Re-labeling ImageNet: from Single to Multi-Labels, from Global to Localized Labels

Sangdoo Yun Seong Joon Oh Byeongho Heo Dongyoon Han Junsuk Choe Sanghyuk Chun

NAVER AI Lab

Abstract

*ImageNet has been the most popular image classification benchmark, but it is also the one with a significant level of label noise. Recent studies have shown that many samples contain multiple classes, despite being assumed to be a single-label benchmark. They have thus proposed to turn ImageNet evaluation into a multi-label task, with exhaustive multi-label annotations per image. However, they have not fixed the training set, presumably because of a formidable annotation cost. We argue that the mismatch between single-label annotations and effectively multi-label images is equally, if not more, problematic in the training setup, where random crops are applied. With the single-label annotations, a random crop of an image may contain an entirely different object from the ground truth, introducing noisy or even incorrect supervision during training. We thus re-label the ImageNet training set with multi-labels. We address the annotation cost barrier by letting a strong image classifier, trained on an extra source of data, generate the multi-labels. We utilize the pixel-wise multi-label predictions before the final pooling layer, in order to exploit the additional location-specific supervision signals. Training on the re-labeled samples results in improved model performances across the board. ResNet-50 attains the top-1 accuracy of **78.9%** on ImageNet with our localized multi-labels, which can be further boosted to **80.2%** with the CutMix regularization. We show that the models trained with localized multi-labels also outperforms the baselines on transfer learning to object detection and instance segmentation tasks, and various robustness benchmarks. The re-labeled ImageNet training set, pre-trained weights, and the source code are available at* https://github.com/naver-ai/relabel_imagenet.

1. Introduction

The ImageNet dataset [38] has been at the center of modern advances in computer vision. Since the introduction of ImageNet, image recognition models based on convolutional neural networks have made quantum jumps in performances [27, 41, 15]. Improving the model performance on ImageNet is seen as a litmus test for the general applicability of the model and the transfer learning performances on downstream tasks [25, 57].

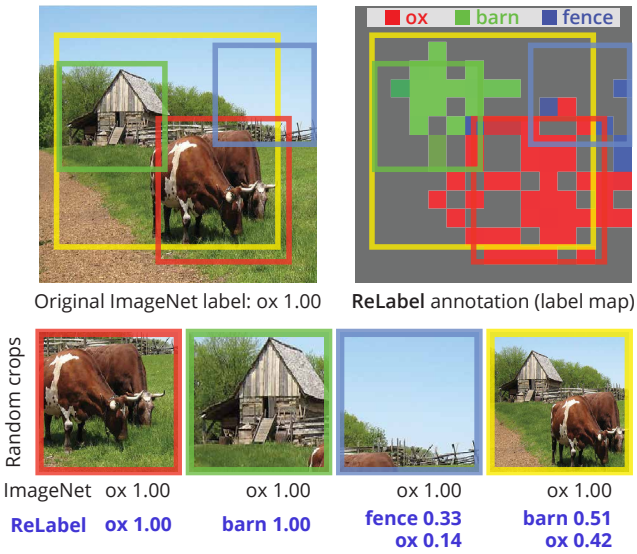


Figure 1. **Re-labeling ImageNet training data.** Original ImageNet annotation is a single label ("ox"), whereas the image contains multiple ImageNet categories ("ox", "barn", and "fence"). Random crops of an image may contain an entirely different object category from the global annotation. Our method (**ReLabel**) generates location-wise multi-labels, resulting in cleaner supervision per random crop.

ImageNet, however, turns out to be noisier than one would expect. Recent studies [42, 49, 2, 39] have shed light on an overlooked problem with ImageNet that a significant portion of the dataset is composed of images with multiple possible labels. This contradicts the underlying assumption that there is only a single object class per image: the evaluation metrics penalize any prediction beyond the single ground-truth class. Thus, researchers have refined the

ImageNet validation samples with multi-labeling policy using human annotators [2, 39], and proposed new multi-label evaluation metrics. Under these new evaluation schemes, recent state-of-the-art models [53, 48] that seem to have surpassed the human level of recognition have been found to fall short of the human performance level.

The mismatch between the multiplicity of object classes per image and the assignment of single labels results in problems not only for evaluation, but also for training: the supervision becomes noisy. The widespread adoption of *random crop* augmentation [44] aggravates the problem. A random crop of an image may contain an entirely different object from the original single label, introducing potentially wrong supervision signals during training, as in Figure 1.

The random crop augmentation makes supervision noisy not only for images with multiple classes. Even for images with a single class, the random crop often contains no foreground object. It is estimated that, under the standard training setup[1], 8% of the random crops have no overlap with the ground truths. Only 23.5% of the random crops have the intersection-over-union (IoU) measure greater than 50% with the ground truth boxes (see Figure 2). Training a model on ImageNet inevitably involves a lot of noisy supervision.

Ideally, for each training image, we want a human annotation telling the model (1) the full set of classes present (multi-label) and (2) where each object is located (localized label). One such format would be a dense pixel labeling $L \in \{0,1\}^{H\times W\times C}$ where C is the number of classes, as done for semantic segmentation ground truths. However, it is hardly scalable to collect even just the multi-label annotations for the 1.28 million ImageNet training samples. It took more than three months for five human experts (authors of [39]) to label mere 2,000 images.

In this paper, we propose a re-labeling strategy, **ReLabel**, to obtain pixel-wise labeling $L \in \mathbb{R}^{H\times W\times C}$, which are both multi-labels and localized labels, on the ImageNet training set. We use strong classifiers trained on external training data to generate those labels. The predictions before the final pooling layer have been used. We also contribute a novel training scheme, **LabelPooling**, for training classifiers based on the dense labels. For each random crop sample, we compute the multi-label ground truth by pooling the label scores from the crop region. ReLabel incurs only a one-time cost for generating the label maps per dataset, unlike *e.g.* Knowledge Distillation [20] which involves one forward pass per training iteration to generate the supervision. Our LabelPooling supervision adds only a small amount of computational cost on the usual single-label cross-entropy supervision.

