Fine-grained Image Classification by Exploring Bipartite-Graph Labels

Feng Zhou
NEC Labs
www.f-zhou.com

Yuanqing Lin
NEC Labs
ylin@nec-labs.com

Abstract

Given a food image, can a fine-grained object recognition engine tell "which restaurant which dish" the food belongs to? Such ultra-fine grained image recognition is the key for many applications like search by images, but it is very challenging because it needs to discern subtle difference between classes while dealing with the scarcity of training data. Fortunately, the ultra-fine granularity naturally brings rich relationships among object classes. This paper proposes a novel approach to exploit the rich relationships through bipartite-graph labels (BGL). We show how to model BGL in an overall convolutional neural networks and the resulting system can be optimized through back-propagation. We also show that it is computationally efficient in inference thanks to the bipartite structure. To facilitate the study, we construct a new food benchmark dataset, which consists of 37,885 food images collected from 6 restaurants and totally 975 menus. Experimental results on this new food and three other datasets demonstrate BGL advances previous works in fine-grained object recognition. An online demo is available at http://www.f-zhou.com/fg_demo/.

1. Introduction

Fine-grained image classification concerns the task of distinguishing sub-ordinate categories of some base classes such as dogs [26, 43], birds [5, 8], flowers [1, 40], plants [32, 45], cars [30, 36, 48], food [3, 6, 38, 59], clothes [14], fonts [10] and furniture [4]. It differs from the base-class classification [15] in that the differences among object classes are more subtle, and thus it is more difficult to distinguish them. Yet fine-grained object classification is extremely useful because it is the key to a variety of challenging applications even difficult for human annotators.

While general image classification has achieved impressive success within the last few years [31, 49, 46, 21, 56, 22], it is still very challenging to recognize object classes with ultra-fine granularity. For instance, how to recognize each of the three food images shown in Fig. 1 into *which restaurant which dish*? The challenge arises in two major aspects. First, different classes may be visually similar, *e.g.*, the Mapo Tofu dish from Restaurant A (the 1^{st} image in Fig. 1) looks very similar to the one from Restaurant B (the 3^{rd} image in Fig. 1). Second, each class may not have enough training images because of the ultra-fine granularity. In such setting, how to share information between similar classes while maintaining strong discriminativeness becomes more critical.

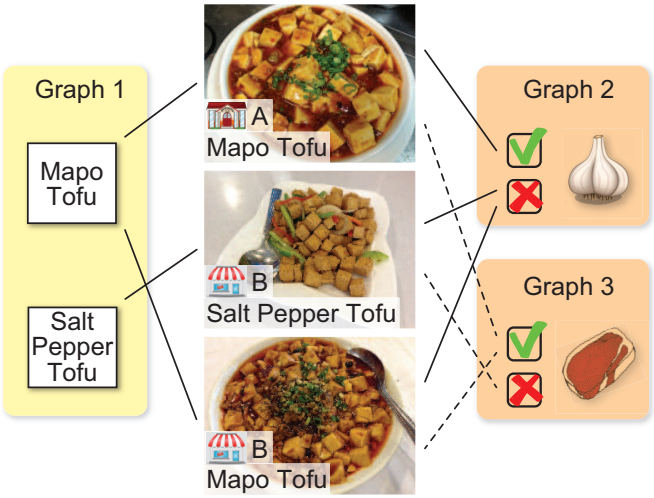


Figure 1. Illustration of three ultra-fine grained classes (middle), Mapo Tofu of Restaurant A, Salt Pepper Tofu of Restaurant B, and Mapo Tofu of Restaurant B. Their relationships can be modeled through three bipartite graphs, fine-grained classes vs. general food dishes (left) and fine-grained classes vs. two ingredients (right). This paper shows how to incorporate the rich bipartite-graph labels (BGL) into convolutional neural network training to improve recognition accuracy.

To that end, we propose a novel approach using bipartite-graph labels (BGL) that models the rich relationships among the ultra-fine grained classes. In the example of Fig. 1, the 1^{st} and 3^{rd} images are both Mapo Tofu dishes; and they share some ingredients with the 2^{nd} one. Such class relationships can be modeled in three bipartite graphs. This paper shows how to incorporate the class bipartite graphs into CNN and learn the optimal classifiers through overall back-propagation.

Using BGL has several advantages: (1) BGL imposes additional constraints to regularize CNN training, thereby largely reducing the possibility of overfitting when only a small amount of training data is available. (2) Knowing classes that belong to the same coarse category or share some common attributes can allow us to borrow some knowledge from relevant classes. (3) The supervised feature learning through a global back-propagation allows learning discriminative features for capturing subtle differences between similar classes. (4) By constraining the structure to bipartite graphs, BGL prevents the exponential explosion from enumerating all possible states in inference.

This work is in parallel to the existing big body of fine-grained image classification research, which has focused on devising more discriminative feature by aligning object poses [17, 20, 62] and filtering out background through object segmentation [42]. The techniques developed in this work can be combined with the ones in those existing research works to achieve better fine-grained image recognition performance.

To facilitate the study, we built an ultra-fine grained image recognition benchmark, which consists of 37,885 food training images collected directly from 6 restaurants with totally 975 menus. Our results show that the proposed BGL approach produces significantly better recognition accuracy compared to the powerful GoogLeNet [49]. We also test the BGL approach on some existing fine-grained benchmark datasets. We observe the benefit of BGL modeling as well although the improvement is less significant because of the less rich class relationships.

2. Previous work

This section reviews the related work on fine-grained image classification, and structural label learning.

2.1. Fine-grained image classification

Fine-grained image classification needs to discern subtle differences among similar classes. The majority of existing approaches have thus been focusing on localizing and describing discriminative object parts in fine-grained domains. Various pose-normalization pooling strategies combined with 2D or 3D geometry have been proposed for recognizing birds [7, 17, 20, 62, 63, 64], dogs [37] and cars [28, 30]. The main drawback of these approaches is that part annotations are significantly more challenging to collect than image labels. Instead, a variety of methods have been developed towards the goal of finding object parts in an unsupervised or semi-supervised fashion. Krause *et al*. [29] combined alignment with co-segmentation to generate parts without annotations. Lin *et al*. [35] proposed an architecture that uses two separate CNN feature extractors to model the appearance due to where the parts are and what the parts look like. Jaderberg *et al*. [23] introduced spatial transformer, a new differentiable module that can be inserted into existing convolutional architectures to spatially transform feature maps without any extra training supervision. In parallel to above efforts, our approach focuses on exploiting rich class relationships and is applicable to generic fine-grained objects even when they do not own learnable structures (*e.g*., food dishes).

To provide good features for recognition, another prominent direction is to adopt detection and segmentation methods as an initial step and to filter out noise and clutter in background. For instance, Parkhi *et al*. [42, 43] proposed to detect some specific part (*e.g*., cat's head) and then performed a full-object segmentation through propagation. In another similar work, Angelova and Zhu [1] further renormalized objects after segmentation to improve the recognition performance. However, better feature through segmentation always comes with computational cost as segmentation is often computationally expensive.

