Which Looks Like Which: Exploring Inter-class Relationships in Fine-Grained Visual Categorization

Jian Pu[1], Yu-Gang Jiang[1], Jun Wang[2], and Xiangyang Xue[1]

[1] School of Computer Science, Shanghai Key Laboratory of Intelligent Information Processing, Fudan University, Shanghai, China
[2] IBM T. J. Watson Research Center, Yorktown Heights, NY, USA
{jianpu,ygj,xyxue}@fudan.edu.cn, wangjun@us.ibm.com

Abstract. Fine-grained visual categorization aims at classifying visual data at a subordinate level, e.g., identifying different species of birds. It is a highly challenging topic receiving significant research attention recently. Most existing works focused on the design of more discriminative feature representations to capture the subtle visual differences among categories. Very limited efforts were spent on the design of robust model learning algorithms. In this paper, we treat the training of each category classifier as a single learning task, and formulate a *generic* multiple task learning (MTL) framework to train multiple classifiers simultaneously. Different from the existing MTL methods, the proposed generic MTL algorithm enforces no structure assumptions and thus is more flexible in handling complex inter-class relationships. In particular, it is able to automatically discover both clusters of similar categories and outliers. We show that the objective of our generic MTL formulation can be solved using an iterative reweighted ℓ_2 method. Through an extensive experimental validation, we demonstrate that our method outperforms several state-of-the-art approaches.

Keywords: Fine-grained visual categorization, inter-class relationship, multiple task learning.

1 Introduction

Object recognition has been extensively studied in computer vision. Significant progress has been made in the recognition of basic categories like bird and car. Recently, an increasing amount of attention is being paid to the study of Fine-Grained Visual Categorization (FGVC), which aims at the identification and distinction of subcategories such as different species of birds or dogs [5,11,18,19,34,37,40,41,42,43]. Algorithms and systems with such capabilities not only enhance the performance of conventional object recognition, but also can aid humans in specific domains, since even human experts may have difficulties in recognizing some subcategories.

Generally, there are two critical challenges in the design of a robust FGVC system. First, object categories under the same coarse semantic level often share

D. Fleet et al. (Eds.): ECCV 2014, Part III, LNCS 8691, pp. 425–440, 2014.

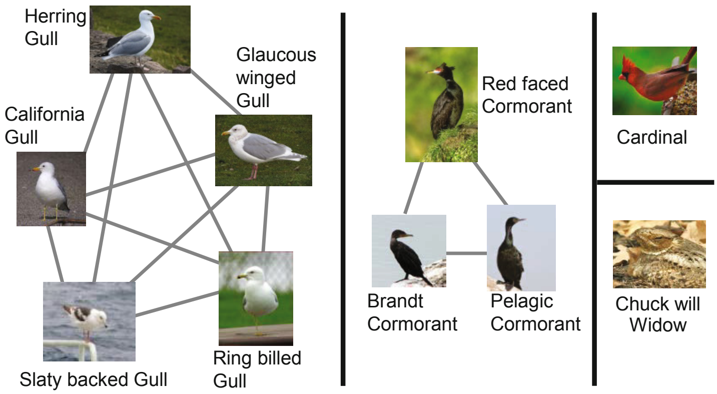


Fig. 1. Illustration of category relationships in fine-grained visual categorization problems, using birds as an example. There may exist multiple *clusters* containing highly similar categories (e.g., various species of Gull and Cormorant), as well as *outlier* categories that are distinct from others (e.g., Cardinal and Chuck will Widow). This paper proposes a generic multiple task learning algorithm, which is able to automatically discover and utilize the category grouping and outlier structure for improved fine-grained categorization performance.

similar appearances and the visual differences are very subtle. As shown in Figure 1, the subcategories within the same group (Gull or Cormorant) tend to share similar appearances. Therefore, very sophisticated features may be needed to distinguish such fine-grained categories. The second challenge is that fine-grained categorization tasks always lack in training data since the acquisition of clean positive samples for each subordinate category needs strong domain knowledge.

To address these challenges, we underline that FGVC systems should encode at least two characteristics of inter-class relationships: 1) sufficient discriminative capability to distinguish the subtle differences among the subcategories, and 2) strong learning power to explore similarities among categories to compensate for the lack of training samples. Most of the existing works only focused on solving the first issue by proposing new feature representations, as they emphasized that insufficient discriminative power resides in the way that features are encoded [19,41,18,43,34,40].

In this paper, we adopt and improve the *multiple task learning* (MTL) framework to design more discriminative classifiers. Instead of training classifiers independently, MTL trains multiple classifiers jointly and simultaneously to explore model commonalities in either structure or parameter space [3,12]. Since simultaneously learning multiple tasks will benefit from the learning of each other, the MTL paradigm often leads to better performance. However, standard MTL highly relies on the assumption of a clean model commonality, which is too idealistic to be practically used for FGVC. Although some recent works relaxed the strong assumption to *grouped structure* [25] or *outlier structure* [22] of tasks, the category relationships in the realistic FGVC problem could be more complicated and do not fit those existing structure assumptions.

