Understanding the Impact of Label Granularity on CNN-based Image Classification

Zhuo Chen[1], Ruizhou Ding[1], Ting-Wu Chin, Diana Marculescu
Electrical and Computer Engineering
Carnegie Mellon University
Pittsburgh, USA
{tonychen, rding, tingwuc, dianam}@cmu.edu

***Abstract*—In recent years, supervised learning using Convolutional Neural Networks (CNNs) has achieved great success in image classification tasks, and large scale labeled datasets have contributed significantly to this achievement. However, the definition of a *label* is often application dependent. For example, an image of a cat can be labeled as "cat" or perhaps more specifically "Persian cat." We refer to this as *label granularity*. In this paper, we conduct extensive experiments using various datasets to demonstrate and analyze how and why training based on fine-grain labeling, such as "Persian cat" can improve CNN accuracy on classifying coarse-grain classes, in this case "cat."**

The experimental results show that training CNNs with fine-grain labels improves both network's optimization and generalization capabilities, as intuitively it encourages the network to learn more features, and hence increases classification accuracy on coarse-grain classes under all datasets considered. Moreover, fine-grain labels enhance data efficiency in CNN training. For example, a CNN trained with fine-grain labels and only 40% of the total training data can achieve higher accuracy than a CNN trained with the full training dataset and coarse-grain labels. These results point to two possible applications of this work: (i) with sufficient human resources, one can improve CNN performance by re-labeling the dataset with fine-grain labels, and (ii) with limited human resources, to improve CNN performance, rather than collecting more training data, one may instead use fine-grain labels for the dataset. We also observe that the improvement brought by fine-grain labeling varies from dataset to dataset, therefore we further propose a metric called *Average Confusion Ratio* to characterize the effectiveness of fine-grain labeling, and show its use through extensive experimentation. Code is available at https://github.com/cmu-enyac/Label-Granularity.

***Index Terms*—Convolutional Neural Networks, Supervised Learning, Image Classification, Labeling**

I. Introduction

We have witnessed tremendous improvement in image classification tasks in recent years thanks to the use of supervised learning combined with the powerful model of Convolutional Neural Networks (CNNs) [1]. At the same time, the use of large-scale labeled datasets is one of the key elements that has led to this breakthrough [2] [3]. However, the definition of label varies from application to application, and there is hardly a universal definition of what a "correct" label is for an image. One such example is how detailed the label should be, *i.e.*, label granularity (or label hierarchy), as illustrated in Figure 1. For example, in the case of animal image classification, it may be sufficient to label all images of carnivores as "carnivore", while in an application of carnivore classification, we may label different images as "dog", "cat", *etc.*, which are fine-grain labels of the coarse-grain label "carnivore". Therefore, it is equally correct to label the image of a dog as "carnivore" or "dog", yet deciding on which label of the two should be used depends on the task. We denote a fine-grain (coarse-grain) class as a class of images that are labeled with the respective fine-grain (coarse-grain) label, and fine-grain (coarse-grain) training as the training process of CNNs using fine-grain (coarse-grain) labels. If the task at hand is classifying coarse-grain classes, *e.g.*, "carnivore" vs. "herbivore", the following question arises: should we directly train and test a CNN using coarse-grain labels as it has usually been done, or would it be beneficial if we trained a CNN with fine-grain labels, *e.g.*, "dog", "cat", "horse", "deer", *etc.* and map them back to coarse-grain labels during testing phase? The first

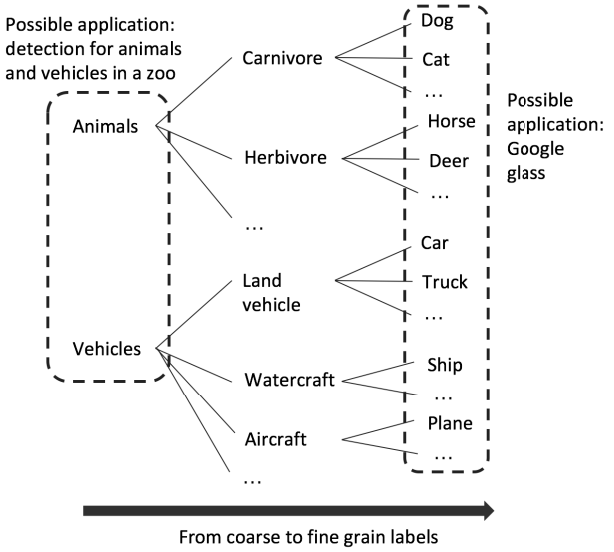


Fig. 1: An example of label granularity (label hierarchy). For example, an image of a dog can be labeled "animal" or "carnivore" or "dog", and it is the target application that determines which label to use. This paper explores whether one should use the targeted coarse-grain labels or finer-grain labels for CNN training.

[1]Authors contributed equally

2375-9259/18/$31.00 ©2018 IEEE
DOI 10.1109/ICDMW.2018.00131

TABLE I: Training and testing accuracy of five datasets when trained with fine-grain labeling (bottom row for each dataset) vs. coarse-grain labeling (top row for each dataset), and tested on coarse-grain labels.

Dataset	# of training classes	# of testing classes	Training accuracy (%)	Testing accuracy (%)
CIFAR-10	2	2	99.9	98.42
	10	2	100.0	99.20
CIFAR-100	20	20	100.0	85.04
	100	20	100.0	85.05
CIFAR-100 animals	10	10	100.0	81.42
	50	10	100.0	83.44
ImageNet dog vs. cat	2	2	94.1	92.68
	10	2	95.3	94.67
ImageNet fruit vs. vege.	2	2	91.8	89.65
	17	2	95.4	93.15

approach is a method commonly used in image classification tasks [4], however, in our experiments, we find that training CNNs with fine-grain labels can achieve higher accuracy than using coarse-grain labels in most of the datasets considered. Table I shows both training and testing accuracy of coarse-grain classification using either coarse-grain or fine-grain labeling. We can see that fine-grain labeling helps improve both training accuracy (network optimization), and testing accuracy (network generalization) across representative image datasets: CIFAR-10 [3], CIFAR-100 [3], and ImageNet [2]. Moreover, helped by fine-grain labeling, the training process converges faster and requires less amount of training data to achieve the same level of testing accuracy, *i.e.*, becomes more data efficient. More specifically, for the CIFAR-10 dataset and two ImageNet subsets, a CNN trained with fine-grain labels and only 40% of the total training data can achieve even higher accuracy than a CNN trained with full training dataset but coarse-grain labels.

