: gain of 5.04% compared to 58.81% for the next best model. On the 1000 classes sets, we achieve slightly improved results on all sets: 76.42% on "Complement", compared to 76.12% (+0.3%). Similarly on INet, we achieve 76.54% similar to 76.47%. Model performance is maintained on all test sets.

**Comparing only seed and DCD**: We observe that although adding only the seed set and training from scratch ("only seed") achieves high accuracy on Seed (as expected when training on the same data), DCD-DINO (ViT-S/8, ViT-S/16) achieves significantly higher performance on Heldout and Compl. sets (by a margin of at least 5%). DCD also outperforms other baselines such as "finetuning" "auto augment", "feat match" significantly. These results provide strong evidence that adding images to the training set that are visually similar to the seed set can significantly improve model performance on failure mode distributions.

**Accuracy on different classes**: We now compare the seed and heldout accuracy for different classes $i \in \mathcal{T}$ as $|\mathcal{F}(i)|$ varies. In both Figures 5a and 5b, we observe that DCD-DINO achieves better accuracy on most classes comparing to Random ($\mathcal{F}$) (as most points lie above the dashed y=x line). Also, DCD-DINO achieves better accuracy for classes with larger amount of data (larger red dots). In Figure 5c, we compare accuracy per class as $|\mathcal{F}(i)|$ increases (x-axis) for four different models: DCD-Resnet (Standard and Robust), DCD-DINO (ViT-S/8) and Random ($\mathcal{F}$). We see that as $|\mathcal{F}(i)|$ increases, we achieve better accuracy (according to the linear regression lines) for all methods except Random ($\mathcal{F}$). For Random ($\mathcal{F}$), there is a slight accuracy reduction for large $|\mathcal{F}(i)|$. Our results provide evidence that by obtaining large amounts of weakly-labeled data and adding selected images to the training sets, we can achieve significantly improved results.

**Possibility of insufficient/low quality Flickr results**: We emphasize that although the results presented in this work use Flickr search for creating the pool $\bar{\mathcal{F}}$, such a pool can be obtained from any data source. For example: for patient MRI scans, the data could be obtained offline from a network of hospitals. Weak labels can be assigned based on the patient information or top-k predictions of some pre-trained model. Moreover, requirement of high quality training data is central to all deep learning algorithms and reducing such dependencies remains an active area of research.

## 5. Related work

**Dataset Design and Data valuation**: Previous works [7, 13, 17, 18, 24–26, 38, 54, 55] use gigantic amounts of training data to achieve high performance. However, recent work [49, 61, 82] suggests that adding small amounts of carefully selected training data may be more effective for improving model performance. Some recent works [21, 80, 84] focus on data valuation and seek to quantify the contribution of an individual datapoint to the overall model performance.

**Debugging and Explainability**: Existing works on explainability of deep networks focus on inspecting the decisions for a single image [2, 5, 11, 12, 19, 23, 30–32, 36, 43, 44, 47, 48, 51, 53, 58, 62, 65, 68–71, 75, 83, 85, 87, 91, 92] or identifying failure modes across a large set of images [14, 40, 50, 67, 78, 79, 88]. Another class of works focus on making edits to the model to modify its predictions [9, 45, 60] or introducing datasets to stress test model performance on images with the main objects in uncommon or rare settings [4, 29, 33, 46, 66].

**Visual Similarity metrics**: Several visual similarity metrics have been proposed [10, 22, 57, 59, 74, 76, 77, 89, 90] using the $l_2$ distance between features of a deep network.

**Semi-supervised and Active learning**: Semi-supervised methods [6, 8, 41, 61] aim to improve model's performance using unlabeled or weakly labeled data. Active learning [3, 34, 35, 63, 64] identifies data subsets from a large unlabeled corpus for annotation, often by humans, that can lead to high performance gains.

## References


[1] Common crawl. https://commoncrawl.org/. Accessed: 2022-05-18. 2

[2] Julius Adebayo, J. Gilmer, M. Muelly, Ian J. Goodfellow, M. Hardt, and Been Kim. Sanity checks for saliency maps. In *NeurIPS*, 2018. 8

[3] Jordan T. Ash, Chicheng Zhang, Akshay Krishnamurthy, John Langford, and Alekh Agarwal. Deep batch active learning by diverse, uncertain gradient lower bounds. In *International Conference on Learning Representations*, 2020. 8

[4] Andrei Barbu, David Mayo, Julian Alverio, William Luo, Christopher Wang, Dan Gutfreund, Josh Tenenbaum, and Boris Katz. Objectnet: A large-scale bias-controlled dataset for pushing the limits of object recognition models. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. 2, 8

[5] David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 117(48):30071–30078, 2020. 8