Recently, many other advances lead to improved results. For instance, Wang *et al*. [53] and Qian *et al*. [44] showed a more discriminative similarity can be learned through deep ranking and metric learning respectively. Xie *et al*. [57] proposed a novel data augmentation approach to better regularize the fine-grained problems. Deng *et al*. [13], Branson *et al*. [8] and Wilber *et al*. [55] developed hybrid systems to introduce human in the loop for localizing discriminative regions for computing features. Focusing on different goal on exploring label structures, our method can be potentially combined with the above methods to further improve the performance.

2.2. Structural label learning

While most existing works focus on single-label classification problem, it is more natural to describe real world images with multiple labels like tags or attributes. According to the assumptions on label structures, previous work on structural label learning can be roughly categorized as learning binary, relative or hierarchical attributes.

Much of prior work focuses on learning binary attributes that indicate the presence of a certain property in an image or not. For instance, previous works have shown the benefit of learning binary attributes for face verification [33], texture recognition [19], clothing searching [14], and zero-shot learning [34]. However, binary attributes are restrictive when the description of certain object property is continuous or ambiguous.

To address the limitation of binary attributes, comparing attributes has gained attention in the last years. The relative-attribute approach [41] learns a global linear ranking function for each attribute, offering a semantically richer way to describe and compare objects. While a promising direction, a global ranking of attributes tends to fail when facing fine-grained visual comparisons. Yu and Grauman [60] fixed

this issue by learning local functions that tailor the comparisons to neighborhood statistics of the data. Recently, Yu and Grauman [61] developed a Bayesian strategy to infer when images are indistinguishable for a given attribute.

Our method falls into the third category where the relation between the fine-grained labels and attributes is modeled in a hierarchical manner. In the past few years, extensive research has been devoted to learning a hierarchical structure over classes (see [51] for a survey). Previous works have shown the benefits of leveraging the semantic class hierarchy using either unsupervised [2, 50] or supervised [11, 18, 39] methods. Our work differs from previous works in the CNN-based framework and the setting focusing on multi-labeled object. The most similar works to ours are [47] and [12], which show the advantages of exploring the tree-like hierarchy in small-scale (*e.g.*, CIFAR-100) and graph-like structure in large-scale (*e.g.*, ImageNet) categories respectively. Compared to [47], our method is able to handle more general structure (*e.g.*, attributes) among fine-grained labels. Unlike [12] relying on approximated inference, our method allows for efficient exact inference by modeling the label dependence as a star-like combination of bipartite graphs. In addition, we explore the hierarchical regularization on the last fully connected layer, which could further reduce the possibility of overfitting.

Our work is also related to previous methods in multi-task learning (MTL) [9]. To better learn multiple correlated subtasks, previous work have explored various ideas including, sharing hidden nodes in neural networks [9], placing a common prior in hierarchical Bayesian models [58] and structured regularization in kernel methods [16], among others. Our method differs from the MTL methods mainly in the setting of multi-label setting for single objects.

3. CNN with Bipartite-Graph Labels

This section describes the proposed BGL method in a common multi-class convolutional neural network (CNN) framework, which is compatible to most popular architectures like, AlexNet [31], GoogLeNet [49] and VGGNet [46]. Our approach modifies their softmax layer as well as the last fully connected layer to explicitly model BGLs. The resulting CNN is optimized by a global back-propagation.

3.1. Background

Suppose that we are given (see notation[1]) a set of n images $\mathcal{X} = \{(\mathbf{x}, y), \cdots\}$ for training, where each image $\mathbf{x}$ is annotated with one of k fine-grained labels, $y = 1, \cdots, k$. Let $\mathbf{x} \in \mathbb{R}^d$ denote the input feature of the last fully-connected layer, which generates k scores $\mathbf{f} \in \mathbb{R}^k$ through a linear function $\mathbf{f} = \mathbf{W}^T\mathbf{x}$ defined by the parameters $\mathbf{W} \in \mathbb{R}^{d\times k}$. In a nutshell, the last layer of CNN is to minimize the negative log-likelihood over the training data, *i.e.*,

$$\min_{\mathbf{W}} \sum_{(\mathbf{x},y)\in\mathcal{X}} -\log P(y|\mathbf{x}, \mathbf{W}), \tag{1}$$

where the softmax score,

$$P(i|\mathbf{x}, \mathbf{W}) = \frac{e^{f_i}}{\sum_{j=1}^{k} e^{f_j}} \doteq p_i, \tag{2}$$

encodes the posterior probability of image $\mathbf{x}$ being classified as the i^{th} fine-grained class.

3.2. Objective function with BGL modeling

Despite the great improvements achieved on base-class recognition in the last few years, recognizing object classes in ultra-fine granularity like the example shown in Fig. 1 is still challenging mainly for two reasons. First, unlike general recognition task, the amount of labels with ultra-fine granularity is often limited. The training of a high-capacity CNN model is thus more prone to overfitting. Second, it is difficult to learn discriminative features to differentiate between similar fine-grained classes in the presence of large within-class difference.

To address these difficulties, we propose bipartite-graph labels (BGL) to jointly model the fine-grained classes with pre-defined *coarse classes*. Generally speaking, the choices of coarse classes can be any grouping of fine-grained classes. Typical examples include bigger classes, attributes or tags. For instance, Fig. 1 shows three types of coarse classes defined on the fine-grained Tofu dishes (middle). In the first case (Graph 1 in Fig. 1), the coarse classes are two general Tofu food classes by neglecting the restaurant tags. In the second and third cases (Graph 2 and 3 in Fig. 1), the coarse classes are binary attributes according to the presence of some ingredient in the dishes. Compared to the original softmax loss (Eq. 2) defined only on fine-grained labels, the introduction of coarse classes in BGL has three benefits: (1) New coarse classes bring in additional supervised information so that the fine-grained classes connected to the same coarse class can share training data with each other. (2) All fine-grained classes are implicitly ranked according to the connection with coarse classes. For instance, Toyota Camry 2014 and Toyota Camry 2015 are much closer to each other compared to Honda Accord 2015. This hierarchical ranking guides CNN training to learn more discriminative features to capture subtle difference between fine-grained classes. (3) Compared to image-level labels (*e.g.*, image attribute, bounding box, segmentation mask) that are expensive to obtain, it is relatively cheaper and more natural

[1]Bold capital letters denote a matrix $\mathbf{X}$, bold lower-case letters a column vector $\mathbf{x}$. All non-bold letters represent scalars. $\mathbf{x}_i$ represents the i^{th} column of the matrix $\mathbf{X}$. x_{ij} denotes the scalar in the i^{th} row and j^{th} column of the matrix $\mathbf{X}$. $1_{[i=j]}$ is an indicator function and its value equals to 1 if $i = j$ and 0 otherwise.

to define coarse classes over fine-grained labels. This fact endows BGL the board applicability in real scenario.