Realizing the limitation of the existing MTL paradigms and the practical needs of FGVC, this paper presents a *generic* MTL framework without any specific structure assumptions. Our method exploits the relationships among the fine-grained categories by imposing a mixture norm penalty on the classifier coefficients to automatically learn the task structures, where a ridge term is used to reflect the categories' grouping structure and a lasso term represents outlier categories. Our objective is formulated as an unconstrained convex optimization problem, whose optimal solution can be obtained by iteratively solving a series of reweighted ℓ_2 problems. Extensive evaluations on two well-known FGVC benchmark datasets demonstrate the effectiveness our proposed method.

The remainder of this paper is organized as follows. Section 2 briefly reviews existing works in FGVC. Section 3 presents our proposed generic MTL framework. Section 4 provides experimental validations and comparative studies, and, finally, Section 5 concludes this paper.

2 Related Works

In this section we briefly review existing works on FGVC; the backgrounds of MTL will be discussed later in Section 3.1. Like any visual categorization applications, an FGVC system normally contains two major components: feature extraction and classifier learning. As mentioned earlier, most existing works focused on improving the discrimination power of the extracted feature representations (e.g., [19,26,43,41], among others), and a standard SVM classifier was often employed in the learning phase.

For feature representations used in FGVC, many researchers adopted local features as the basis to develop more powerful visual descriptors. For instance, locality-constrained linear coding [36], an effective bag-of-words quantization method, was adopted in [42] as a baseline. In [26], Khan et al. proposed a multi-cue method to build discriminative compound visual words from primitive cues. The kernel descriptors (KDES) [10] have also been adopted for FGVC and shown to be promising [40].

Another popular group of works used template-based feature representation and demonstrated good performance for FGVC applications. In [41], Yao et al. used randomly selected templates and generated visual features through concatenating the pooling results on template response maps. In [40], Yang et al. proposed an unsupervised template learning method to capture the common shape patterns across different categories. Several other factors such as the co-occurrence and diversity of templates were also taken into account.

In addition, Chai et al. [13] proposed a method called TriCoS to segment discriminative foregrounds. Segmentation based approaches were also explored in [1,14,39]. Several works further adopted localization approaches to identify discriminative parts or details of the target objects [44,21]. These methods, however, are computationally slow as both the segmentation and the part localization processes are expensive.

All the aforementioned approaches are less powerful in exploring the category relatedness and identifying the subtle differences across categories. Perhaps the most intuitive way to identify the subtle differences among categories is to use human-assisted techniques. Representative works include the human-in-the-loop approaches that asked humans to input object attributes [11,34], and the crowdsourcing-based method to identify more discriminative features [17]. Another approach that is loosely related to this category is the poselet-like methods [19,43], where human inputs were needed to label keypoint locations. The acquisition of manual annotations needed by these approaches is very expensive, and the inputs of the annotation tasks (e.g., the attribute questions) also need to be designed by experts with sufficient domain knowledge.

The lack of research on better model learning techniques in FGVC, especially those exploring the category relatedness, motivated us to propose the following generic MTL framework, which can automatically discover and utilize the complex inter-class relationships to achieve better performance. An intuitive way of using the category relationships is to explore domain knowledge. For example, adopting class taxonomies or hierarchies [6,23] may help train better prediction models by sharing appearance [20,30], visual parts [33], or classifiers [4]. However, the specific domain knowledge is not easy to be obtained, and the developed taxonomy for one FGVC application cannot be generalized to other different domains. The idea of using the class relationships was also exploited very recently in [8,9], where the authors proposed to use one-vs-*most* SVMs to find significant features that distinguish different species by omitting some similar classes, which is fundamentally different from our proposed solution.

3 Exploring Inter-class Relationships in FGVC

In this section, we present our proposed method to explore the inter-class relationships in FGVC. We start with introducing the notations and a brief background of MTL.

3.1 Notations and Background

Given a categorization problem, denote the training data containing n samples as $\{\mathbf{X}, \mathbf{z}\}$, where $\mathbf{X} = \{\mathbf{x}_i\}_{i=1}^n$ is the training set with $\mathbf{x}_i \in \mathbb{R}^D$ representing a D-dimensional feature of the i-th sample, and $\mathbf{z} = \{z_i\}_{i=1}^n, z_i \in \{1, \cdots, L\}$ is the label set for L categories. For a typical multi-class case, the one-vs-all strategy is widely used to train a classifier for each category. Hence, for the l-th category, we convert the multi-class label vector $\mathbf{z} = \{z_i\}_{i=1}^n$ to a binary label vector $\mathbf{y}_l = \{y_{li}\}_{i=1}^n, y_{li} \in \{-1, 1\}$ as $y_{li} = 1$ if $z_i = l$, otherwise $y_{li} = -1$. Assume that the classifier for the l-th category is defined in a linear form as $\hat{y}_l = \mathbf{w}_l^\top \mathbf{x} + b_l$, where $\hat{y}_l$ is the prediction; $\mathbf{w}_l \in \mathbb{R}^D$ and b_l are the coefficient vector and the bias[1], respectively. The cost function for training all the classifiers $\{\mathbf{w}_l\}_{l=1}^L$ is often written as