We present an extensive set of evaluations for various model architectures trained with ReLabel on multiple datasets and tasks. On ImageNet classification, training

[1]A random crop is sampled from 8% to 100% of the entire image area.

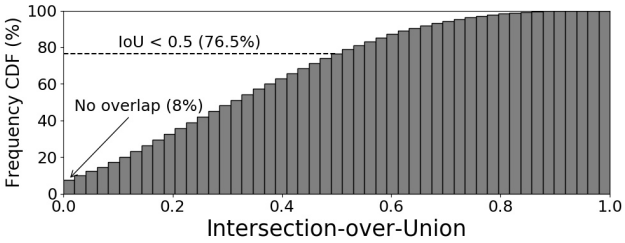


Figure 2. **Cumulative distribution of Intersection-over-Union (IoU)** between the random crops and ground-truth bounding boxes. We sample 100 random crops per image on the ImageNet validation set (50K images).

ResNet-50 with ImageNet ReLabel has achieved a top-1 accuracy of **78.9%**, a **+1.4 pp** gain over the baseline model trained with the original labels. The accuracy of ResNet-50 reaches **80.2%** by employing the CutMix regularization on top, a new state-of-the-art performance on ImageNet to the best of our knowledge. Models trained with ReLabel have also consistently improved accuracies on ImageNet multi-label evaluation metrics proposed by [2, 39]. ReLabel and LabelPooling result in consistent improvements for transfer learning experiments, including the object detection and instance segmentation tasks on COCO and fine-grained classifications tasks. We further test LabelPooling on the multi-label classification task on COCO. Finally, we show that models trained with ReLabel are more resilient to test-time perturbations, as will be verified through experiments on several robustness benchmarks.

2. Related Works

We start this section by introducing prior works discussing the issues with ImageNet labels. We then discuss a few other research areas that share similarities with our approach. We describe the key differences from our approach.

Labeling issues in ImageNet. ImageNet [38] has effectively served as the standard benchmark for the image classifiers: "methods live or die by their performance on this benchmark", as argued by Shankar *et al*. [39]. The reliability of the benchmark itself has thus come to be the subject of careful research and analysis. As with many other datasets, ImageNet contains much label noise [50, 36]. One of the most persistent and systematic types of label error on ImageNet is the erroneous single labels [42, 39, 49, 2], referring to the cases where only one out of multiple present categories is annotated. Such errors are prevalent, as ImageNet contains many images with multiple classes. Shankar *et al*. [39] and Beyer *et al*. [2] have identified three subcategories for the erroneous single labels: (1) an image has multiple object classes, (2) there exist multiple labels that are synonymous or hierarchically including the other, and (3) inherent ambiguity in an image makes multiple labels plau-

	multi-label	local label	efficient
Original ImageNet training	✘	✘	✔
Knowledge distillation	✔	✔	✘
ReLabel & LabelPooling (ours)	✔	✔	✔

Table 1. **What are the differences?** Comparison among the training options on ImageNet.

sible. Those studies have refined the validation set labels into multi-labels to establish an truthful and fair evaluation of models on effectively multi-label images. The focus of [39], however, has been only the validation, not training. [2] has introduced a clean-up scheme to remove training samples with potentially erroneous labels by validating them with predictions from a strong classifier. Our work focuses on the clean-up strategy for the ImageNet training labels. Like [2], we utilize strong classifiers to clean up the training labels. Unlike [2], we *correct* the wrong labels, not *remove*. Our labels are also given per region. In our experiments, our method shows improved results compared to [2].

Knowledge distillation. Knowledge distillation (KD) [20] also utilizes machine supervisions generated by the "teacher" network. Studies on KD have enriched and diversified the options for the teacher, such as feature map distillation [58, 19, 18], relation-based distillation [34, 47], ensemble distillation [40, 61], or iterative self-distillation [9, 55, 53]. While those studies pursue stronger forms of supervision, none of them have considered a strong, state-of-the-art network as a teacher because it makes the KD supervision far heavier and impractical. With the random crop augmentation in place, every training iteration would involve a forward pass through the strong yet heavy teacher. Ours is similar in that the model is trained with machine supervision, but is more efficient[2]. LabelPooling supervises a network with pre-computed label maps, rather than generating the label on the fly through the teacher for every random crop during training. We present the key advantages of ours against KD in Table 1.

Training tricks for ImageNet. Data augmentation is a simple yet powerful strategy for ImageNet training. The standard augmentation setting includes random cropping, flipping, and color jittering, as used in [12, 10, 21, 44, 57, 48]. In particular, the random crop augmentation, which crops random coordinates in an image and resize to a fixed size, is indispensable for a reasonable performance on ImageNet. Our work considers localized labels that make the supervision provided for each random crop region more sensible. There are additional training tricks for training classifiers [59, 57, 10, 22, 8] that are orthogonal to our re-labeled training data. We show that those tricks can be combined with our re-labeling for improved performances.

[2]From a more general view of KD that utilizes teacher and student in any form, our method can be seen as a new and efficient type of KD.

3. Method

We propose a re-labeling strategy **ReLabel** to obtain pixel-level ground truth labels on the ImageNet training set. The label maps have two characteristics: (1) multi-class labels and (2) localized labels. The labels maps are obtained from a machine annotator: a strong image classifier trained on an extra data. We describe how to obtain the label maps and present a novel training framework, **LabelPooling**, to train image classifier using such localized multi-labels.

3.1. Re-labeling ImageNet

We obtain dense ground truth labels from a *machine annotator*, a state-of-the-art classifier that has been pre-trained on a super-ImageNet scale (*e.g.* JFT-300M [43] or InstagramNet-1B [32]) and fine-tuned on ImageNet to predict ImageNet classes. Predictions from such a model are arguably close to human predictions [2]. Since training the machine annotators requires an access to proprietary training data [43, 32] and hundreds of GPU or TPU days, we have adopted the open-source trained weights as the machine annotators. We show the comparison of different available machine annotators later in Section 3.3.