[6] David Berthelot, Nicholas Carlini, Ian Goodfellow, Nicolas Papernot, Avital Oliver, and Colin Raffel. Mixmatch: A holistic approach to semi-supervised learning. In *NeurIPS*, 2019. 8

[7] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020. 8

[8] Kaidi Cao, Colin Wei, Adrien Gaidon, Nikos Arechiga, and Tengyu Ma. Learning imbalanced datasets with label-distribution-aware margin loss. In *Advances in Neural Information Processing Systems*, 2019. 8

[9] Nicola De Cao, Wilker Aziz, and Ivan Titov. Editing factual knowledge in language models. 2021. 8

[10] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 3, 4, 8, 13

[11] Shan Carter, Zan Armstrong, Ludwig Schubert, Ian Johnson, and Chris Olah. Activation atlas. *Distill*, 4(3):e15, 2019. 8

[12] Chun-Hao Chang, Elliot Creager, Anna Goldenberg, and David Duvenaud. Explaining image classifiers by counterfactual generation. In *International Conference on Learning Representations*, 2019. 8

[13] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR, 2020. 8

[14] Yeounoh Chung, Tim Kraska, Neoklis Polyzotis, Ki Hyun Tae, and Steven Euijong Whang. Slice finder: Automated data slicing for model validation. In *2019 IEEE 35th International Conference on Data Engineering (ICDE)*, pages 1550–1553. IEEE, 2019. 8

[15] Ekin D. Cubuk, Barret Zoph, Dandelion Mané, Vijay Vasudevan, and Quoc V. Le. Autoaugment: Learning augmentation strategies from data. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019. 6

[16] J. Deng, Wei Dong, R. Socher, L. Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009. 3

[17] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018. 8

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 8

[19] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016. 8

[20] Logan Engstrom, Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Brandon Tran, and Aleksander Madry. Adversarial robustness as a prior for learned representations, 2019. 13

[21] Amirata Ghorbani and James Zou. Data shapley: Equitable valuation of data for machine learning. In *International Conference on Machine Learning*, pages 2242–2251, 2019. 8

[22] A. Gordo, J. Almazan, J. Revaud, and D. Larlus. End-to-end learning of deep visual representations for image retrieval. *IJCV*, 2017. 4, 8

[23] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *ICML*, 2019. 8

[24] Kaiming He, Xinlei Chen, Saining Xie, Yanghao Li, Piotr Dollár, and Ross Girshick. Masked autoencoders are scalable vision learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 16000–16009, 2022. 8

[25] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 8

[26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 8

[27] D. Hendrycks, S. Basart, N. Mu, S. Kadavath, F. Wang, E. Dorundo, R. Desai, T. Zhu, S. Parajuli, M. Guo, D. Song, J. Steinhardt, and J. Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8320–8329, Los Alamitos, CA, USA, oct 2021. IEEE Computer Society. 2

[28] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 2

[29] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *CVPR*, 2021. 2, 8

[30] Aya Abdelsalam Ismail, Mohamed K. Gunady, Héctor Corrada Bravo, and Soheil Feizi. Benchmarking deep learning interpretability in time series predictions. In *NeurIPS*, 2020. 8

[31] Aya Abdelsalam Ismail, Mohamed K. Gunady, Luiz Pessoa, Héctor Corrada Bravo, and Soheil Feizi. Input-cell attention reduces vanishing saliency of recurrent neural networks. In *NeurIPS*, 2019. 8

[32] Saachi Jain, Hadi Salman, Eric Wong, Pengchuan Zhang, Vibhav Vineet, Sai Vemprala, and Aleksander Madry. Missingness bias in model debugging. *arXiv preprint arXiv:2204.08945*, 2022. 8

[33] Priyatham Kattakinda and Soheil Feizi. FOCUS: Familiar objects in common and uncommon settings. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 10825–10847. PMLR, 17–23 Jul 2022. 2, 5, 8

[34] Kwanyoung Kim, Dongwon Park, Kwang In Kim, and Se Young Chun. Task-aware variational adversarial active learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8162–8171, 2020. 8

[35] Andreas Kirsch, Joost R. van Amersfoort, and Yarin Gal. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Neural Information Processing Systems*, 2019. 8

[36] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, page 1885–1894. JMLR.org, 2017. 8

[37] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part V*, page 491–507, Berlin, Heidelberg, 2020. Springer-Verlag. 5

[38] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017. 8

[39] Chia-Wen Kuo, Chih-Yao Ma, Jia-Bin Huang, and Zsolt Kira. Featmatch: Feature-based augmentation for semi-supervised learning. In *European Conference on Computer Vision*, pages 479–495. Springer, 2020. 6