Given m types of coarse classes, where each type j contains k_j coarse classes, BGL models their relations with the k fine-grained classes as m bipartite graphs grouped in a star-like structure. Take Fig. 1 for instance, where the three types of coarse classes form three separated bipartite graphs with the fine-grained Tofu dishes, and there is no direct connection among the three types of coarse classes. For each graph of coarse type j, we encode its bipartite structure in a binary association matrix $\mathbf{G}_j \in \{0,1\}^{k\times k_j}$, whose element $g^j_{ic_j} = 1$ if the i^{th} fine-grained label is connected with coarse label c_j. As it will become clear later, this star-like composition of bipartite graphs enables BGL to perform exact inference as opposed to the use of other general label graphs (*e.g.*, [12]).

To generate the scores $\mathbf{f}_j = \mathbf{W}_j^T\mathbf{x} \in \mathbb{R}^{k_j}$ for coarse classes of type j, we augment the last fully-connected layer with m additional variables, $\{\mathbf{W}_j\}_j$, where $\mathbf{W}_j \in \mathbb{R}^{d\times k_j}$. Given an input image $\mathbf{x}$ of i^{th} fine-gained class, BGL models its joint probability with any m coarse labels $\{c_j\}_j$ as,

$$P(i,\{c_j\}_j|\mathbf{x},\mathbf{W},\{\mathbf{W}_j\}_j) = \frac{1}{z}e^{f_i}\prod_{j=1}^{m} g^j_{ic_j}e^{f^j_{c_j}},$$

where z is the partition function computed as:

$$z = \sum_{i=1}^{k} e^{f_i}\prod_{j=1}^{m}\sum_{c_j=1}^{k_j} g^j_{ic_j}e^{f^j_{c_j}}.$$

At first glance, computing z is infeasible in practice. Because of the bipartite structure of the label graph, however, we could denote the non-zero element in i^{th} row of $\mathbf{G}_j$ as $\phi^j_i = c_j$ where $g^j_{ic_j} = 1$. With this auxiliary function, the computation of z can be simplified as

$$z = \sum_{i=1}^{k} e^{f_i}\prod_{j=1}^{m} e^{f^j_{\phi^j_i}}. \tag{3}$$

Compared to general CRF-based methods (*e.g.*, [12]) with exponential number of possible states, the complexity $O(km)$ of computing z in BGL through Eq. 3 scales linearly with respect to the number of fine-grained classes (k) as well as the type of coarse labels (m). Given z, the marginal posterior probability over fine-grained and coarse labels can be computed as:

$$P(i|\mathbf{x},\mathbf{W},\{\mathbf{W}_j\}_j) = \frac{1}{z}e^{f_i}\prod_{j=1}^{m} e^{f^j_{\phi^j_i}} \doteq p_i,$$

$$P(c_j|\mathbf{x},\mathbf{W},\{\mathbf{W}_l\}_l) = \frac{1}{z}\sum_{i=1}^{k} g^j_{ic_j}e^{f_i}\prod_{l=1}^{m} e^{f^l_{\phi^l_i}} \doteq p^j_{c_j}.$$

As discussed before, one of the difficulties in training CNN is the possibility of overfitting. One common solution is to add a l_2 weight decay term, which is equivalent to sampling the columns of $\mathbf{W}$ from a Gaussian prior. Given the connection among fine-grained and coarse classes, BGL provides another natural hierarchical prior for sampling the weights:

$$P(\mathbf{W},\{\mathbf{W}_j\}_j) = \prod_{i=1}^{k}\prod_{j=1}^{m}\prod_{c_j=1}^{k_j} e^{-\frac{\lambda}{2}g^j_{ic_j}\|\mathbf{w}_i-\mathbf{w}^j_{c_j}\|^2} \doteq p_w.$$

This prior expects $\mathbf{w}_i$ and $\mathbf{w}^j_{c_j}$ have small distance if i^{th} fine-grained label is connected to coarse class c_j of type j. Notice that this idea is a generalized version of the one proposed in [47]. However, [47] only discussed a special type of coarse label (big class), while BGL can handle much more general coarse labels such as multiple attributes.

In summary, given the training data $\mathcal{X}$ and the graph label defined by $\{\mathbf{G}_j\}_j$, the last layer of CNN with BGL aims to minimize the joint negative log-likelihood with proper regularization over the weights:

$$\min_{\mathbf{W},\{\mathbf{W}_j\}_j}\sum_{(\mathbf{x},y)\in\mathcal{X}}\Big(-\log p_y - \sum_{j=1}^{m}\log p^j_{\phi^j_y}\Big) - \log p_w. \tag{4}$$

3.3. Optimization

We optimized BGL using the standard back-propagation with mini-batch stochastic gradient descent. The gradients for each parameter can be computed[2] all in closed-form:

$$\frac{\partial \log p_y}{\partial f_i} = 1_{[i=y]} - p_i,\ \frac{\partial \log p_y}{\partial f^j_{c_j}} = 1_{[g^j_{yc_j}=1]} - p^j_{c_j},$$

$$\frac{\partial \log p^j_{\phi^j_y}}{\partial f_i} = \frac{p_i}{p^j_{\phi^j_y}}1_{[g^j_{i\phi^j_y}=1]} - p_i,\ \frac{\partial \log p^j_{\phi^j_y}}{\partial f^j_{c_j}} = 1_{[c_j=\phi^j_y]} - p^j_{c_j},$$

$$\frac{\partial \log p^j_{\phi^j_y}}{\partial f^l_{c_l}} = \sum_{i=1}^{k} g^j_{i\phi^j_y}g^l_{il}\frac{p_i}{p^j_{\phi^j_y}} - p^l_{c_l}, l\neq j, \tag{5}$$

$$\frac{\partial \log p_w}{\partial \mathbf{w}_i} = -\lambda\sum_{j=1}^{m}\sum_{c_j=1}^{k_j} g^j_{ic_j}(\mathbf{w}_i - \mathbf{w}^j_{c_j}), \tag{6}$$

$$\frac{\partial \log p_w}{\partial \mathbf{w}^j_{c_j}} = -\lambda\sum_{i=1}^{k} g^j_{ic_j}(\mathbf{w}^j_{c_j} - \mathbf{w}_i). \tag{7}$$

Here we briefly discuss some implementation issues. (1) Computing $p_i/p^j_{\phi^j_y}$ by independently calculating p_i and $p^j_{\phi^j_y}$ is not numerically stable because $p^j_{\phi^j_y}$ could be very small. It is better to jointly normalize the two terms first. (2) Directly computing Eq. 5 has a quadratic complexity with respect to the number of coarse classes. But it can be reduced to linear because most computations are redundant. See supplementary materials for more details. (3) So far (Fig. 2b) we assume the same feature $\mathbf{x}$ is used for computing both the fine-grained $\mathbf{f} = \mathbf{W}^T\mathbf{x}$ and coarse scores $\mathbf{f}_j = \mathbf{W}_j^T\mathbf{x}$. In

[2] See supplementary materials for detailed derivation.