We remark that while the machine annotators are trained with single-label supervision on ImageNet, they still tend to make multi-label predictions for images with multiple categories. As an illustration, consider an image x with two correct categories 0 and 1. Assume that the model is fed with both $(x, y = 0)$ and $(x, y = 1)$ equal number of times during training, with those noisy labels. Then, the cross-entropy loss is given by $-\frac{1}{2}(\sum_k y_k^0 \log p_k(x) + \sum_k y_k^1 \log p_k(x)) = -\sum_k \frac{y_k^0 + y_k^1}{2} \log p_k(x)$ where y^c is the one-hot vector with 1 at index c and $p(x)$ is the prediction vector for x. Note that the minimal value for the function $-\sum_k q_k \log p_k$ with respect to p is taken at $p = q$. Thus, in this example, the model minimizes the loss by predicting $p(x) = (\frac{1}{2}, \frac{1}{2})$. Thus, if there exist much label noise in the dataset, a model trained with the single-label cross-entropy loss tends to predict multi-label outputs.

As an additional benefit of obtaining labels from a classifier, we consider extracting the location-specific labels. We remove the *global average pooling* layer of the classifier and turn the following linear layer into a 1×1 convolutional layer, thereby turning the classifier into a fully-convolutional network [60, 30]. The output of the model then becomes $f(x) \in \mathbb{R}^{W \times H \times C}$. We use the output $f(x)$ as our *label map* annotations $L \in \mathbb{R}^{W \times H \times C}$. We present the detailed procedure to obtain label maps in Appendix B.

3.2. Training a Classifier with Dense Multi-labels

Having obtained the dense multi-labels $L \in \mathbb{R}^{W \times H \times C}$ as above, how do we train a classifier with them? For this, we propose a novel training scheme, LabelPooling, that

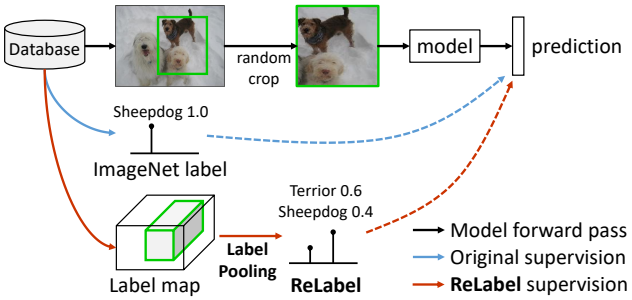


Figure 3. **Illustration of LabelPooling.** Original ImageNet supervision is single-label ("Sheepdog"). LabelPooling trains the model with ReLabel, localized multi-labels, ("Sheepdog" and "Terrior") based on the crop region.

takes the localized ground truths into account. We show the difference between LabelPooling and the original ImageNet training in Figure 3. In a standard ImageNet training setup, the supervision for the randomly crop is given by the single label ground truth given per image. On the other hand, LabelPooling loads a pre-computed label map and conducts a regional pooling operation on the label map corresponding to the coordinates of the random crop. We adopt the `RoIAlign` [14] regional pooling approach. Global average pooling and `softmax` operations are performed on the pooled prediction maps to get a multi-label ground-truth vector in $[0,1]^C$ with which the model is trained. We use the cross-entropy loss. Code-level implementation of our training scheme is presented in Appendix A.

3.3. Discussion

So far we have introduced our labeling strategy and the supervision scheme using the label maps. We study the space and time consumption for our approach and examine design choices.

Space consumption. We utilize EfficientNet-L2 [53] as the machine annotator whose input resolution is 475×475 and the resulting label map dimension is $L \in \mathbb{R}^{15\times15\times1000}$. Saving the entire label maps for all classes will require more than 1 TB of storage: (1.28×10^6) images $\times$ $(15 \times 15 \times 1000)$ $^{\text{dim}}/_{\text{image}}$ $\times$ $4^{\text{bytes}}/_{\text{dim}} \approx 1.0$ TB. Fortunately, for each image, pixel-wise predictions beyond a few top-k classes are essentially zero. Hence, we save the storage space by storing only the top-5 predictions per image, resulting in 10 GB of label map data. This corresponds to only 10% additional space on top of the original ImageNet data.

Time consumption. ReLabel requires a one-time cost for forward passing the ImageNet training images through the machine annotator. This procedure takes about 10 GPU-hours, which is only 3.3% of the entire train time for ResNet-50 (328 GPU-hours[3]). For each training iteration,

[3]300 epochs on four NVIDIA V100 GPUs.

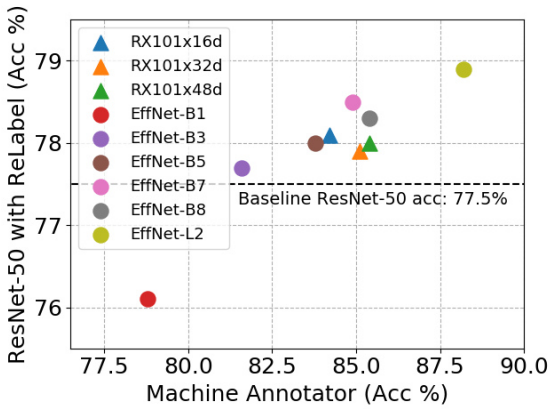


Figure 4. **Machine annotators.** We plot the top-1 accuracy of ResNet-50 trained with ReLabel, where ReLabel is generated by various machine annotators.

LabelPooling performs the label map loading and regional pooling operations on top of the standard ImageNet supervision, which leads to only 0.5% additional training time. Note that ReLabel is much more computationally efficient than knowledge distillation which requires a forward pass through the teacher at every iteration. For example, KD with EfficientNet-B7 teacher takes more than four times the original training time.