In this paper, we design and conduct extensive experiments on various datasets to investigate this interesting phenomenon, and analyze and shed some light on how and why fine-grain label helps enhance coarse-grain image classification. We further propose a metric called *Average Confusion Ratio* (ACR) to characterize the accuracy improvement of fine-grain training, and verify its effectiveness through extensive experiments under different datasets. Our results show two potential practical use of this work: (i) when human resources are abundant, we can increase CNN accuracy by re-labeling the dataset with fine-grain labels and train the CNN using these new labels, and (ii) when human resources are limited and training data is hard to obtain, rather than relying on collecting more training data to improve CNN accuracy, we may instead re-label the dataset with fine-grain labels.

II. Related Work

To the best of our knowledge, this is the first work to analyze the use of finer-gain labeling for improving accuracy and training data efficiency for CNN-based image classification tasks. Though there has been significant prior work looking into the hierarchy of classes/categories [5]–[16], our work has a distinct objective compared to prior art. Some of the prior work [6], [11], [13] aim to utilize the hierarchical label information to improve classification accuracy for the finest categories. On the theory side, Dekel *et al.* [6] propose a learning framework using large margin kernel methods and Bayesian analysis to deal with the classification problem with hierarchical label structures. Cesa-Bianchi *et al.* [13] propose a new loss function and use Support Vector Machine (SVM) as well as a probabilistic data model so that higher accuracy can be achieved with exponentially fast convergence speed. From a more practical viewpoint, Zhao *et al.* [11] leverage hierarchical information of the class structure and select different feature subsets for super-classes. Other work [14] [7] [10] aim to predict either coarse- or fine-grain labels conditioning on the confidence level. Deng *et al.* [14] optimize the trade-off between specificity (how fine-grain the predicted label is) and accuracy, while Bi *et al.* [7] develop a Bayes-optimal classifier to minimize the Bayesian risk. More recently, Wang *et al.* propose to stop the prediction process for a coarse-grain label so as to avoid an incorrect prediction. In addition, other prior work [8] [15] [9] focuses on the understanding of hierarchical labels. For example, Song *et al.* [8] study dataless hierarchical text classification with unsupervised methods. Hoyoux *et al.* [15] show some counter-examples where using hierarchical methods degrades the accuracy, and explore the reasons for such results. Oh [9] studies the combination of hierarchical classification and top-*k* accuracy.

However, all these studies aim to increase the classification accuracy of fine-grain classes. Instead, we focus on the case of coarse-grain classes being the target of classification task, and we explore whether directly training with finer-grain labels can achieve higher classification accuracy on coarse-grain classes than training with coarse-grain labels.

A work close to ours is done by Mo *et al.* [17], who propose active over-labeling to generate finer-grain labels than the target coarse-grain labels, and demonstrate that fine-grained label data can improve precision of a classifier for the coarse-grained concept. Similar ideas were also studied by other prior work [18]–[22]. However, none of them explores deep learning models which are very different from conventional machine learning models, *e.g.*, Support Vector Machine (SVM), logistic regression, etc. Fradkin [20] performs experiments on linear and non-linear SVM, and finds that fine-grain training can improve accuracy for linear SVM since fine-grain labeling can learn a piece-wise linear decision boundary that better approximate the true non-linear boundary. However, fine-grain training does not help non-linear (RBF-kernel) SVM due to their inherent non-linearity. CNNs are highly non-linear models and relatively more difficult to optimize [23]. No results of CNNs has yet been shown on this topic.

The remainder of this paper is organized as follows. Section III demonstrates in detail the effects of fine-grain labeling improving CNN-based image classification and its capability of enhancing training data efficiency. Section IV analyzes why fine-grain labeling helps in terms of both network *optimization* and *generalization* via extensive experiments on various

datasets. In Section V, we propose a metric called *Average Confusion Ratio* to characterize the accuracy gain of fine-grain training, and verify its effectiveness under different datasets. In Section VI, we further discuss how (i) customized coarse-grain classes for diverse applications, (ii) noisy fine-grain labels obtained via automatic clustering methods, and (iii) the number of coarse-grain classes may impact effectiveness of fine-grain training. We conclude our work in Section VII.

III. Label Granularity and Training Data

In this section, we demonstrate the effects of fine-grain labels on improving image classification accuracy and further show its capability of enhancing training efficiency.

We define A_{FC}^{train} and A_{FC}^{test} as the training and testing accuracy of a CNN trained on fine-grain labels and evaluated on coarse-grain labels, respectively. In detail, we first train a network with fine-grain labels and output the predicted fine-grain labels of all input images. Then we map the predicted fine-grain labels to their respective coarse-grain labels via the predefined mapping as shown in table II. Finally, the accuracy is computed by comparing the predicted coarse-grain labels with the ground-truth labels. Similarly, we define A_{CC}^{train} and A_{CC}^{test} as the training and testing accuracy of a CNN trained on coarse-grain labels and evaluated on the same labels. To do this, we directly train a network with coarse-grain labels and compute accuracy by comparing the predicted labels, which are already coarse-grain labels, of the input images with their ground-truth labels.

We design and conduct experiments on well-known image classification datasets: CIFAR-10 [3], CIFAR-100 [3] and ImageNet [2], and we list their coarse- and fine-grain classes in Table II. CIFAR-10 dataset is a great fit for applications similar to the one shown in Figure 1, *i.e.*, classifying whether an image contains an animal or a vehicle. CIFAR-10 has six animals: "bird", "cat", "deer", "dog", "frog", "horse", and four vehicles: "plane","car", "ship", "truck". CIFAR-100 provides 20 coarse-grain classes and five fine-grain classes per coarse-grain class, resulting in 100 fine-grain classes in total. We also select all ten animal coarse-grain classes from CIFAR-100 to form another dataset serving applications like animal classification, and we call this dataset: CIFAR-100 animals. ImageNet dataset is collected and organized according to the WordNet hierarchy [2], [24] and therefore it naturally follows the coarse-to-fine-grain label hierarchy. We use subsets of ImageNet dataset to better visualize and demonstrate the benefits of training CNN with fine-grain labels. The first ImageNet subset task is to classify dog vs. cat, with a total of ten fine-grain classes of random breeds of dogs and cats. The second task is classifying fruit vs. vegetable with a total of 17 fine-grain classes.

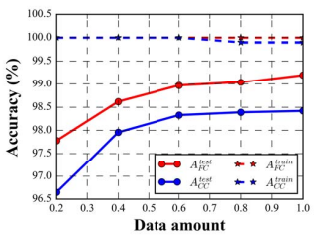
Table III shows the network configurations used for different datasets. For CIFAR-10, we use the full pre-activation residual network with 512 filters for the widest layer similar to the one in [25]. We use wide residual network [26] for CIFAR-100 dataset, which achieves 81.15% accuracy on 100 classes. We use "thinner" networks for ImageNet subsets to avoid overfitting because ImageNet subsets have fewer training images (22K images for fruit vs. vegetable, and 13K images for dog vs. cat) than CIFAR-10 (50K images) and CIFAR-100 (50K images). The network configuration for ImageNet subsets is similar to CIFAR-10, but with 75% fewer filters per convolution layer. We use random cropping and random flipping data augmentation [27] for all datasets. For the training configuration, we use momentum 0.9 and weight decay 5e-4. The learning rate starts at 0.1 for CIFAR-10 and CIFAR-100, and 0.01 for ImageNet subsets, and decays when the loss plateaus. We train CIFAR-10 and CIFAR-100 for 200

TABLE II: Coarse-grain and fine-grain classes of five datasets.