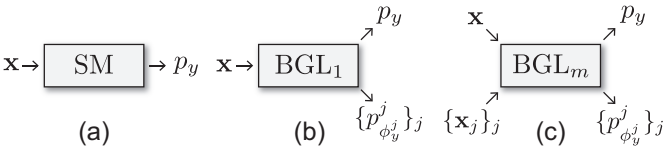


Figure 2. Comparison of output layers. (a) Softmax (SM). (b) BGL with one feature (BGL$_1$). (c) BGL with multiple features (BGL$_m$).

fact, BGL can naturally combine multiple CNNs as shown in Fig. 2c. This allows the model to learn different low-level features $\mathbf{x}_j$ for coarse labels $\mathbf{f}_j = \mathbf{W}_j^T \mathbf{x}_j$.

4. Experiments

This section evaluates BGL's performance for fine-grained object recognition on four benchmark datasets. The first two datasets, Stanford cars [30], CUB-200-2011 [52], contains fine-grained categories sharing one level of coarse attributes. The last two datasets, Car-333 [57] and a new Food-975 dataset, consist of ultra-fine grained classes and richer class relationships. We wish the proposed BGL approach would be able to improve classification accuracy even with simple BGLs and significantly improve performance when richer class relationships are present.

BGL was implemented on the off-the-shelf Caffe [24] platform. We test on three popular CNN frameworks, AlexNet (**AN**) [31], GoogLeNet (**GN**) [49], and VGGNet (**VGG**) [46]. For each of them, we compared three settings: **SM** with the traditional softmax loss; **BGL**$_1$ by modifying the softmax layer and the last fully connected layer as the proposed BGL approach; and **BGL**$_m$ by combining multiple networks[3] that generate different features for fine-grained and coarse classes.

Our models were trained for 100 epochs on a single NVIDIA K40 GPU. We adopted the default hyper-parameters as used by Caffe. In all experiments, we fine-tuned from pre-trained ImageNet model as in [25] because it always achieved better result. During training, we down-sampled the images to a fixed 256-by-256 resolution, from which we randomly cropped 224-by-224 patches for training. We also did their horizontal reflection for further data augmentation. During testing, we evaluated the top-1 accuracy using two cropping strategies: (1) single-view (**SV**) by cropping the center 224-by-224 patch of the testing image, and (2) multi-view (**MV**) by averaging the center, 4 corners and their mirrored versions. In the first three datasets, we evaluated our methods using two protocols, without (**w/o. BBox**) and with (**w/. BBox**) the use of ground-truth bounding box to crop out the object both at training and testing.

[3]Limited by the GPU memory, we always combined two networks in BGL$_m$, one for fine-grained classes and the other for all coarse labels.

4.1. Stanford car dataset

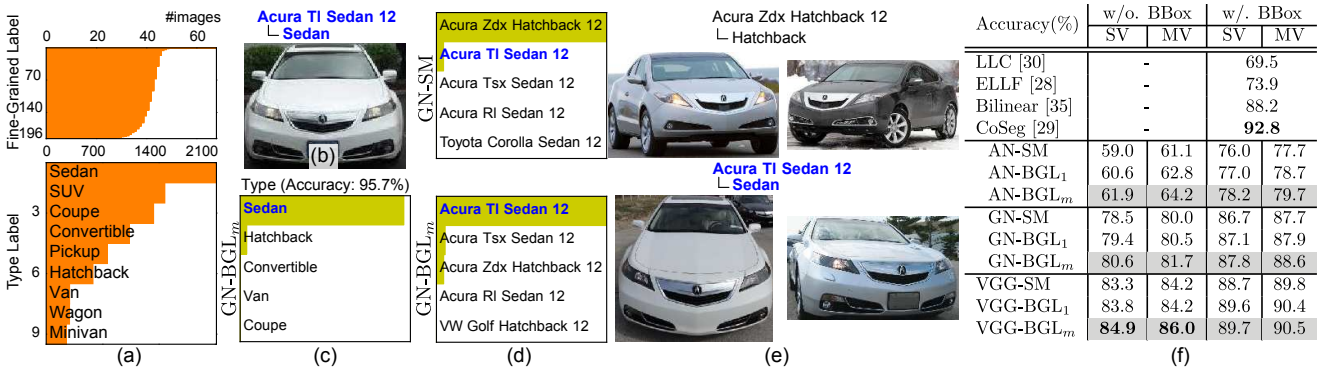
The first experiment validates our approach on the Stanford car dataset [30], which contains 16,185 images of 196 car categories. We adopted the same 50-50 split as in [30] by dividing the data into 8,144 images for training and the rest for testing. Each category is typically at the level of maker, model and year, *e.g.*, Audi A5 Coupe 12. Following [27], we manually assigned each fine-grained label to one of 9 coarse body types. Fig. 3a summarizes the distribution of training images for fine-grained and coarse labels.

Fig. 3b-e compare between the original GN-SM and the proposed GN-BGL$_m$ on a testing example. GN-SM confused the ground-truth hatchback model (Acura Zdx Hatchback) with a very similar sedan one (Acura TI Sedan). By jointly modeling with body type (Fig. 3c), however, GN-BGL$_m$ was able to predict the correct fine-grained label. Fig. 3f further compares BGL with several previous works using different CNN architectures. Without using CNN, the best published result, 69.5%, was achieved in [30] by using a traditional LLC-based representation. This number was beaten by ELLF [28] by learning more discriminative features using CNN. The bilinear model [35] recently obtained 88.2% by combining two VGG nets. By exploring the label dependency, the proposed BGL further improves all the three CNN architectures using either single-view (SV) or multi-view (MV) cropping. This consistent improvement demonstrates BGLs advantage in modeling structure among fine-grained labels. Our method of VGG-BGL$_m$ beats all previous works except [29], which leveraged the part information in an unsupervised way. However, we believe BGL can be combined with [29] to achieve better performance. In addition, BGL has the advantage of predicting coarse label. For instance, GN-BGL$_m$ achieved 95.7% in predicting the type of a car image.

4.2. CUB-200-2011 dataset

In the second experiment, we test our method on CUB-200-2011 [8], which is generally considered the most competitive dataset within fine-grained recognition. CUB-200-2011 contains 11,788 images of 200 bird species. We used the provided train/test split and reported results in terms of classification accuracy. To get the label hierarchy, we adopted the annotated 312 visual attributes associated with the dataset. See Fig. 4a for the distribution of the fine-grained class and attributes. These attributes are divided into 28 groups, each of which has about 10 choices. According to the provided confidence score, we assigned each attribute group with the most confident choice for each bird specie. For instance, Fig. 4b shows an example of Scarlet Tanager, most of which own black-color and pointed-shape wings.