Which machine annotator should we select? Ideally, we want the machine annotator to provide precise labels on training images. For this we consider ReLabel generated by a few state-of-the-art classifiers EfficientNet-{B1,B3,B5,B7,B8} [46], EfficientNet-L2 [53] trained with JFT-300M [43], and ResNeXT-101_32x{32d,48d} [54] trained with InstagramNet-1B [32]. We train ResNet-50 with the above label maps from diverse classifiers. Note that ResNet-50 achieves the top-1 validation accuracy of 77.5% when trained on vanilla single labels. We show the results in Figure 4. The performance of the target model overall follows the performance of the machine annotator. When the machine supervision is not sufficiently strong (*e.g.*, EfficientNet-B1), the trained model shows a severe performance drop (76.1%). We choose EfficientNet-L2 as the machine annotator that has led to the best performance for ResNet-50 (78.9%) in the rest of the experiments.

Factor analysis of ReLabel. ReLabel is both multi-label and pixel-wise. To examine the necessity of the two properties, we conduct an experiment by ablating each of them. We consider the *localized single labels* by taking `argmax` operation instead of `softmax` after the `RoIAlign` regional pooling, resulting in $L_{\text{loc,single}} \in \{0,1\}^C$. For *global multi-labels*, we take the global average pooling, instead of the `RoIAlign`, over the label map, resulting in the label $L_{\text{glob,multi}} \in [0,1]^C$. Finally, by first performing the global average pooling and then performing argmax, we obtain

Variants	ImageNet top-1 (%)
ReLabel (localized mutli-labels)	78.9
Localized single labels	78.4 (-0.5)
Global multi-labels	78.5 (-0.4)
Global single labels	77.5 (-1.4)
Original ImageNet labels	77.5 (-1.4)

Table 2. **Factor analysis of ReLabel.** Results when either or both of the multi-labelness and localizability properties are removed from ReLabel.

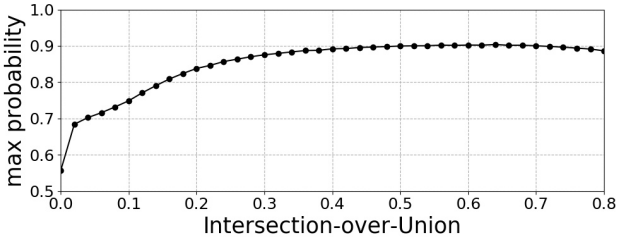


Figure 5. **ReLabel confidence versus GT overlap.** We plot the relationship between the confidence level for ReLabel pooled from the crop regions and the their overlap (IoU) with the ground truth boxes.

the *global single-labels*, $L_{\text{glob,single}} \in \{0,1\}^C$. Note that $L_{\text{glob,single}} \in \{0,1\}^C$ labels have the same format as the original ImageNet labels, but are machine-generated.

The results for those four variants are in Table 2. We observe that from the ReLabel performance of 78.9%, the removal of multi-labels and localized labels results in -0.5 pp and -0.4 pp drops, respectively. When both are missing, there is a significant -1.4 pp drop. We thus argue that both ingredients are indispensable for a good performance. Note also that the global, single labels generated by a machine do not bring about any gain compared to the original ImageNet labels. This further signifies the importance of the aforementioned properties to benefit maximally from the machine annotations.

Confidence of ReLabel supervision. We study the confidence of ReLabel supervisions at different simulated levels of overlap between the random crop and the ground-truth bounding box. We draw 5M random crop samples as done for Figure 2. We measure the confidence for the ReLabel's supervision in terms of the maximum class probability of the pooled label (*i.e.*, confidence $= \max_c L(c)$ where $L \in [0,1]^C$). The results are shown in Figure 5. The averaged degree of supervision of ReLabel overall follows the degree of object existence, in particular, with small overlaps with object region (IoU < 0.4). For example, when IoU is zero (*i.e.*, random crops are outside the object region), the label confidence is below 0.6, providing some uncertainty signals for the trained model.

4. Experiments

We present various experiments where we apply our labeling and training schemes for localized multi-label training. We first show the effectiveness of ReLabel on ImageNet classification with various network architectures and evaluation metrics, including the recently proposed multi-label evaluation metrics and robustness benchmarks (Section 4.1). Next, we show the transfer-learning performances for models trained with ReLabel when they are fine-tuned for object detection, instance segmentation, and fine-grained classification tasks (Section 4.2). We show that ReLabel improves the performances also for models on COCO multi-label classification tasks (Section 4.3). The relabeled ImageNet training set, pre-trained weights, and the source code are availalble at https://github.com/naver-ai/relabel_imagenet.

4.1. ImageNet Classification

We evaluate ReLabel strategy on the ImageNet-1K [38] containing 1.28 million training images and 50,000 validation images of 1,000 object categories. We use standard data augmentation such as random cropping, flipping, color jittering, as in [12, 10, 21, 44, 57, 48] for all the models considered. We have trained the models with SGD for 300 epochs with the initial learning rate 0.1 and the cosine learning rate scheduling without restarts [31]. The batch size and weight decay are set to $1,024$ and 0.0001, respectively.