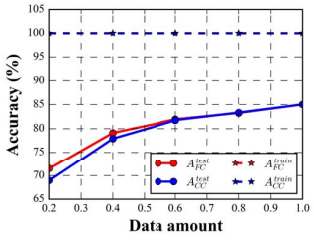
Dataset	Coarse-grain classes	Fine-grain classes
CIFAR-10	animal	bird, cat, deer, dog, frog, horse
	vehicle	plane, car, ship, truck
CIFAR-100	aquatic mammals*	beaver, dolphin, otter, seal, whale
	fish*	aquarium fish, flatfish, ray, shark, trout
	flowers	orchid, poppy, rose, sunflower, tulip
	food containers	bottle, bowl, can, cup, plate
	fruit and vegetables	apple, mushroom, orange, pear, sweet pepper
	household electrical devices	clock, keyboard, lamp, telephone, television
	household furniture	bed, chair, couch, table, wardrobe
	insects*	bee, beetle, butterfly, caterpillar, cockroach
	large carnivores*	bear, leopard, lion, tiger, wolf
	large man-made outdoor things	bridge, castle, house, road, skyscraper
	large natural outdoor scenes	cloud, forest, mountain, plain, sea
	large omnivores and herbivores*	camel, cattle, chimpanzee, elephant, kangaroo
	medium mammals*	fox, porcupine, possum, raccoon, skunk
	non-insect invertebrates*	crab, lobster, snail, spider, worm
	people*	baby, boy, girl, man, woman
	reptiles*	crocodile, dinosaur, lizard, snake, turtle
	small mammals*	hamster, mouse, rabbit, shrew, squirrel
	trees	maple tree, oak tree, palm tree, pine tree, willow tree
	vehicles 1	bicycle, bus, motorcycle, pickup truck, train
	vehicles 2	lawn mower, rocket, streetcar, tank, tractor
CIFAR-100 animals	(10 coarse-grain classes above marked with *)	(50 corresponding fine-grain classes)
ImageNet dog vs. cat	dog	basset, chihuahua, maltese, papillon, pekinese,
	cat	tabby, tiger cat, Persian, Siamese, Egyptian
ImageNet fruit vs. vege	fruit	strawberry, orange, lemon, fig, pineapple, banana, jackfruit, custard apple
	vege	head cabbage, broccoli, cauliflower, zucchini, butternut squash, cucumber, artichoke, pepper, mushroom

TABLE III: Configuration of CNNs used in the experiments.

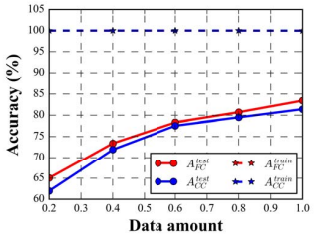
Dataset	# of layers	# of filters in widest layer	# of parameters
CIFAR-10	18	512	11.1M
CIFAR-100 CIFAR-100 animals	26	640	36.5M
ImageNet subsets	18	128	0.7M



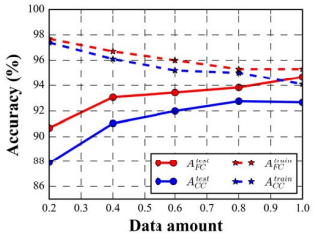
(a) CIFAR-10



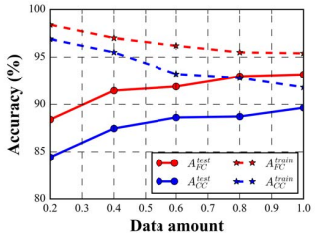
(b) CIFAR-100



(c) CIFAR-100-animal



(d) Dog vs. cat



(e) Fruit vs. vegetable

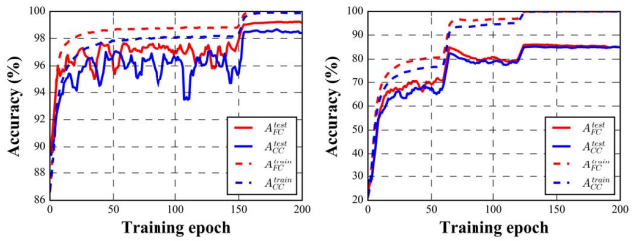
Fig. 2: Training (dotted) and testing (solid) accuracy curves with increasing amount of training data. CNNs trained with fine-grain labels are shown in red and those trained with coarse-grain labels are shown in blue. Experiments are conducted using five datasets: (a) CIFAR-10, (b) CIFAR-100, (c) CIFAR-100-animal, and two subsets of ImageNet datasets (d) dog vs. cat, (e) fruit vs. vegetable.

epochs, and ImageNet subsets for 225 epochs. The above CNN architectures and training schemes are derived from existing works, which are usually near-optimal via extensive hyper-parameter tuning. To keep the comparison fair, when we train on fine-grain labels, we keep everything unchanged, except for the last layer which will need to output more classes. Note that without fine-tuning on our fine-grain training, we are still able to outperform the hand-tuned models.

Table I shows the results. The second column gives the number of classes CNN is trained on and the third column shows the number of classes the CNN is tested on. If these two numbers are the same, it means training and testing are both using the same coarse-grain labels. If they are different, it means that CNN is trained first with fine-grain labels and then tested on the coarse-grain labels. We can see that training using fine-grain labels almost always improves testing accuracy compared to training using coarse-grain labels. In the case of CIFAR-100, fine-grain training provides negligible improvement on testing accuracy. We conjecture that this is due to the diminishing return when there are more coarse-grain labels, and we verify this hypothesis in Section VI-C. For the CIFAR-10 dataset, although the absolute value of improvement is 0.78%, considering the original testing accuracy is already near perfect, this further improvement is non-trivial. We also observe that CNN training accuracy gets better when using fine-grain labels. Above results indicate that fine-grain labels help improve both network optimization and generalization, and we will analyze the reasons in Sections IV-A and IV-B, respectively.