Fig. 4d compares the outputs between the baseline GN-SM and the proposed GN-BGL$_m$. GN-SM predicted the



Accuracy(%)	w/o. BBox		w/. BBox	
	SV	MV	SV	MV
LLC [30]	-		69.5	
ELLF [28]	-		73.9	
Bilinear [35]	-		88.2	
CoSeg [29]	-		**92.8**	
AN-SM	59.0	61.1	76.0	77.7
AN-BGL$_1$	60.6	62.8	77.0	78.7
AN-BGL$_m$	61.9	64.2	78.2	79.7
GN-SM	78.5	80.0	86.7	87.7
GN-BGL$_1$	79.4	80.5	87.1	87.9
GN-BGL$_m$	80.6	81.7	87.8	88.6
VGG-SM	83.3	84.2	88.7	89.8
VGG-BGL$_1$	83.8	84.2	89.6	90.4
VGG-BGL$_m$	**84.9**	**86.0**	89.7	90.5

(f)

Figure 3. Comparison on the Stanford car dataset. (a) The number of training images for each fine-grained (top) and coarse (bottom) category. (b) An example of testing images. (c) Type predicted by GN-BGL$_m$. (d) Top-5 predictions generated by GN-SM approach (top) and the proposed BGL approach GN-BGL respectively. (e) Similar training exemplars according to the input feature $\mathbf{x}$. (f) Accuracy.

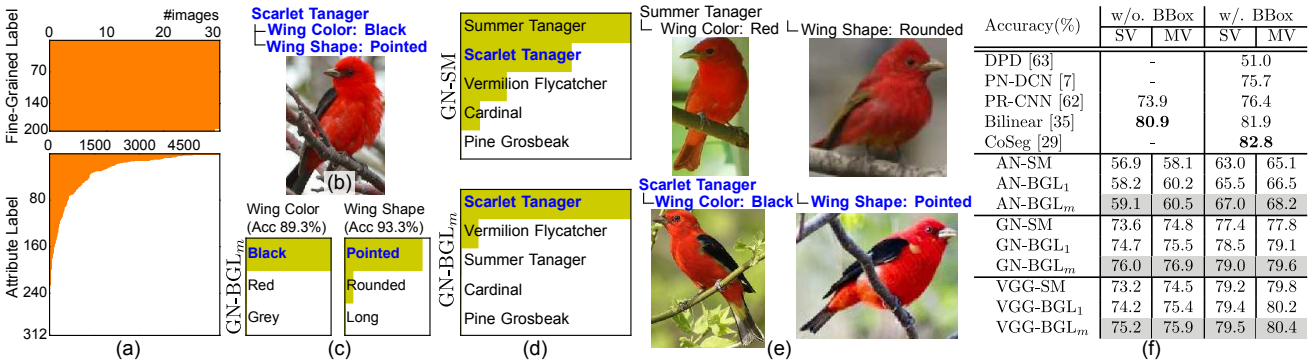
bird in Fig. 4b as a Summer Tanager, which is very close to the actual class of Scarlet Tanager. By more closely comparing the color and shape of the wings (Fig. 4c), however, GN-BGL$_m$ decreased its possibility of being Summer Tanager. This challenging example demonstrates the advantages of BGL in capturing the subtle appearance difference between similar bird species. Fig. 4f further compares BGL with state-of-the-arts approaches in different settings. Early works such as DPD [63] performed poorly when only using hand-crafted features. The integration with CNN-based pose alignment techniques in PN-DCN [7] and PR-CNN [62] greatly improve the overall performance. From the experiments, we observed using BGL modules can consistently improved AN-SM, GN-SM and VGG-SM. Without any pose alignment steps, our method GN-BGL$_m$ obtained 76.9% without the use of bounding box, improving the recent part-based method [62] by 3%. In addition, GN-BGL$_m$ achieved 89.3% and 93.3% accuracy on predicting attributes of wing color and shape. However, our method still performed worse than the latest methods [35] and [29], which show the significant advantage by exploring part information for bird recognition. It is worth to emphasize that BGL improves the last fully connected and loss layer for attribute learning, while [35] and [29] focus on integrating object part information into convolutional layers. Therefore, it is possible to combine these two orthogonal efforts to further improve the overall performance.

4.3. Car-333 dataset

In the third experiment, we test our method on the recently introduced Car-333 dataset [57], which contains $157,023$ training images and $7,840$ testing images. Compared to the Stanford car dataset, the images in Car-333 were end-user photos and thus more naturally photographed. Each of the 333 labels is composed by maker, model and year range. Notice that two cars of the same model but manufactured in different year ranges are considered different classes. To test BGL, we generated two sets of coarse labels: 10 "type" coarse labels manually defined according to the geometric shape of each car model and 140 "model" coarse labels by aggregating year range labels. Please refer to Fig. 5 for the distribution of the training images at each label level. The bounding box of each image was generated by Regionlets [54], the state-of-the-art object detection method.

Fig. 5b shows a testing example of Ford Ranchero 70-72. GN-SM recognized it as Ford Torino 70-71 because of the similar training exemplars as shown in the top of Fig. 5e. However, these two confused classes can be well separated by jointly modeling the type and model probability in GN-BGL$_m$. Fig. 5f summarizes the performance of our method using different CNN architectures. The best published result on this dataset was 83.6% achieved by HAR [57], where the authors augmented the original training data with an additional car dataset labeled view point information. We test BGL with three combinations of the coarse labels: using either model or type, and using model and type jointly. In particular, BGL gains much more improvements using the 140 model coarse labels than the 10 type labels. This is because the images of the cars of the same "model" are more similar than the ones in the same "type" and it defines richer relationships among fine-grained classes. Nevertheless, BGL can still get benefit from putting the "type" labels on top of the "model" labels to form a three-level label hierarchy. Finally, GN-BGL$_m$ significantly improved the performance of GN-SM from 79.8% to 86.4% without the use of bounding box. For more result on AN and VGG, please refer to the supplementary material.