[40] Guillaume Leclerc, Hadi Salman, Andrew Ilyas, Sai Vemprala, Logan Engstrom, Vibhav Vineet, Kai Xiao, Pengchuan Zhang, Shibani Santurkar, Greg Yang, et al. 3db: A framework for debugging computer vision models. *arXiv preprint arXiv:2106.03805*, 2021. 8

[41] Junnan Li, Richard Socher, and Steven C.H. Hoi. Dividemix: Learning with noisy labels as semi-supervised learning. In *International Conference on Learning Representations*, 2020. 8

[42] Dhruv Kumar Mahajan, Ross B. Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *ECCV*, 2018. 5, 6

[43] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5188–5196, 2015. 8

[44] Aravindh Mahendran and A. Vedaldi. Visualizing deep convolutional neural networks using natural pre-images. *International Journal of Computer Vision*, 120:233–255, 2016. 8

[45] Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D Manning. Fast model editing at scale. *arXiv preprint arXiv:2110.11309*, 2021. 1, 8

[46] Mazda Moayeri, Sahil Singla, and Soheil Feizi. Hard imagenet: Segmentations for objects with strong spurious cues. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 8

[47] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 427–436, 2015. 8

[48] Anh Mai Nguyen, Jason Yosinski, and Jeff Clune. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. In *ICML Workshop on Visualization for Deep Learning*, 2016. 8

[49] Thao Nguyen, Gabriel Ilharco, Mitchell Wortsman, Sewoong Oh, and Ludwig Schmidt. Quality not quantity: On the interaction between dataset design and robustness of clip. *arXiv preprint arXiv:2208.05516*, 2022. 8

[50] Besmira Nushi, Ece Kamar, and Eric Horvitz. Towards accountable AI: hybrid human-machine analyses for characterizing system failure. In Yiling Chen and Gabriella Kazai, editors, *Proceedings of the Sixth AAAI Conference on Human Computation and Crowdsourcing, HCOMP*, pages 126–135. AAAI Press, 2018. 8

[51] Chris Olah, Arvind Satyanarayan, Ian Johnson, Shan Carter, Ludwig Schubert, Katherine Ye, and Alexander Mordvintsev. The building blocks of interpretability. *Distill*, 2018. https://distill.pub/2018/building-blocks. 8

[52] OpenAI. Gpt-4 technical report. *ArXiv*. 5

[53] Matthew O'Shaughnessy, Gregory Canal, Marissa Connor, Mark Davenport, and Christopher Rozell. Generative causal explanations of black-box classifiers. In *NeurIPS*, 2019. 8

[54] Hieu Pham, Zihang Dai, Golnaz Ghiasi, Hanxiao Liu, Adams Wei Yu, Minh-Thang Luong, Mingxing Tan, and Quoc V Le. Combined scaling for zero-shot transfer learning. *arXiv preprint arXiv:2111.10050*, 2021. 8

[55] 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 *International Conference on Machine Learning*, pages 8748–8763. PMLR, 2021. 5, 8

[56] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do ImageNet classifiers generalize to ImageNet? In *Proceedings of the 36th International Conference on Machine Learning*, 2019. 3, 5, 6, 7, 15, 16

[57] J. Revaud, J. Almazan, R.S. Rezende, and C.R. de Souza. Learning with average precision: Training image retrieval with a listwise loss. In *ICCV*, 2019. 4, 8

[58] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?": Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 1135–1144, New York, NY, USA, 2016. Association for Computing Machinery. 8

[59] Olivier Risser-Maroix, Camille Kurtz, and Nicolas Loménie. Learning an adaptation function to assess image visual similarities. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 2498–2502, 2021. 4, 8

[60] Shibani Santurkar, Dimitris Tsipras, Mahalaxmi Elango, David Bau, Antonio Torralba, and Aleksander Madry. Editing a classifier by rewriting its prediction rules. *Advances in Neural Information Processing Systems*, 34:23359–23373, 2021. 1, 8

[61] Lars Schmarje, Monty Santarossa, Simon-Martin Schröder, Claudius Zelenka, Rainer Kiko, Jenny Stracke, Nina Volkmann, and Reinhard Koch. A data-centric approach for improving ambiguous labels with combined semi-supervised classification and clustering. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2022. 8

[62] R. R. Selvaraju, Abhishek Das, Ramakrishna Vedantam, Michael Cogswell, D. Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128:336–359, 2019. 8

[63] Ozan Sener and Silvio Savarese. Active learning for convolutional neural networks: A core-set approach. In *International Conference on Learning Representations*, 2018. 8

[64] Changjian Shui, Fan Zhou, Christian Gagn'e, and Boyu Wang. Deep active learning: Unified and principled method for query and training. In *International Conference on Artificial Intelligence and Statistics*, 2019. 8

[65] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *Workshop at International Conference on Learning Representations*, 2014. 8

[66] Sahil Singla and Soheil Feizi. Salient imagenet: How to discover spurious features in deep learning? In *International Conference on Learning Representations*, 2022. 8