Comparison against other label manipulations. We compare ReLabel against prior methods that directly adjust the ImageNet labels. Label smoothing [45] assigns a slightly weaker weight on the foreground class $(1-\epsilon)$ and distributes the remaining weight ϵ uniformly across background classes. Label cleaning by Beyer *et al.* [2] prunes out all training samples where the ground truth annotation does not agree with the prediction of a strong teacher classifier, namely BiT-L [24]. For this, we use the list of clean sampled provided by the authors [2] with our own training setting. We conducted the above label manipulation methods and ReLabel on ResNet-50. Results are given in Table 3. We measure the single-label accuracies on ImageNet validation and ImageNetV2 (Top-Images [36][4]). We show multi-label accuracies on two versions: ReaL [2] and Shankar *et al.* [39]. The metrics are identical: $\frac{1}{N}\sum_{n=1}^{N} 1(\arg\max f(x_n) \in y_n)$, where $1(\cdot)$ is the indicator function and $\arg\max f(x_n)$ is the top-1 prediction for a model f. The ground-truth multi-label for image x_n is given as a set y_n. The difference between the metrics lies in the ground-truth multi-label annotation. We observe that ReLabel consistently achieves the best performance over all the metrics. We obtain 78.9% validation ac-

[4]Results on ImageNetV2 "MatchedFrequency" and " Threshold 0.7" are in Appendix C

Network	Supervision	ImageNet single-label	ImageNetV2 [36] single-label	ReaL [2] multi-label	Shankar *et al*. [39] multi-label
ResNet-50	Original	77.5	79.0	83.6	85.3
ResNet-50	Label smoothing (ϵ=0.1) [45]	78.0	79.5	84.0	84.7
ResNet-50	Label cleaning [2]	78.1	79.1	83.6	85.2
ResNet-50	**ReLabel**	**78.9**	**80.5**	**85.0**	**86.1**

Table 3. **ImageNet classification.** Results with different types of supervision. We report performances on the single-label benchmarks (ImageNet validation set and ImageNetV2 [36]) and multi-label benchmarks (ReaL [2] and Shankar *et al*. [39]).

Architecture	Resources Params	Flops	Supervision Vanilla	ReLabel
ResNet-18	11.7M	1.8B	71.7	72.5 (+0.8)
ResNet-50	25.6M	3.8B	77.5	78.9 (+1.4)
ResNet-101	44.7M	7.6B	78.1	80.7 (+2.6)
EfficientNet-B0	5.3M	0.4B	77.4	78.0 (+0.6)
EfficientNet-B1	7.8M	0.7B	79.2	80.3 (+1.1)
EfficientNet-B2	9.2M	1.0B	80.3	81.0 (+0.7)
EfficientNet-B3	12.2M	1.8B	81.7	82.5 (+0.8)
ReXNet ($\times$1.0)	4.8M	0.4B	77.9	78.4 (+0.5)

Table 4. **ReLabel on multiple architectures.** Validation top-1 results when supervised with the original labels (Vanilla) and ReLabel.

Model	ImageNet top1 (%)
ResNet-50	77.5
+ **ReLabel**	78.9 (+1.4)
+ **ReLabel** + CutMix	80.2 (+2.7)
+ **ReLabel** + CutMix + Extra data	81.2 (+3.7)
ResNet-101	78.1
+ **ReLabel**	80.7 (+2.6)
+ **ReLabel** + CutMix	81.6 (+3.5)

Table 5. **Towards the SOTA.** ReLabel with additional training tricks. "Extra data" refers to the ImageNet-21k dataset.

curacy with $+1.4$ pp gain from the original labels, while the label smoothing and label cleaning boost only $+0.5$ pp and $+0.6$ pp, respectively. On ImageNetV2, ReaL, and Shankar *et al*. metrics, ReLabel achieves 80.5%, 85.0%, and 86.1% accuracies, where the gains are $+1.5$ pp, $+1.4$ pp, and $+0.8$ pp, respectively. It is notable that only ReLabel achieves remarkable boosts on the multi-label benchmarks. Label smoothing and cleaning shows only marginal gains or even worse multi-label accuracies (*e.g*. label cleaning results in a 0.1 pp worse result on Shankar *et al*.). We confirm that ReLabel improves the performances of image classifiers and that it helps models truly learn to make better multi-label predictions.

Results on various network architectures. We have trained various architectures with ReLabel to show that ReLabel is applicable to a wide range of networks with different training recipes. We consider ResNet-18, ResNet-101, EfficientNet-{B0,B1,B2,B3} [46], and ReXNet [13]. Training details to make the best performance out of EfficientNet models [46] are different from our base setting; we describe them in Appendix D.2. We follow the original paper's training details for ReXNet [13]. Results are shown in Table 4. ReLabel consistently enhances the performance of various network architectures. The 81.7% accuracy of EfficientNet-B3 is further improved to 82.5% with ReLabel.

State-of-the-art performance. ReLabel is complementary to many other training tricks used for achieving the best model performances. For example, we combine a strong regularizer CutMix [57] with ReLabel. CutMix mixes two training images via cut-and-paste manner and likewise mixes the labels. To use it with ReLabel, we perform CutMix on the randomly cropped images. The pooled labels are then mixed according to the CutMix algorithm. We set the hyper-parameter of CutMix α to 1.0. We show the results in Table 5. ReLabel with CutMix achieves the state-of-the-art ImageNet top-1 accuracies of **80.2%** and **81.6%** for the ResNet-50 and ResNet-101 backbones. On top of this, we further consider using the extra training data based on the ImageNet-21K dataset [7]: 14M images with 21K categories. Unlike the previous work utilizing the ImageNet-21K [24] with their original single-class labels over 21K categories, we perform ReLabel on them to generate multi-labels over the 1K classes. We then sub-sample 4M training data from the entire 14M training images by balancing the top-1 class distributions, as done in [54]. Training with this extra data and CutMix on top of ReLabel boosts the accuracy of ResNet-50 to **81.2%**. In summary, ReLabel is a practical addition to existing training tricks that consistently improves the backbone performances.