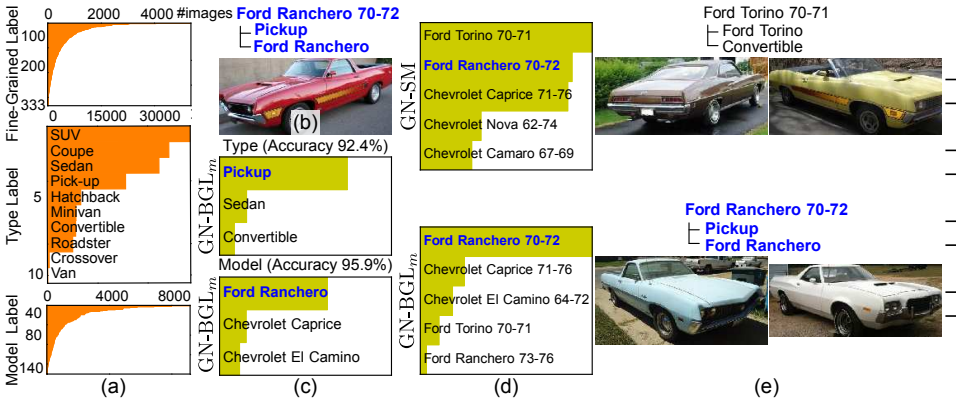
Since the Car-333 dataset is now big enough, we provide more comparisons on the size of training data and time cost between GN-SM and GN-BGL. Fig. 6a evaluates the performance with respect to the different amounts of training data. BGL is able to provide good improvement especially when training data is relatively small. This is because the



Accuracy(%)	w/o. BBox		w/. BBox	
	SV	MV	SV	MV
DPD [63]	-		51.0	
PN-DCN [7]	-		75.7	
PR-CNN [62]	73.9		76.4	
Bilinear [35]	**80.9**		81.9	
CoSeg [29]	-		**82.8**	
AN-SM	56.9	58.1	63.0	65.1
AN-BGL$_1$	58.2	60.2	65.5	66.5
AN-BGL$_m$	59.1	60.5	67.0	68.2
GN-SM	73.6	74.8	77.4	77.8
GN-BGL$_1$	74.7	75.5	78.5	79.1
GN-BGL$_m$	76.0	76.9	79.0	79.6
VGG-SM	73.2	74.5	79.2	79.8
VGG-BGL$_1$	74.2	75.4	79.4	80.2
VGG-BGL$_m$	75.2	75.9	79.5	80.4

(f)

Figure 4. Comparison on CUB-200-2011 dataset. (a) The number of training images for each fine-grained category (top) and attribute (bottom). (b) An example of testing images. (c) 2 attributes predicted by GN-BGL$_m$. (d) Top-5 predictions generated by GN-SM approach (top) and the proposed BGL approach (bottom) respectively. (e) Similar training exemplars. (f) Accuracy.



Accuracy(%)	w/o. BBox		w/. BBox	
	SV	MV	SV	MV
HAR [57]	83.6%		-	
AN-SM	72.3	75.8	85.3	87.0
GN-SM	78.7	79.8	87.0	88.0
Type Coarse Labels				
GN-BGL$_1$	78.8	80.5	87.0	89.2
GN-BGL$_m$	83.5	84.6	88.8	89.9
Model Coarse Labels				
GN-BGL$_1$	81.6	83.0	87.9	89.2
GN-BGL$_m$	84.8	86.0	89.1	89.8
Both Coarse Labels				
GN-BGL$_1$	82.0	83.8	88.5	89.4
GN-BGL$_m$	**85.3**	**86.4**	**89.6**	**90.5**

(f)

Figure 5. Comparison on the Car-333 dataset. (a) Distribution of training images for each fine-grained (top) and coarse (bottom) category. (b) An example of testing image. (c) Type and model predicted by GN-BGL$_m$. (d) Top-5 prediction generated by GN (top) and the proposed BGL (bottom) respectively. (e) Similar training exemplars according to the input feature $\mathbf{x}$. (f) Accuracy.

BGL formulation provides a way to regularize CNN training to alleviate its overfitting issue. Fig. 6b-c show the time cost for performing forward and backward passing respectively given a 128-image mini-batch. Compared to GN-SM, GN-BGL needs only very little additional computation to perform exact inference in the loss function layer. This demonstrates the efficiency of modeling label dependency in a bipartite graphs. For the last fully connected (FC) layer, BGL performs exactly the same computation as GN in the forward passing, but needs additional cost for updating the gradient (Eq. 6 and Eq. 7) in the backward passing. Because both the loss and last FC layers take a very small portion of the whole pipeline, we found the total time difference between BGL and GN was minor.

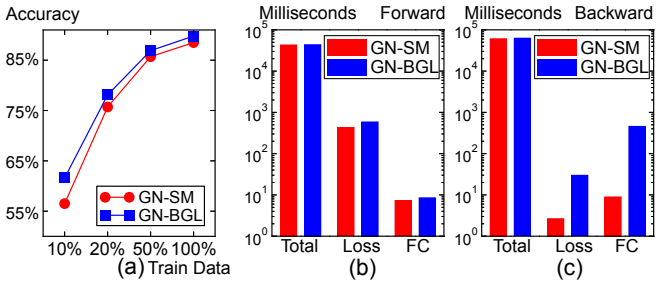
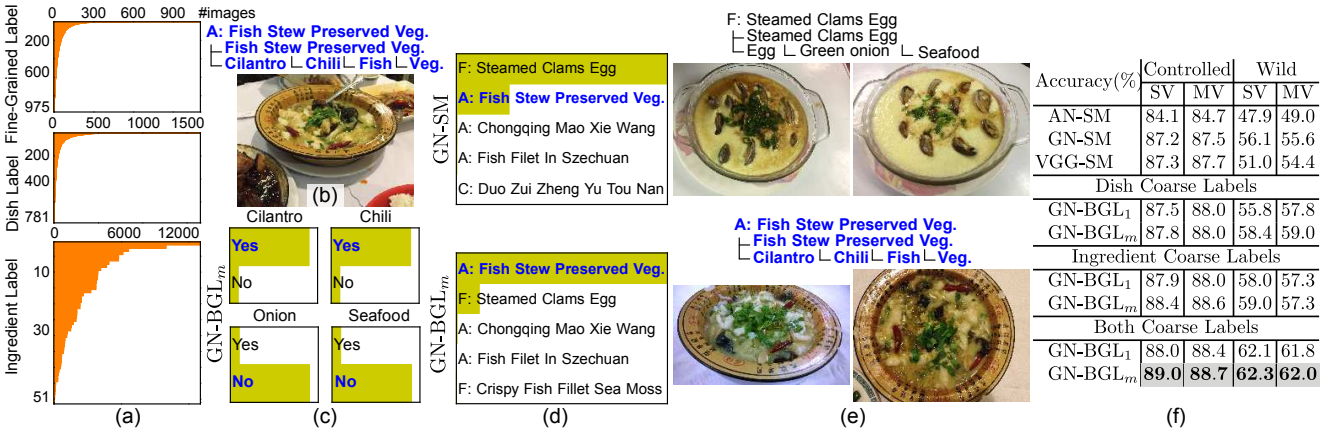


Figure 6. More comparison on the Car-333 datasets. (a) Accuracy as a function of the amount of data used in training. (b) Time cost for forward passing given a 128-image mini-batch, where Loss and FC denote the computation of the loss function and the last fully-connected layers respectively. (c) Time cost for backward passing.

4.4. Food-975 dataset

Now, let's come back to the task that we raised in the beginning of the paper: given a food image from a restaurant, are we able to recognize it as "which restaurant which dish"? Apparently, this is an ultra-fine grained image recognition problem and is very challenging. As we mentioned before, such ultra-fine granularity brings along rich relationships among the classes. In particular, different restaurants may cook similar dishes with different ingredients.