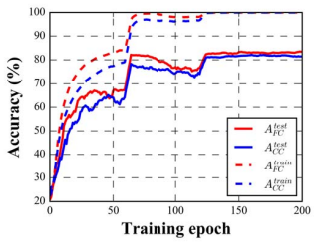
We further investigate how fine-grain labels affect training data efficiency. High training data efficiency means that (i) with the same amount of data, CNNs are able to learn and perform better, *i.e.*, achieve higher testing accuracy, and (ii) to achieve the same testing accuracy, CNNs require fewer training data. To this end, we randomly chose 20%, 40%, 60% or 80% of the entire training dataset to form four new training sets with increasing data amount (same proportion in each class so that the number of images within each class is still balanced), use the full testing dataset for testing, and compare the accuracy of fine-grain and coarse-grain training. Since we find that keeping the same number of epochs for reduced data amounts leads to fewer weight updates, we use proportionally more training epochs for less training data to keep the number of weight updates the same. In other words, when 20% of training data is used, we train for 5X epochs.

The results are depicted in Fig 2, where we show training and testing accuracy for both fine-grain and coarse-grain training, *i.e.*, A_{FC}^{train}, A_{FC}^{test}, A_{CC}^{train} and A_{CC}^{test}. We observe that training with fine-grain labels almost always improves testing accuracy. Especially in the case of (a), (d) and (e) of Figure 2 (CIFAR-10, ImageNet dog vs. cat, and ImageNet fruit vs. vegetable, respectively), we observe a significant improvement from the use of fine-grain labels: with less than 40% of the total training data, training with fine-grain labels is able to achieve even higher accuracy than using the full training dataset with coarse-grain labels. For CIFAR-100 animals (c), with only 80% of the total data amount, training with fine-grain labels is able to achieve comparable accuracy as using coarse-grain labels and full training dataset. Although fine-grain training has negligible improvement on testing accuracy with full dataset in case of CIFAR-100, when using fewer than 40% of the full dataset, fine-grain training still exhibits a clear advantage. This indicates that when availability of data is limited, having fine-grain labels can be helpful.

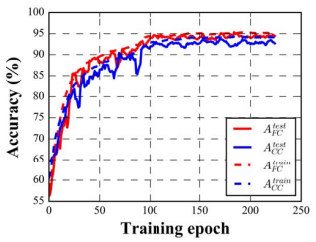


(a) CIFAR-10

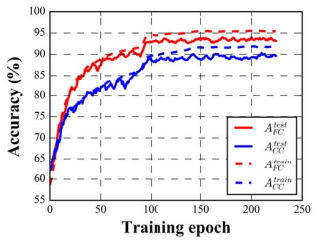
(b) CIFAR-100



(c) CIFAR-100 animals



(d) Dog vs. cat



(e) Fruit vs. vegetable

Fig. 3: Training (dotted) and testing (solid) accuracy curves for five datasets. CNNs trained with fine-grain labels are shown in red and those trained with coarse-grain labels are shown in blue. Experiments are conducted using five datasets: (a) CIFAR-10, (b) CIFAR-100, (c) CIFAR-100 animals, and two subsets of ImageNet datasets (d) dog vs. cat and (e) fruit vs. vegetable.

These experimental results show that training with fine-grain labels can help CNNs better utilize the available training data and can almost always improve CNN accuracy. One may think this counter-intuitive, as how could training on more classes require fewer training samples? The reasons are that i) fine-grain labels encourage the CNN to learn more features which helps with generalization, and ii) the test accuracy is eventually evaluated based on coarse-grain labels rather than fine-grain labels. We will discuss in detail on how fine-grain labels improve CNN performance in the following section. These results indicate two potential practical usage of this work: i) if we have sufficient human resources, we can improve CNN performance by re-labeling data with fine-grain labels, and ii) if we have limited human resources, in order to improve CNN performance, it can be more advantageous to re-labeling images with fine-grain labels rather than collecting more data.

IV. Optimization and Generalization

As we have discussed in Section III, training with fine-grain labels can improve not only testing but also training accuracy. This means that fine-grain labels help with both network *optimization* and *generalization*. In this section, we design and conduct extensive experiments on all datasets showing how both optimization and generalization are improved.

A. Optimization

In Figure 3, the dotted curves show the training accuracy of both fine-grain training, A_{FC}^{train}, and coarse-grain training, A_{CC}^{train}, for all datasets. The training accuracy is evaluated at the end of every epoch during the training phase by using the training dataset. We can see that training with fine-grain labels not only achieves higher training accuracy, but also converges faster as the red curve is always above the blue curve. The accuracy jumps are the results of reduced learning rate and are common phenomena in training neural networks [27].

Prior art investigating fine-grain labels on simple linear classifiers argues that the reason fine-grain labeling helps is the ability to learn piece-wise linear decision boundaries that can better approximate the true non-linear decision boundary [18]. That is, fine-grain training can have higher non-linearity compared to coarse-grain training due to increased parameters in the model. However, a further study [20] shows that in the case of non-linear classifiers, *e.g.*, RBF-kernel Support Vector Machine (SVM), fine-grain training no longer improves accuracy compared to using coarse-grain labels because the network itself has sufficient non-linearity to learn the non-linear decision boundary without the help of fine-grain labels. In the case of CNNs, we ask the question: is this piece-wise linear nature the reason for better training accuracy for fine-grain labels compared to coarse-grain training?

CNNs are already highly non-linear, so we conjecture that the answer is no. To evaluate this, we insert another fully-connected layer to a coarse-grain trained network right after the global pooling layer, so that compared to the original network, it can also achieve a piece-wise linear boundary on the high-level features. We train the new network end to end from scratch instead of pre-loading and freezing the weights of the preceeding layers, such that it fully utilizes all the degrees of freedoms of the model, possibly achieving higher training accuracy. We train this new network structure with coarse-grain labels, and compare the results with the baseline network trained with coarse- or fine-grain labels. We keep the training scheme for the slightly deeper network the same as that of the baseline network for a fair comparison with coarse- and fine-grain training.

Table IV shows our results. In the "CNN Arch" column, 'Extra layer' means that we add the fully-connected layer to the baseline CNN as described above. In the "Train Label" column, "F" and "C" indicate fine-grain and coarse-grain labels, respectively. The values in parentheses following each training and testing accuracy value are the improvement/degradation with respect to the training and testing accuracy of a baseline CNN trained with coarse-grain labels, respectively.

TABLE IV: Experiments on increasing CNN non-linearity and capacity under coarse-grain training. In "CNN Arch": 'Extra layer' means that we add the fully-connected layer to the baseline CNN to increase network non-linearity and capacity as described in Section IV-A. In "Train Label": "F" and "C" indicate fine-grain and coarse-grain labels, respectively. In the training and testing accuracy columns, the values indicated in the parentheses are the improvement/degradation with respect to the training and testing accuracy of a baseline CNN trained with coarse-grain labels, respectively.