Comparison against knowledge distillation. We compare ReLabel against knowledge distillation (KD) [20] in terms of the performance and training time costs. We train ResNet-50 with EfficientNet teachers: EfficientNet-{B1,B3,B5,B7}; we have not considered performing KD with EfficientNet-L2 as it would take 160 GPU days, be-

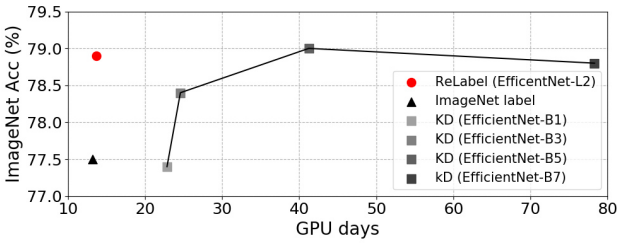
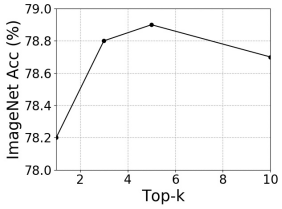


Figure 6. **Comparison against knowledge distillation.** We plot ImageNet top-1 accuracies against the required training time for ReLabel and knowledge distillation (KD) approaches.



ImageNet top1 (%)	
Baseline	77.5
ReLabel	78.9
16-bit label map	78.8
8-bit label map	78.6

Table 6. **Storage versus performance.** How much performance do we lose by trying to cut storage space?

yond our computational capacity. Training details for KD are in Appendix D.3. Figure 6 shows the results. We plot the target model's performance versus the required number of GPU days. KD with smaller teacher variants (EfficientNet-{B1,B3}) shows worse top-1 accuracies than ReLabel at higher training costs. For larger teachers (EfficientNet-{B5,B7}), KD achieves comparable performances with ReLabel (*e.g.* 79.0% for KD with B7 and 78.8% for ReLabel). However, they require 41 and 78 GPU days to train, compared to mere 13.6 using ReLabel. ReLabel training is almost as fast as the original training.

Storage-performance trade off. We study the trade off between the storage space for the label maps and the model performance. ReLabel only saves top-k prediction maps for the interest of efficient storage, where the default k value is 5. We explore $k \in \{1, 3, 5, 10\}$. We also study the impact of quantization levels for label maps: 16-bit and 8-bit floating point, instead of the default 32-bit floating point. The results are in Table 6. ReLabel achieves a good performance-efficiency trade-off when $k = 5$. Quantizing label maps tend to yield only small performance drops (-0.1 pp to -0.3 pp). When storage space is a crucial constraint, we advise users to adopt labels maps of coarser formats.

Combination with original labels. When we combine ReLabel's annotation $L_{\text{ours}} \in [0, 1]^C$ with the original label $L_{\text{gt}} \in \{0, 1\}^C$ as $0.5L_{\text{ours}} + 0.5L_{\text{gt}}$, the performance degrades from 78.9% to 78.3% accuracy on ImageNet. They do not seem to make a good combination.

Robustness. We evaluate the robustness of ReLabel-trained models against test-time perturbations. We consider adversarial and natural perturbations: FGSM [11], ImageNet-A [17], ImageNet-C [16], and background challenge (BGC) [52]. FGSM introduces one-step adversarial perturbations on images, while ImageNet-A samples consistent of common failure cases for modern image classifiers. ImageNet-C consists of 15 different types of natural perturbations. BGC evaluates the robustness against backgrounds by selecting background images adversarially from the dataset. Results are in Table 7. We observe that ReLabel consistently improves the resilience of models on adversarial and natural perturbations. Especially, ReLabel shows remarkable improvements in the background robustness (+8.7%) owing the localized supervision. Furthermore, combining ReLabel with other training strategies, *e.g.*, CutMix [57] and extra training data, significantly boosts the performances in the all robustness benchmarks.

Models	FGSM	ImageNet-A	ImageNet-C	BCG
ResNet-50	25.7	4.9	27.9	25.9
+ **ReLabel**	31.3 (+5.6)	7.1 (+2.2)	28.1 (+0.2)	34.6 (+8.7)
+ **CutMix**	42.4 (+16.7)	11.4 (+6.5)	47.5 (+19.6)	34.1 (+8.2)
+ **Extra data**	45.0 (+19.3)	24.8 (+19.9)	54.2 (+26.3)	36.0 (+10.1)

Table 7. **Robustness.** Impact of ReLabel on FGSM [11], ImageNet-A [17], ImageNet-C [16], and background challenge [52] benchmarks. All numbers are accuracies.

ReLabel examples on ImageNet. We present examples generated by ReLabel during ImageNet training in Appendix E. As shown in the examples, ReLabel can generate location-specific multi-labels with more precise supervision than the original ImageNet labels.

4.2. Transfer Learning

Apart from serving as the standard benchmark, ImageNet has contributed to the computer vision research and engineering with its suite of pre-trained models. When the target task has only a small number of annotated data, transfer learning from the ImageNet pre-training usually helps [25]. We examine here whether the ReLabel-induced improvements on the ImageNet performances transfer to various downstream tasks. We present the results of 5 fine-grained classification tasks and the object detection and instance segmentation tasks on COCO with models pre-trained on ImageNet with ReLabel.

Fine-grained classification tasks. We evaluate ReLabel-pretrained ResNet-50 on five fine-grained classification tasks: Food-101 [3], Stanford Cars [26], DTD [6], FGVC Aircraft [33], and Oxford Pets [35]. We use the standard data augmentation as in Section 4.1. Models are fine-tuned with SGD for 5,000 iterations, following the convention for fine-tuning tasks [4]. To find the best learning rate and weight decay values for each task, we perform a grid search

	Food-101 [3]	Stanford Cars [26]	DTD [6]	FGVC Aircraft [33]	Oxford Pets [35]
ResNet-50 (Baseline)	87.98	92.64	75.43	85.09	93.92
ResNet-50 (**ReLabel**-trained)	88.12	92.73	75.74	88.89	94.28

Table 8. **Fine-grained classification.** Performance on five tasks where the model starts either from weights regularly pre-trained on ImageNet or from weights pre-trained via ReLabel.

	Faster-RCNN	Mask-RCNN	
	bbox AP	bbox AP	mask AP
ResNet-50 (Baseline)	37.7	38.5	34.7
ResNet-50 (**ReLabel**-trained)	38.2	39.1	35.2

Table 9. **Detection and instance segmentation.** Transfer learning performances for Faster-RCNN [37] and Mask-RCNN [14] on COCO dataset [29].