We collected a high quality food dataset. We sent 6 data collectors to 6 restaurants. Each data collector was in charge



Accuracy(%)	Controlled		Wild	
	SV	MV	SV	MV
AN-SM	84.1	84.7	47.9	49.0
GN-SM	87.2	87.5	56.1	55.6
VGG-SM	87.3	87.7	51.0	54.4
Dish Coarse Labels				
GN-BGL$_1$	87.5	88.0	55.8	57.8
GN-BGL$_m$	87.8	88.0	58.4	59.0
Ingredient Coarse Labels				
GN-BGL$_1$	87.9	88.0	58.0	57.3
GN-BGL$_m$	88.4	88.6	59.0	57.3
Both Coarse Labels				
GN-BGL$_1$	88.0	88.4	62.1	61.8
GN-BGL$_m$	**89.0**	**88.7**	**62.3**	**62.0**

Figure 7. Comparison on the Food-975 dataset. (a) Distribution of training images for each fine-grained (top), dish (middle), and ingredient (bottom) category. (b) An example of testing image. (c) 4 ingredients predicted by GN-BGL$_m$. (d) Top-5 predictions generated by GN-SM (top) and the proposed GN-BGL$_m$ (bottom) respectively. (e) Similar training exemplars according to the input feature $\mathbf{x}$. (f) Accuracy.

of one restaurant and took the photo of almost every dish the restaurant had cooked during a period of $1 \sim 2$ months. Finally, we captured $32,135$ high-resolution food photos of 975 menu items from the 6 restaurants for training. We evaluated our method in two settings. To test in a *controlled* setting, we took additional 4951 photos in different days. To mimic a realistic scenario in the *wild*, we downloaded 351 images from yelp.com posted by consumers visiting the same restaurants. To model the class relationship, we created a three-level hierarchy. In the first level, we have the 975 fine-grained labels; in the middle, we created 781 different dishes by aggregating restaurant tags; at last, we came up a detailed list of 51 ingredient attributes[4] that precisely describes the food composition.

Fig. 7 compared the proposed BGL approach with different baselines. Fig. 7e compares our method with AN-SM, GN-SM and VGG-SM in both the controlled and wild settings. We noticed that BGL approach consistently outperformed AN and GN in both settings. This indicates the effectiveness of exploring the label dependency in ultra-fine grained food recognition. Interestingly, by using both the dish and ingredient labels, BGL can gain a much larger improvement than only using one of them. This implies the connections between the dish labels and the ingredient labels have very useful information. Overall, by exploring the label dependency, the proposed BGL approach achieved $6 \sim 7\%$ improvement from GN baseline at the wild condition.

5. Conclusion

This paper proposed BGL to exploit the rich class relationships in the very challenging ultra-fine grained tasks. BGL improves the traditional softmax loss by jointly modeling fine-grained and coarse labels through bipartite-graph labels. The use of a special bipartite structure enables BGL to be efficient in inference. We also contribute Food-975, an ultra-fine grained food recognition benchmark dataset. We show that the proposed BGL approach improved previous work on a variety of datasets.

There are several future directions to our work. (1) For the ultra-fine grained image recognition, we may soon need to handle many more classes. For example, we are constructing a large food dataset from thousands of restaurants where the number of ultra-fine grained food classes can grow into hundreds of thousands. We believe that the research in this direction, ultra-fine-grained image recognition (recognizing images almost on instance-level), holds the key for using images as a media to retrieve information, which is often called *search by image*. (2) Although currently the label structure is manually defined, it can potentially be learned during training (*e.g.*, [47]). On the other hand, we are designing a web interface to scale up the attribute and ingredient labeling. (3) This paper mainly discusses discrete labels in BGL. It is also interesting to study the application with continuous labels (*e.g.*, regression or ranking). (4) Instead of operating only at class level, we plan to generalize BGL to deal with image-level labels. This can make the performance of BGL more robust in the case when the attribute label is ambiguous for fine-grained class.

[4]Find ingredient and restaurant list in the supplementary materials.

References

[1] A. Angelova and S. Zhu. Efficient object detection and segmentation for fine-grained recognition. In *CVPR*, 2013. 1, 2

[2] E. Bart, I. Porteous, P. Perona, and M. Welling. Unsupervised learning of visual taxonomies. In *CVPR*, 2008. 3

[3] O. Beijbom, N. Joshi, D. Morris, S. Saponas, and S. Khullar. Menu-Match: Restaurant-specific food logging from images. In *WACV*, 2015. 1

[4] S. Bell and K. Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Trans. Graph.*, 34(4):98, 2015. 1

[5] T. Berg, J. Liu, S. W. Lee, M. L. Alexander, D. W. Jacobs, and P. N. Belhumeur. Birdsnap: Large-scale fine-grained visual categorization of birds. In *CVPR*, 2014. 1

[6] L. Bossard, M. Guillaumin, and L. V. Gool. Food-101 - mining discriminative components with random forests. In *ECCV*, 2014. 1

[7] S. Branson, G. V. Horn, P. Perona, and S. Belongie. Bird species categorization using pose normalized deep convolutional nets. In *BMVC*, 2014. 2, 6

[8] S. Branson, G. V. Horn, C. Wah, P. Perona, and S. Belongie. The ignorant led by the blind: A hybrid human-machine vision system for fine-grained categorization. *Int. J. Comput. Vis.*, 108(1-2):3–29, 2014. 1, 2, 5

[9] R. Caruana. Multitask learning. *Machine Learning*, 28:41–75, 1997. 3

[10] G. Chen, J. Yang, H. Jin, J. Brandt, E. Shechtman, A. Agarwala, and T. X. Han. Large-scale visual font recognition. In *CVPR*, 2014. 1

[11] J. Deng, A. C. Berg, K. Li, and L. Fei-Fei. What does classifying more than 10,000 image categories tell us? In *ECCV*, 2010. 3

[12] J. Deng, N. Ding, Y. Jia, A. Frome, K. Murphy, S. Bengio, Y. Li, H. Neven, and H. Adam. Large-scale object classification using label relation graphs. In *ECCV*, 2014. 3, 4

[13] J. Deng, J. Krause, and L. Fei-Fei. Fine-grained crowdsourcing for fine-grained recognition. In *CVPR*, 2013. 2

[14] W. Di, C. Wah, A. Bhardwaj, R. Piramuthu, and N. Sundaresan. Style finder: Fine-grained clothing style detection and retrieval. In *CVPR Workshop on Mobile Vision*, 2013. 1, 2

[15] M. Everingham, S. M. A. Eslami, L. V. Gool, C. K. I. Williams, J. M. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.*, 111(1):98–136, 2015. 1

[16] T. Evgeniou and M. Pontil. Regularized multi-task learning. In *KDD*, 2004. 3