[67] Sahil Singla, Besmira Nushi, Shital Shah, Ece Kamar, and Eric Horvitz. Understanding failures of deep networks via robust feature extraction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 8

[68] Sahil Singla, Eric Wallace, Shi Feng, and Soheil Feizi. Understanding impacts of high-order loss approximations and features in deep learning interpretation. In *ICML*, 2019. 8

[69] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. In *ICML Workshop on Visualization for Deep Learning*, 2017. 8

[70] Pascal Sturmfels, Scott Lundberg, and Su-In Lee. Visualizing the impact of feature attribution baselines. *Distill*, 5(1):e22, 2020. 8

[71] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, 2017. 8

[72] The Mosaic ML Team. composer. `https://github.com/mosaicml/composer/`, 2021. 6

[73] Bart Thomee, David Shamma, Gerald Friedland, Benjamin Elizalde, Karl Ni, Douglas Poland, Damian Borth, and Li-Jia Li. The new data and new challenges in multimedia research. 03 2015. 6

[74] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. 2016. 4, 8

[75] Sahil Verma, John Dickerson, and Keegan Hines. Counterfactual explanations for machine learning: A review, 2020. 8

[76] 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. 4, 8

[77] Z. Wang, E.P. Simoncelli, and A.C. Bovik. Multiscale structural similarity for image quality assessment. In *The Thirty-Seventh Asilomar Conference on Signals, Systems & Computers, 2003*, volume 2, pages 1398–1402 Vol.2, 2003. 4, 8

[78] Eric Wong, Shibani Santurkar, and Aleksander Madry. Leveraging sparse linear layers for debuggable deep networks. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11205–11216. PMLR, 18–24 Jul 2021. 8

[79] Tongshuang Wu, Marco Tulio Ribeiro, Jeffrey Heer, and Daniel S Weld. Errudite: Scalable, reproducible, and testable error analysis. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 747–763, 2019. 8

[80] Xinlei Xu, Awni Hannun, and Laurens Van Der Maaten. Data appraisal without data sharing. In Gustau Camps-Valls, Francisco J. R. Ruiz, and Isabel Valera, editors, *Proceedings of The 25th International Conference on Artificial Intelligence and Statistics*, volume 151 of *Proceedings of Machine Learning Research*, pages 11422–11437. PMLR, 28–30 Mar 2022. 8

[81] Ismet Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Kumar Mahajan. Billion-scale semi-supervised learning for image classification. *ArXiv*, abs/1905.00546, 2019. 6

[82] Jiachen Yang, Zhuo Zhang, Yicheng Gong, Shukun Ma, Xiaolan Guo, Yue Yang, Shuai Xiao, Jiabao Wen, Yang Li, Xinbo Gao, et al. Do deep neural networks always perform better when eating more data? *arXiv preprint arXiv:2205.15187*, 2022. 8

[83] Chih-Kuan Yeh, Cheng-Yu Hsieh, Arun Sai Suggala, David I. Inouye, and Pradeep D. Ravikumar. On the (in)fidelity and sensitivity of explanations. In *NeurIPS*, 2019. 8

[84] Jinsung Yoon, Sercan Arik, and Tomas Pfister. Data valuation using reinforcement learning. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 10842–10851. PMLR, 13–18 Jul 2020. 8

[85] Jason Yosinski, Jeff Clune, Anh Mai Nguyen, Thomas J. Fuchs, and Hod Lipson. Understanding neural networks through deep visualization. In *ICML Deep Learning Workshop*, 2016. 8

[86] John R. Zech, Marcus A. Badgeley, Manway Liu, Anthony Beardsworth Costa, Joseph J. Titano, and Eric Karl Oermann. Variable generalization performance of a deep learning model to detect pneumonia in chest radiographs: A cross-sectional study. *PLoS Medicine*, 15, 2018. 1

[87] Matthew D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 8

[88] Jiawei Zhang, Yang Wang, Piero Molino, Lezhi Li, and David S Ebert. Manifold: A model-agnostic framework for interpretation and diagnosis of machine learning models. *IEEE transactions on visualization and computer graphics*, 25(1):364–373, 2018. 8

[89] Lin Zhang, Lei Zhang, Xuanqin Mou, and David Zhang. Fsim: A feature similarity index for image quality assessment. *IEEE Transactions on Image Processing*, 20(8):2378–2386, 2011. 4, 8

[90] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018. 4, 8

[91] Bolei Zhou, David Bau, Aude Oliva, and Antonio Torralba. Interpreting deep visual representations via network dissection. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2131–2145, 2018. 8

[92] B. Zhou, A. Khosla, Lapedriza. A., A. Oliva, and A. Torralba. Learning Deep Features for Discriminative Localization. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 8