	COCO (mAP)
ResNet-50	69.0
ResNet-50 + **ReLabel** (machine)	72.7
ResNet-50 + **ReLabel** (oracle)	73.2
ResNet-101	76.6
ResNet-101 + **ReLabel** (machine)	79.0
ResNet-101 + **ReLabel** (oracle)	80.9

Table 10. **Originally multi-label tasks.** Results of ReLabel on COCO multi-class classification task [29].

per task and report the best performance. Table 8 shows the results. Note that the ReLabel-trained model results in a consistent improvement over the vanilla pre-trained model. For example, on FGVC Aircraft, ReLabel pre-training improves the downstream task performance by $+3.8$ pp.

Object detection and instance segmentation. We used Faster-RCNN [37] and Mask-RCNN [14] with feature pyramid network (FPN [28]) as the base models for object detection and instance segmentation tasks, respectively. The backbone networks of Faster-RCNN and Mask-RCNN are initialized with ReLabel-pretrained ResNet-50 model, and then fine-tuned on COCO dataset [29] by the original training strategy [37, 14] with the image size of 1200×800. Table 9 shows the results. Pre-training with ReLabel improves the bbox AP of Faster-RCNN by $+0.5$ pp and the mask AP of Mask-RCNN by $+0.5$ pp. Pre-training a model with cleaner supervision like ReLabel leads to better feature representations and boosts the object detection and instance segmentation performances.

4.3. Multi-label Classification

ReLabel is designed to transform a single-label training set into a multi-label training set. Nonetheless, ReLabel and LabelPooling also helps improving an originally multi-label training set by providing additional localized supervision signals, given that the random crop augmentation is a popular recipe for multi-label training as well [5, 51, 56]. To see this effect, we experiment with the multi-label classification dataset COCO [29] with multiple human-annotated labels per image. The baseline multi-label training uses multi-hot annotation $L \in \{0,1\}^C$ ($C = 80$ for COCO). Instead, we utilize the segmentation ground truth of COCO dataset as label maps $L \in \{0,1\}^{H \times W \times C}$ (*i.e.*, an *oracle* case of ReLabel). We also compare with machine-generated label maps $L \in \mathbb{R}^{H \times W \times C}$ from a state-of-the-art multi-label classifier [1] to see the effectiveness of the oracle label map. We then train our multi-label classifiers with our LabelPooling based on the label maps according to the random crop coordinates. We conduct experiments on ResNet-50 and ResNet-101 networks with the input size 224×224 and 448×448, respectively, using the binary cross-entropy loss for all methods considered. More training details are in Appendix D.4. Table 10 shows the results. We observe that applying ReLabel with machine-generated label maps results in $+3.7$ pp and $+2.4$ pp mAP gains and, with oracle label maps, ReLabel achieves more gain of **+4.2** pp and **+4.3** pp mAP gains on ResNet-50 and ResNet-101 networks, respectively. In summary, the location-wise supervision from ReLabel helps the multi-label classification training.

5. Conclusion

We have proposed a re-labeling strategy, **ReLabel**, for the 1.28 million training images on ImageNet. ReLabel transforms the single-class labels assigned once per image into multi-class labels assigned for every region in an image, based on a machine annotator. The machine annotator is a strong classifier trained on a large extra source of visual data. We also proposed a novel scheme for training a classifier with the localized multi-class labels (**LabelPooling**). We experimentally verified significant performance gains induced by our labels and the corresponding training technique. ReLabel results in a consistent gain across tasks, including the ImageNet benchmarks, transfer-learning tasks, and multi-label classification tasks. We will open-source the localized multi-labels from ReLabel and the corresponding pre-trained models.

Acknowledgement We thank NAVER AI Lab members for valuable discussion and advice. NAVER Smart Machine Learning (NSML) [23] has been used for experiments.

References

[1] Emanuel Ben-Baruch, Tal Ridnik, Nadav Zamir, Asaf Noy, Itamar Friedman, Matan Protter, and Lihi Zelnik-Manor. Asymmetric loss for multi-label classification. *arXiv preprint arXiv:2009.14119*, 2020. 8

[2] Lucas Beyer, Olivier J Hénaff, Alexander Kolesnikov, Xiaohua Zhai, and Aäron van den Oord. Are we done with imagenet? *arXiv preprint arXiv:2006.07159*, 2020. 1, 2, 3, 5, 6

[3] Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101–mining discriminative components with random forests. In *European conference on computer vision*, pages 446–461. Springer, 2014. 7, 8

[4] Ken Chatfield, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Return of the devil in the details: Delving deep into convolutional nets. *arXiv preprint arXiv:1405.3531*, 2014. 7

[5] Zhao-Min Chen, Xiu-Shen Wei, Peng Wang, and Yanwen Guo. Multi-label image recognition with graph convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5177–5186, 2019. 8

[6] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 7, 8

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009. 6

[8] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017. 3

[9] Tommaso Furlanello, Zachary C Lipton, Michael Tschannen, Laurent Itti, and Anima Anandkumar. Born again neural networks. *arXiv preprint arXiv:1805.04770*, 2018. 3

[10] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Dropblock: A regularization method for convolutional networks. In *Advances in Neural Information Processing Systems*, pages 10750–10760, 2018. 3, 5

[11] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*, 2015. 7

[12] Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *CVPR*, 2017. 3, 5

[13] Dongyoon Han, Sangdoo Yun, Byeongho Heo, and YoungJoon Yoo. Rexnet: Diminishing representational bottleneck on convolutional neural network. *arXiv preprint arXiv:2007.00992*, 2020. 6

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 4, 8

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 1

[16] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations*, 2019. 7

[17] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. *arXiv preprint arXiv:1907.07174*, 2019. 7