[17] R. Farrell, O. Oza, N. Zhang, V. I. Morariu, T. Darrell, and L. S. Davis. Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In *ICCV*, 2011. 2

[18] R. Fergus, H. Bernal, Y. Weiss, and A. Torralba. Semantic label sharing for learning with many categories. In *ECCV*, 2010. 3

[19] V. Ferrari and A. Zisserman. Learning visual attributes. In *NIPS*, 2007. 2

[20] E. Gavves, B. Fernando, C. G. M. Snoek, A. W. M. Smeulders, and T. Tuytelaars. Fine-grained categorization by alignments. In *ICCV*, 2013. 2

[21] K. He, X. Zhang, S. Ren, and J. Sun. Delving deep into rectifiers: Surpassing human-level performance on ImageNet classification. *CoRR*, abs/1502.01852, 2015. 1

[22] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. 1

[23] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu. Spatial transformer networks. In *NIPS*, 2015. 2

[24] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. *arXiv:1408.5093*, 2014. 5

[25] S. Karayev, M. Trentacoste, H. Han, A. Agarwala, T. Darrell, A. Hertzmann, and H. Winnemoeller. Recognizing image style. In *BMVC*, 2014. 5

[26] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *CVPR Workshop on Fine-Grained Visual Categorization*, 2011. 1

[27] J. Krause, J. Deng, M. Stark, and L. Fei-Fei. Collecting a large-scale dataset of fine-grained cars. Technical report, Stanford, 2013. 5

[28] J. Krause, T. Gebru, J. Deng, L. Li-Jia, and L. Fei-Fei. Learning features and parts for fine-grained recognition. In *ICPR*, 2014. 2, 5

[29] J. Krause, H. Jin, J. Yang, and L. Fei-Fei. Fine-grained recognition without part annotations. In *CVPR*, 2015. 2, 5, 6

[30] J. Krause, M. Stark, J. Deng, and L. Fei-Fei. 3D object representations for fine-grained categorization. In *ICCV Workshop on 3D Representation and Recognition*, 2013. 1, 2, 5

[31] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 1, 3, 5

[32] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. John, I. C. Lopez, and J. V. B. Soares. LeafSnap: A computer vision system for automatic plant species identification. In *ECCV*, 2012. 1

[33] N. Kumar, A. C. Berg, P. N. Belhumeur, and S. K. Nayar. Describable visual attributes for face verification and image search. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(10):1962–1977, 2011. 2

[34] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *CVPR*, 2009. 2

[35] T.-Y. Lin, A. R. Chowdhury, and S. Maji. Bilinear CNN models for fine-grained visual recognition. In *ICCV*, 2015. 2, 5, 6

[36] Y. Lin, V. I. Morariu, W. H. Hsu, and L. S. Davis. Jointly optimizing 3D model fitting and fine-grained classification. In *ECCV*, 2014. 1

[37] J. Liu, A. Kanazawa, D. W. Jacobs, and P. N. Belhumeur. Dog breed classification using part localization. In *ECCV*, 2012. 2

[38] A. Myers, N. Johnston, V. Rathod, A. Korattikara, A. Gorban, N. Silberman, S. Guadarrama, G. Papandreou, J. Huang, and K. Murphy. Im2Calories: towards an automated mobile vision food diary. In *ICCV*, 2015. 1

[39] J. Nam, L. Mencía, E., H. J. Kim, and J. Fürnkranz. Predicting unseen labels using label hierarchies in large-scale multi-label learning. In *ECML*, 2015. 3

[40] M. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *ICVGIP*, 2008. 1

[41] D. Parikh and K. Grauman. Relative attributes. In *ICCV*, 2011. 2

[42] O. M. Parkhi, A. Vedaldi, C. V. Jawahar, and A. Zisserman. The truth about cats and dogs. In *ICCV*, 2011. 2

[43] O. M. Parkhi, A. Vedaldi, A. Zisserman, and C. V. Jawahar. Cats and dogs. In *CVPR*, 2012. 1, 2

[44] Q. Qian, R. Jin, S. Zhu, and Y. Lin. Fine-grained visual categorization via multi-stage metric learning. In *CVPR*, 2015. 2

[45] A. R. Sfar, N. Boujemaa, and D. Geman. Vantage feature frames for fine-grained categorization. In *CVPR*, 2013. 1

[46] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014. 1, 3, 5

[47] N. Srivastava and R. Salakhutdinov. Discriminative transfer learning with tree-based priors. In *NIPS*, 2013. 3, 4, 8

[48] M. Stark, J. Krause, B. Pepik, D. Meger, J. J. Little, B. Schiele, and D. Koller. Fine-grained categorization for 3D scene understanding. In *BMVC*, 2012. 1

[49] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014. 1, 2, 3, 5

[50] S. Todorovic and N. Ahuja. Learning subcategory relevances for category recognition. In *CVPR*, 2008. 3

[51] A.-M. Tousch, S. Herbin, and J.-Y. Audibert. Semantic hierarchies for image annotation: A survey. *Pattern Recognit.*, 45(1):333–345, 2012. 3

[52] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD birds-200-2011 dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5

[53] J. Wang, Y. Song, T. Leung, C. Rosenberg, J. Wang, J. Philbin, B. Chen, and Y. Wu. Learning fine-grained image similarity with deep ranking. In *CVPR*, 2014. 2

[54] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, 2013. 6

[55] M. Wilber, I. Kwak, D. Kriegman, and S. Belongie. Learning concept embeddings with combined human-machine expertise. In *ICCV*, 2015. 2

[56] R. Wu, S. Yan, Y. Shan, Q. Dang, and G. Sun. Deep image: Scaling up image recognition. *CoRR*, abs/1501.02876, 2015. 1

[57] S. Xie, T. Yang, X. Wang, and Y. Lin. Hyper-class augmented and regularized deep learning for fine-grained image classification. In *CVPR*, 2015. 2, 5, 6

[58] Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *J. Mach. Learn. Res.*, 8:35–63, 2007. 3

[59] S. Yang, M. Chen, D. Pomerleau, and R. Sukthankar. Food recognition using statistics of pairwise local features. In *CVPR*, 2010. 1

[60] A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *CVPR*, 2014. 2

[61] A. Yu and K. Grauman. Just noticeable differences in visual attributes. In *ICCV*, 2015. 3

[62] N. Zhang, J. Donahue, R. B. Girshick, and T. Darrell. Part-based R-CNNs for fine-grained category detection. In *ECCV*, 2014. 2, 6

[63] N. Zhang, R. Farrell, F. N. Iandola, and T. Darrell. Deformable part descriptors for fine-grained recognition and attribute prediction. In *ICCV*, 2013. 2, 6

[64] N. Zhang, M. Paluri, M. Ranzato, T. Darrell, and L. D. Bourdev. PANDA: Pose aligned networks for deep attribute modeling. In *CVPR*, 2014. 2