Dataset	CNN Arch	Train Label	Training accuracy (%)	Testing accuracy (%)
CIFAR-10	Baseline CNN	F	100.0 (+0.1)	99.20 (+0.78)
	Extra layer	C	99.9 (+0.0)	98.50 (+0.08)
CIFAR-100	Baseline CNN	F	100.0 (+0.0)	85.05 (+0.01)
	Extra layer	C	100.0 (+0.0)	86.33 (+1.29)
CIFAR-100 animals	Baseline CNN	F	100.0 (+0.0)	83.44 (+2.02)
	Extra layer	C	100.0 (+0.0)	80.73 (-0.69)
ImageNet dog vs. cat	Baseline CNN	F	95.3 (+1.2)	94.87 (+2.19)
	Extra layer	C	93.8 (-0.3)	92.2 (-0.48)
ImageNet fruit vs. vege	Baseline CNN	F	95.4 (+3.6)	93.15 (+3.5)
	Extra layer	C	91.7 (-0.1)	89.67 (+0.02)

We can see that, compared to the baseline CNN trained with coarse-grain labels, adding one extra layer does not bring significant improvement in either optimization or generalization. In certain cases, the testing accuracy is degraded, in CIFAR-100 animals and ImageNet subset dog vs. cat, possibly due to the difficulty in optimizing a larger network.

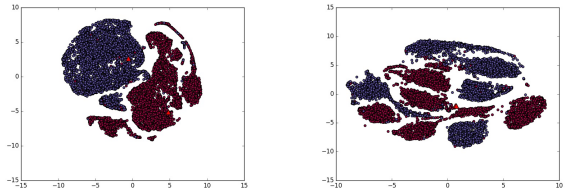
This means that simply adding non-linearity to coarse-grain training cannot match the training accuracy brought by fine-grain training. That is, the slightly higher non-linearity brought by fine-grain training is not the only reason for achieving higher training accuracy. Rather, it is more likely that fine-grain labels give more hints to the network about which features to learn. This is also supported by the experimental results in Section VI-B, where we randomly generate fine-grain labels for each coarse-grain class, and find out that fine-grain training does not optimize better than coarse-grain training.

B. Generalization

As shown in both Table I and Figure 3, training with fine-grain labels (vs. coarse-grain) achieves higher testing accuracy. This may partially be due to better network optimization, because under ImageNet subsets, fine-grain training improves both training and testing accuracy. However, in the cases of CIFAR-10, CIFAR-100 animals and CIFAR-100, even for the same training accuracy, testing accuracy for fine-grain trained CNNs is still higher than coarse-grain training. This indicates that fine-grain training delivers higher generalization capability.

Our intuition is that with fine-grain labels, the CNN is able to learn more features than training with coarse-grain labels. For example, suppose that all cat images in the training set have whiskers, while none of the dogs has whiskers. Then, as long as the network trained with coarse-grain labels learns this feature, it can produce 100% training accuracy with no need to learn any other features. This is a well known phenomenon in weakly-supervised learning, in which the network only learns the most discriminative features [28]. Then, in the testing set, if a cat image does not include whiskers, the network will make an incorrect prediction. However, with fine-grain labels, the network needs to learn more features (*e.g.*, ears, tails, etc.) to distinguish among different breeds of dogs and cats. These extra features learned through fine-grain labeling may help the network's performance on coarse-grain class classification on the testing set, *e.g.*, it now can tell if it is a cat through ears, tails, etc, even though it does not have whiskers.

Figure 4 shows the t-distributed Stochastic Neighbor Embedding (t-SNE) visualization [29] of all CIFAR-10 testing images with coarse-grain (a) and fine-grain training (b).



(a) t-SNE visualization of two coarse-grain classes in CIFAR-10 trained with two coarse-grain labels

(b) t-SNE visualization of two coarse-grain classes in CIFAR-10 trained with ten fine-grain labels

Fig. 4: t-SNE visualization of CIFAR-10 test set trained with coarse-grain labels vs. fine-grain labels. Data points shown in the same color belong to the same coarse-grain class.

Image features used for t-SNE visualization are the outputs of the second-to-last fully-connected layer, which is a technique commonly used to extract compact semantic representation of the raw input images [29]. These feature vectors are then transformed by the t-SNE technique [29] to a two-dimensional space for visualization. All data points are colored according to their ground-truth coarse-grain labels. We also show the position of means of each coarse-grain class as red triangles in the figures. We can see that, for both coarse-grain training, Figure 4a, and fine-grain training, Figure 4b, there is a noticeable margin between coarse-grain classes, and a decision boundary can be drawn to separate them. However, the network has to learn extra features to further separate the fine-grain classes within each coarse-grain class when trained with fine-grain labels (as shown in Figure 4b), while when trained with only coarse-grain labels, the data points are merged together as there is no need to separate them (as being visualized in Figure 4a).

An orthogonal method used for enhancing the variety of learned features and thereby increasing generalization ability is dropout [30]. Dropout randomly drops some of the features to encourage CNN to learn more various features. A possible question arises: by adding dropout to the network, will coarse-grain training reach the same testing accuracy as fine-grain training?

In cases of CIFAR-100 and CIFAR-100 animals, the original network already has dropout layers within each residual block with the optimal dropout rate 0.3 determined by experiments [26]. However, as shown in Table I, fine-grain training still outperforms coarse-grain training, under optimal dropout rate, which indicates that fine-grain training delivers benefits that dropout alone may not. We further conduct experiments on CIFAR-10 and ImageNet subsets, by adding a dropout layer between the global pooling layer and the fully-connected layer. The dropout rate is set as 0.3 in our following experiments.

Table V shows the experimental results of using the dropout technique. We observe that adding the dropout layer provides limited improvement in testing accuracy for coarse-grain training, and dropout for coarse-grain training still generates a noticeable margin when compared to the fine-grain training with or without dropout. This indicates that fine-grain labels can further improve CNN learning beyond what the traditional dropout technique can do. Actually, since fine-grain training and dropout are two orthogonal techniques, one can use both to further improve CNN performance. For example, in ImageNet subsets dog vs.cat and fruit vs. vegetable, combining the two techniques can able to push the testing accuracy to 95% and 93.86% from 92.68% and 89.65%, respectively.

TABLE V: Experiments on increasing CNN dropout rate. Values in "Dropout" column indicates dropout rates used. In "Train Label" column: "F" and "C" indicate fine-grain and coarse-grain labels, respectively.