[18] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1921–1930, 2019. 3

[19] Byeongho Heo, Minsik Lee, Sangdoo Yun, and Jin Young Choi. Knowledge transfer via distillation of activation boundaries formed by hidden neurons. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 3779–3787, 2019. 3

[20] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2, 3, 6

[21] Gao Huang, Zhuang Liu, and Kilian Q Weinberger. Densely connected convolutional networks. In *CVPR*, 2017. 3, 5

[22] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Weinberger. Deep networks with stochastic depth. In *ECCV*, 2016. 3

[23] Hanjoo Kim, Minkyu Kim, Dongjoo Seo, Jinwoong Kim, Heungseok Park, Soeun Park, Hyunwoo Jo, KyungHyun Kim, Youngil Yang, Youngkwan Kim, Nako Sung, and Jung-Woo Ha. NSML: meet the mlaas platform with a real-world case study. *CoRR*, abs/1810.09957, 2018. 8

[24] Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Joan Puigcerver, Jessica Yung, Sylvain Gelly, and Neil Houlsby. Big transfer (bit): General visual representation learning. *arXiv preprint arXiv:1912.11370*, 6, 2019. 5, 6

[25] Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2661–2671, 2019. 1, 7

[26] Jonathan Krause, Jia Deng, Michael Stark, and Li Fei-Fei. Collecting a large-scale dataset of fine-grained cars. *In Second Workshop on Fine-Grained Visual Categorization*, 2013. 7, 8

[27] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[28] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 8

[29] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 8

[30] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, 2015. 3

[31] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 5

[32] Dhruv Mahajan, Ross Girshick, Vignesh Ramanathan, Kaiming He, Manohar Paluri, Yixuan Li, Ashwin Bharambe, and Laurens van der Maaten. Exploring the limits of weakly supervised pretraining. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 181–196, 2018. 3, 4

[33] Subhransu Maji, Esa Rahtu, Juho Kannala, Matthew Blaschko, and Andrea Vedaldi. Fine-grained visual classification of aircraft. *arXiv preprint arXiv:1306.5151*, 2013. 7, 8

[34] Wonpyo Park, Dongju Kim, Yan Lu, and Minsu Cho. Relational knowledge distillation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3967–3976, 2019. 3

[35] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 7, 8

[36] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do imagenet classifiers generalize to imagenet? In *International Conference on Machine Learning*, pages 5389–5400, 2019. 2, 5, 6

[37] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 8

[38] 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. 1, 2, 5

[39] Vaishaal Shankar, Rebecca Roelofs, Horia Mania, Alex Fang, Benjamin Recht, and Ludwig Schmidt. Evaluating machine accuracy on imagenet. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. 1, 2, 3, 5, 6

[40] Zhiqiang Shen, Zhankui He, and Xiangyang Xue. Meal: Multi-model ensemble via adversarial learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 4886–4893, 2019. 3

[41] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 1

[42] Pierre Stock and Moustapha Cisse. Convnets and imagenet beyond accuracy: Understanding mistakes and uncovering biases. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 498–512, 2018. 1, 2

[43] Chen Sun, Abhinav Shrivastava, Saurabh Singh, and Abhinav Gupta. Revisiting unreasonable effectiveness of data in deep learning era. In *Proceedings of the IEEE international conference on computer vision*, pages 843–852, 2017. 3, 4

[44] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *CVPR*, 2015. 2, 3, 5

[45] 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*, pages 2818–2826, 2016. 5, 6

[46] Mingxing Tan and Quoc Le. Efficientnet: Rethinking model scaling for convolutional neural networks. In *International Conference on Machine Learning*, pages 6105–6114, 2019. 4, 6

[47] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive representation distillation. *arXiv preprint arXiv:1910.10699*, 2019. 3

[48] Hugo Touvron, Andrea Vedaldi, Matthijs Douze, and Hervé Jégou. Fixing the train-test resolution discrepancy. In *Advances in Neural Information Processing Systems*, pages 8252–8262, 2019. 2, 3, 5

[49] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. From imagenet to image classification: Contextualizing progress on benchmarks. In *Proceedings of the 37th International Conference on Machine Learning*, 2020. 1, 2

[50] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 595–604, 2015. 2

[51] Zhouxia Wang, Tianshui Chen, Guanbin Li, Ruijia Xu, and Liang Lin. Multi-label image recognition by recurrently discovering attentional regions. In *Proceedings of the IEEE international conference on computer vision*, pages 464–472, 2017. 8

[52] Kai Xiao, Logan Engstrom, Andrew Ilyas, and Aleksander Madry. Noise or signal: The role of image backgrounds in object recognition. *arXiv preprint arXiv:2006.09994*, 2020. 7

[53] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10687–10698, 2020. 2, 3, 4

[54] I Zeki Yalniz, Hervé Jégou, Kan Chen, Manohar Paluri, and Dhruv Mahajan. Billion-scale semi-supervised learning for image classification. *arXiv preprint arXiv:1905.00546*, 2019. 4, 6

[55] Chenglin Yang, Lingxi Xie, Chi Su, and Alan L Yuille. Snapshot distillation: Teacher-student optimization in one generation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2859–2868, 2019. 3

[56] Renchun You, Zhiyao Guo, Lei Cui, Xiang Long, Yingze Bao, and Shilei Wen. Cross-modality attention with semantic graph embedding for multi-label classification. In *AAAI*, pages 12709–12716, 2020. 8

[57] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable

features. In *International Conference on Computer Vision (ICCV)*, 2019. 1, 3, 5, 6, 7

[58] Sergey Zagoruyko and Nikos Komodakis. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. *arXiv preprint arXiv:1612.03928*, 2016. 3

[59] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 3

[60] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016. 3

[61] Xiatian Zhu, Shaogang Gong, et al. Knowledge distillation by on-the-fly native ensemble. In *Advances in neural information processing systems*, pages 7517–7527, 2018. 3