Dataset	Dropout	Train Label	Training accuracy (%)	Testing accuracy (%)
CIFAR-10	0.3	F	100.0	99.10
	0.3	C	99.9	98.48
	0	F	100.0	99.20
	0	C	99.9	98.42
ImageNet dog vs. cat	0.3	F	94.8	95.00
	0.3	C	94.3	92.80
	0	F	95.3	94.87
	0	C	94.1	92.68
ImageNet fruit vs. vege	0.3	F	95.0	93.86
	0.3	C	91.7	89.93
	0	F	95.4	93.15
	0	C	91.8	89.65

V. Characterizing the effectiveness of fine-grain labels: Average Confusion Ratio

Fine-grain labels can improve both CNN optimization and generalization as shown by the experiments in the previous sections. However, we also note the varying benefit from fine-grain label usage under different datasets: fine-grain training sometimes improves testing accuracy by a considerably large margin, *e.g.*, 3.5% improvement in ImageNet fruit vs. vegetable, while sometimes the improvement is rather limited, *e.g.*, 0.01% improvement in CIFAR-100. Similar to the inter- and intra-cluster variance used in unsupervised clustering algorithms, *e.g.*, k-means [31], the benefit from fine-grain training may come from the relative difficulty of distinguishing between coarse-grain classes (inter-class confusion) vs. fine-grain classes (intra-class confusion). To quantify this, we propose the *Average Confusion Ratio* (ACR) metric to characterize the disparity within the coarse-grain and fine-grain classes, respectively, by using the confusion matrix shown in Fig 5. We denote the *confusion matrix* as $\mathbb{C}$, where $\mathbb{C}_{i,j}$ indicates the number of occurrences of confusing class i with class j and it can be obtained via counting those occurrences through the test dataset [32]. From the confusion matrix $\mathbb{C}$ for the fine-grain classes as in Figure 5, we can compute the ACR:

$$ACR = \frac{\sum_{(i,j)\in\overline{\mathbb{A}}} \mathbb{C}_{i,j}/|\overline{\mathbb{A}}|}{\sum_{(i,j)\in\mathbb{A}} \mathbb{C}_{i,j}/|\mathbb{A}|}, \quad (1)$$

where $\mathbb{A} = \{(i,j)|\mathbb{B}_{i,j} = 1\}$, $\overline{\mathbb{A}} = \{(i,j)|\mathbb{B}_{i,j} = 0\}$, and $\mathbb{B}$ is an indicator matrix with $\mathbb{B}_{i,j}$ indicating whether class i and j belong to the same coarse-grain class. Intuitively, ACR is the average inter-class confusion divided by average intra-class confusion, where inter- and intra-classes are considered from the perspective of coarse-grain classes.

ACR is correlated to the improvement produced by fine-grain training. We define the improvement from fine-grain training as the difference between the testing accuracy of a CNN trained with fine-grain labels and the testing accuracy of a CNN trained with coarse-grain labels, *i.e.*, $\Delta A^{test} = A^{test}_{FC} - A^{test}_{CC}$. Lower ACR means lower relative confusion across coarse-grain classes and hence higher distance between coarse-grain classes. This is a similar concept to high inter-cluster distance in clustering algorithms [20] [33], and those clusters are less prone to be mixed or confused. As a result, coarse-grain classes in this case are relatively easier to be separated even without the help of fine-grain labels, which leads to a low ΔA^{test} value, and vice versa.

To demonstrate how ACR can be an indicator of how much improvement fine-grain labels deliver in different datasets, we compute the ACR metric for all datasets in Table I and plot the relationship between ACR and ΔA^{test} in Figure 6. In general the data points in Figure 6 show higher ACR leading to higher ΔA^{test}, as expected. Other than the five datasets used throughout the paper, we also introduce two extra datasets: CIFAR-100-5 and CIFAR-100-15 with 5 and 15 coarse-grain classes, respectively. In the next section, we will detail these two datasets and the corresponding ACR metric under different settings of coarse- and fine-grain classes.

VI. Discussion

In this section, we further explore several scenarios in which the setting of coarse-grain and fine-grain labels change. More specifically, coarse-grain classes may vary due to the requirement of the application and the fine-grain labels may be

TABLE VI: Testing accuracy, trained with coarse-grain vs. fine-grain labels, of customized coarse-grain classes of CIFAR-10 dataset. Zero and one indicates which coarse-grain class each fine-grain class belongs to.

ID	Ratio	Classes										A_{CC}^{test} (%)	A_{FC}^{test} (%)	ΔA^{test} (%)
		plane	car	bird	cat	deer	dog	frog	horse	ship	truck			
(1)	6:4	0	0	1	1	1	1	1	1	0	0	98.42	99.20	+0.78
(2)		1	0	1	1	1	1	0	1	0	0	97.68	98.64	+0.96
(3)		0	0	0	1	1	1	1	1	1	0	96.95	98.02	+1.07
(4)		0	0	1	0	1	0	1	1	1	1	95.26	97.20	+1.94
(5)		0	0	1	0	0	1	1	1	1	1	93.44	96.22	+2.78
(6)	5:5	0	0	0	1	1	1	1	1	0	0	97.60	98.51	+0.91
(7)		0	0	1	0	1	1	1	1	0	0	95.90	97.59	+1.69
(8)		0	1	0	1	0	1	1	1	0	0	96.17	97.54	+1.37
(9)		0	0	1	0	0	1	0	1	1	1	94.15	96.28	+2.13
(10)		1	0	0	0	0	1	1	1	0	1	94.19	96.16	+1.97

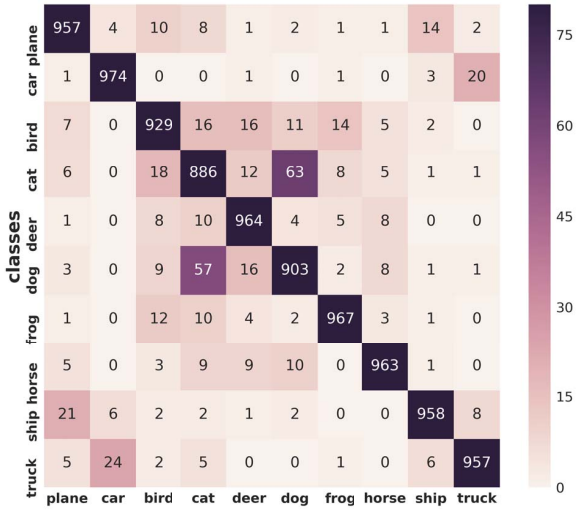


Fig. 5: Confusion matrix for ten classes of CIFAR-10 dataset.

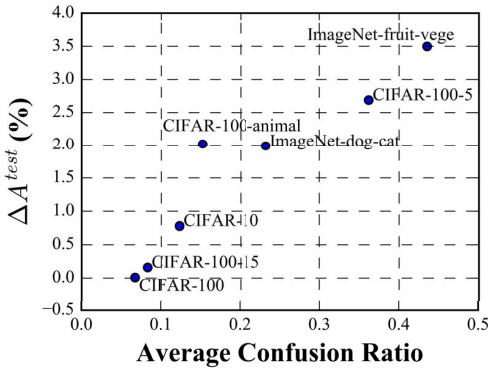


Fig. 6: ΔA^{test} vs. Average Confusion Ratio across different datasets.

noisy if it is generated via automatic unsupervised clustering algorithms. Again, we show that ACR is able to capture these characteristics and correctly reflect the effect of fine-grain training. We also investigate how increasing the number of coarse-grain classes impacts the improvement from using fine-grain labels, *i.e.*, ΔA^{test}.

TABLE VII: Testing accuracy trained with noisy fine-grain labels of CIFAR-10 dataset.

Randomness factor	A_{FC}^{test} (%)	ΔA^{test} (%)
0	99.20	0.78
0.01	98.94	0.52
0.03	98.55	0.13
0.1	98.12	-0.30
0.3	97.72	-0.70

In the following experiments, we use CIFAR-10 as an example to show how ACR can be used to characterize the effectiveness of fine-grain labels via the relationship between ACR and ΔA^{test} under different settings of coarse-grain and fine-grain labels. We use CIFAR-100 for the experiments on varying number of coarse-grain classes as it provides as many as 20 coarse-grain classes.

A. Customized Coarse-grain Classes

As mentioned, coarse-grain classes are the classification target, and as a result, the definition of coarse-grain classes is application dependent. For example, given an animal dataset, a task can be identifying cat vs. dog. vs horse, while another task can be separating standing animals from sitting and/or lying animals. Because of the diversity of applications, this mapping from fine-grain classes to coarse-grain classes can be drastically different. In this section, we conduct experiments to see how these customized coarse-grain classes affect the effectiveness of fine-grain labels and use ACR to characterize it.

A natural partition of CIFAR-10 dataset is the "animal" coarse-grain class vs. the "vehicle" coarse-grain class, where "animal" has six fine-grain classes and "vehicle" has four as depicted in Table II. To simulate various applications, we keep the 6:4 ratio of the two coarse-grain classes and randomly switch their fine-grain classes to create new coarse-grain classes. Rows (1) through (5) in Table VI show five experiments with different coarse-grain class definitions. We use two coarse-grain classes in this case (denoted by 0 and 1), and values in the table indicate which coarse-grain class (0 vs. 1) each fine-grain class (plane, car, etc.) belongs to. The last three columns of Table VI give the testing accuracy of the CNN trained with coarse-grain and fine-grain labels, respectively as well as the relative improvement of fine-grain training. We observe that fine-grain training achieves up

to 2.78% improvement and always outperforms coarse-grain training under various customized coarse-grain classes.

We further experiment with balanced coarse-grain classes. In the previous experiments, we have a 6:4 ratio for the number of fine-grain classes within each coarse-grain class. Now, we balance it to a 5:5 ratio, and similarly, we randomly switch fine-grain classes across the two coarse-grain classes. Rows (6) through (10) in Table VI show five experiments with different coarse-grain class definitions and a 5:5 ratio. Again, we can see that fine-grain training always produces higher testing accuracy than coarse-grain training.

As discussed before, higher ACR leads to higher ΔA^{test} and vice versa. We compute the ACR metric of all ten experiments and plot the relationship between ACR and ΔA^{test} in Figure 7a. Numbers on the data points are experiment IDs. We can see the trend of increasing benefit from fine-grain labeling, *i.e.*, increasing ΔA^{test}, when ACR gets larger. This demonstrate that ACR is a good indicator of how effective fine-grain labels are.

B. Noisy Fine-grain Classes

By using fine-grain labels, we are able to improve CNN performance. To obtain fine-grain labels, we can either ask human to label the images, or by automatically clustering every coarse-grain class into multiple fine-grain classes. The first approach is human-labor intensive but it is usually defined as the ground-truth, while the second approach is relatively cheap, but error-prone. In this part, we investigate how a noisy fine-grain label, *e.g.*, generated from a coarse-grain class by using unsupervised clustering methods, may affect effectiveness of training with fine-grain labels.

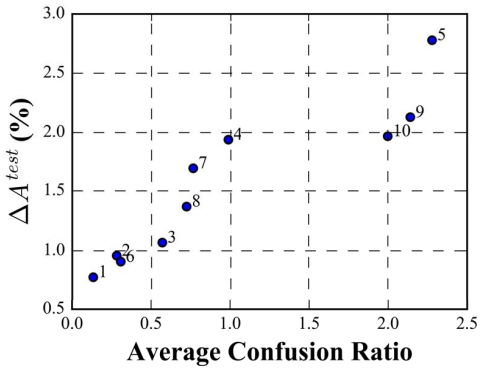
To this end, we keep the coarse-grain labels fixed and randomly change the fine-grain labels within each coarse-grain class to simulate the effect of noisy labeling. We tune the probability of randomizing the fine-grain labels, *i.e.*, randomness factor, to control the amount of noise in the experiments. Table VII shows the results under different randomness factors for CIFAR-10 dataset. We can see that with increased randomness factor, both A_{FC}^{test} and the improvement brought by fine-grain training, $\Delta A^{test} = A_{FC}^{test} - A_{CC}^{test}$, keep dropping. This means that training with highly incorrect fine-grain labels may actually hurt CNN performance. Therefore, how to automatically cluster each coarse-grain class into less-noisy fine-grain classes is an important direction to explore. We leave it for future work.

Again, we compute their ACR values and plot ΔA^{test} vs. ACR in Figure 7b. Numbers on the data points are randomness factors. With decreased randomness factor, confusion between fine-grain classes becomes less and ACR value increases. As expected, increased ACR value leads to increased ΔA^{test} as we can see in the figure.

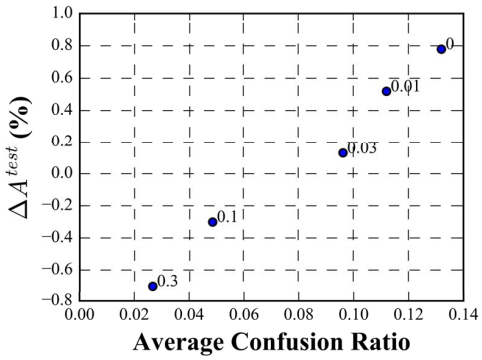
C. Varying number of coarse-grain classes

We further investigate how the number of coarse-grain classes affects the effectiveness of fine-grain labels. As we have discussed in Section IV.B, with fine-grain labels, the network is encouraged to learn more features than it needs when trained with only coarse-grain labels, and these extra features help in network generalization, *i.e.*, improving the testing accuracy. We conjecture that, to achieve high testing accuracy, a certain number of features needs to be learned by the network. Fine-grain labels help learn more features, however, with more coarse-grain classes, more features will be learned from only coarse-grain labels and hence it may be sufficient for classifying the test set, even without fine-grain labels. In other words, fine-grain labels bring diminishing returns when the number of coarse-grain classes increases.

To verify this, we experiment by varying the number of coarse-grain classes in the CIFAR-100 dataset and the results are shown in Table VIII. We can see that with increasing number of coarse-grain classes, *i.e.*, from 5, 10, 15 to 20, the benefit from fine-grain training, *i.e.*, ΔA^{test}, decreases, which is consistent with our expectation. We also compute their ACR values and show the relationship with ΔA^{test} in Figure 6. In the case of CIFAR-100 dataset, when the number of coarse-grain classes goes beyond 15, the improvement brought by fine-grain labeling is negligible. However, this threshold is



(a)



(b)

Fig. 7: (a): ΔA^{test} vs. Average Confusion Ratio computed from experiments of customized coarse-grain classes in Table VI . Number next to the data points is experiment ID. (b): ΔA^{test} vs. Average Confusion Ratio computed from experiments of noisy fine-grain classes in Table VII. Number next to the data points is randomness factor.

TABLE VIII: Testing accuracy, trained with coarse-grain vs. fine-grain labels, when varying number of coarse-grain classes in CIFAR-100 dataset. The coarse-grain class index follows the same order as in Table II. The values inside the parenthesis in column A_{FC}^{test} is ΔA^{test}, the calculated improvement of fine-grain training over coarse-grain training.

Coarse-grain class index																				Total	A_{CC}^{test} (%)	A_{FC}^{test} (%)
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19			
✓								✓			✓	✓				✓				5	80.53	83.22 (+2.69)
✓	✓						✓	✓			✓	✓	✓	✓	✓	✓				10	81.42	83.44 (+2.02)
✓	✓	✓		✓			✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	15	85.14	85.30 (+0.16)
✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	20	85.04	85.05 (+0.01)

application and dataset dependent and should be determined by experiments in a case-by-case manner.

VII. Conclusion

In this paper, we investigate the intriguing problem of how label granularity impacts CNN-based image classification. Our extensive experimentation shows that using fine-grain labels, rather than the target coarse-grain labels, can lead to higher accuracy and training data efficiency by improving both network optimization and generalization. Our results further suggest two practical applications: (i) with sufficient human resources, one can improve CNN accuracy by re-labeling the dataset with fine-grain labels, and (ii) with limited human resources, to improve CNN performance, rather than collecting more training data, one may instead collect fine-grain labels for the existing data. Furthermore, we propose a metric called *Average Confusion Ratio* (ACR) to quantify the accuracy gain from fine-grain labels, and demonstrate its effectiveness through experiments on various datasets and label settings.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, p. 436, 2015.

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 248–255.

[3] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," 2009.

[4] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[5] A. Sun and E.-P. Lim, "Hierarchical text classification and evaluation," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. IEEE, 2001, pp. 521–528.

[6] O. Dekel, J. Keshet, and Y. Singer, "Large margin hierarchical classification," in *Proceedings of the twenty-first international conference on Machine learning*. ACM, 2004, p. 27.

[7] W. Bi and J. T. Kwok, "Hierarchical multilabel classification with minimum bayes risk," in *Data Mining (ICDM), 2012 IEEE 12th International Conference on*. IEEE, 2012, pp. 101–110.

[8] Y. Song and D. Roth, "On dataless hierarchical text classification." in *AAAI*, vol. 7, 2014.

[9] S. Oh, "Top-k hierarchical classification." in *AAAI*, 2017, pp. 2450–2456.

[10] Y. Wang, Q. Hu, Y. Zhou, H. Zhao, Y. Qian, and J. Liang, "Local bayes risk minimization based stopping strategy for hierarchical classification," in *Data Mining (ICDM), 2017 IEEE International Conference on*. IEEE, 2017, pp. 515–524.

[11] H. Zhao, P. Zhu, P. Wang, and Q. Hu, "Hierarchical feature selection with recursive regularization," in *Proceedings of the 26th International Joint Conference on Artificial Intelligence*. AAAI Press, 2017, pp. 3483–3489.

[12] D. Wang, H. Huang, C. Lu, B.-S. Feng, L. Nie, G. Wen, and X.-L. Mao, "Supervised deep hashing for hierarchical labeled data," in *AAAI*, 2018.

[13] N. Cesa-Bianchi, C. Gentile, and L. Zaniboni, "Incremental algorithms for hierarchical classification," *Journal of Machine Learning Research*, vol. 7, no. Jan, pp. 31–54, 2006.

[14] J. Deng, J. Krause, A. C. Berg, and L. Fei-Fei, "Hedging your bets: Optimizing accuracy-specificity trade-offs in large scale visual recognition," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 3450–3457.

[15] T. Hoyoux, A. J. Rodríguez-Sánchez, and J. H. Piater, "Can computer vision problems benefit from structured hierarchical classification?" *Machine Vision and Applications*, vol. 27, no. 8, pp. 1299–1312, 2016.

[16] R. Cerri, R. C. Barros, and A. C. de Carvalho, "Hierarchical classification of gene ontology-based protein functions with neural networks," in *Neural Networks (IJCNN), 2015 International Joint Conference on*. IEEE, 2015, pp. 1–8.

[17] Y. Mo, S. D. Scott, and D. Downey, "Learning hierarchically decomposable concepts with active over-labeling," in *Data Mining (ICDM), 2016 IEEE 16th International Conference on*. IEEE, 2016, pp. 340–349.

[18] A. Hoffmann, R. Kwok, and P. Compton, "Using subclasses to improve classification learning," in *European Conference on Machine Learning*. Springer, 2001, pp. 203–213.

[19] Y. Luo, "Can subclasses help a multiclass learning problem?" in *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE, 2008, pp. 214–219.

[20] D. Fradkin, "Clustering inside classes improves performance of linear classifiers," in *Tools with Artificial Intelligence, 2008. ICTAI'08. 20th IEEE International Conference on*, vol. 2. IEEE, 2008, pp. 439–442.

[21] N. Ahmed and M. Campbell, "On estimating simple probabilistic discriminative models with subclasses," *Expert Systems with Applications*, vol. 39, no. 7, pp. 6659–6664, 2012.

[22] M. Ristin, J. Gall, M. Guillaumin, and L. Van Gool, "From categories to subcategories: large-scale image classification with partial class label refinement," in *Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on*. IEEE, 2015, pp. 231–239.

[23] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[24] G. A. Miller, "Wordnet: a lexical database for english," *Communications of the ACM*, vol. 38, no. 11, pp. 39–41, 1995.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *European Conference on Computer Vision*. Springer, 2016, pp. 630–645.

[26] S. Zagoruyko and N. Komodakis, "Wide residual networks," *arXiv preprint arXiv:1605.07146*, 2016.

[27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[28] H. Bilen and A. Vedaldi, "Weakly supervised deep detection networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2846–2854.

[29] L. v. d. Maaten and G. Hinton, "Visualizing data using t-sne," *Journal of machine learning research*, vol. 9, no. Nov, pp. 2579–2605, 2008.

[30] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[31] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[32] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information Processing & Management*, vol. 45, no. 4, pp. 427–437, 2009.

[33] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: a review," *ACM computing surveys (CSUR)*, vol. 31, no. 3, pp. 264–323, 1999.