[1] In the following we omit the bias term b_l for simplicity.

$$\min_{\mathbf{W}} \sum_{l=1}^{L} \left(\sum_{i=1}^{n} \mathcal{V}(\mathbf{w}_l^\top \mathbf{x}_i, y_{li}) + \lambda \|\mathbf{w}_l\|_2 \right). \tag{1}$$

A major issue of the above formulation is that the relationships of different categories are ignored and the training for each category is performed independently. This normally leads to degraded performance particularly when the positive training samples are insufficient, which is often observed in FGVC applications. Simultaneously training multiple classifiers by MTL can effectively alleviate this problem. Formally, a basic MTL method is to replace the ℓ_2 penalty of each classifier with a structure penalty to constrain all the classifiers, with the following cost function:

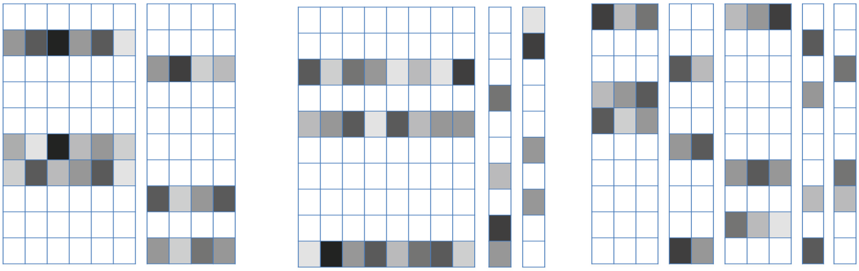
$$\min_{\mathbf{W}} \sum_{l=1}^{L} \sum_{i=1}^{n} \mathcal{V}(\mathbf{w}_l^\top \mathbf{x}_i, y_{li}) + \lambda \|\mathbf{W}\|_{2,1}. \tag{2}$$

where the matrix $\mathbf{W}$ is formed through concatenating single classifiers as $\mathbf{W} = [\mathbf{w}_1, \mathbf{w}_2, \cdots, \mathbf{w}_l]$. The regularization term $\|\mathbf{W}\|_{2,1} = \sum_i \left(\sum_j w_{ij}^2 \right)^{1/2}$ induces row sparsity that encourages the elements of the same row to maintain similar zero/nonzero patterns. Minimizing the above cost to derive the optimal $\mathbf{W}$ can also be viewed as a feature selection process since the commonly shared discriminative features will be preserved as non-zero row vectors in the $\mathbf{W}$ matrix. The major limitation of this basic MTL formulation lies in the assumption that all the classifiers $\{\mathbf{w}_l\}_{l=1}^{L}$ share a common sparse structure.

To relax the common structure assumption, there are two major categories of advanced MTL methods. First, *cluster-based* MTL methods consider the existence of several task (category) clusters, where features are only shared within each cluster and irrelevant to tasks outside the cluster. Thus, several approaches introduced latent variables to indicate the cluster information or to select the features to be shared [25,28,45]. The optimization of the latent variables is usually merged into the main MTL procedure as a subroutine. Second, *robust* MTL methods assume that all the tasks consist of a major task group peppered with several outlier tasks. A popular way of tackling this robust MTL problem is to use a decomposition framework, which forms a learning objective with a structure term and an outlier penalty term [24,16,22]. Then, the target model is further decomposed into two components, i.e., a group component and an outlier component, which can be efficiently solved separately. Figure 2(a) illustrates the learned classifiers $\mathbf{W}$ by cluster-based MTL, where each column vector represents a single learning task and a total of two groups of tasks are identified. Similarly, Figure 2(b) demonstrates a structure of the learned classifiers from the robust MTL, where a major group of tasks and two outlier tasks can be observed.

3.2 Generic Multiple Task Learning (GMTL)

Since there always exist certain kinds of relationships between the categories in FGVC applications, performing MTL can help boost the classification performance



(a) Cluster-based MTL (b) Robust MTL (c) Our GMTL

Fig. 2. Illustration of the structures of the learned classifiers $\mathbf{W}$ using cluster-based MTL, robust MTL, and our GMTL. Each column represents a classifier $\mathbf{w}_l$ and each row represents the learned coefficients corresponding to a feature dimension. The white color indicates zero coefficient value, and the gray-scale colors reflect the magnitude of nonzero values. See text for more explanations.

through identifying the shared features across the categories. Especially, MTL is very suitable when the positive samples are inadequate for each classifier and the feature representation is in very high dimensions. However, due to the complex structures of the fine-grained categories, the aforementioned MTL models are infeasible for such applications since they all rely on strong assumptions of simple task structures. As illustrated by the bird examples in Figure 1, a usual FGVC problem may have the following structure characteristics: 1) some categories are strongly related and form a category group since they share similar visual signatures; 2) the similarity between different category groups may be very low (e.g., the Gull and Cormorant groups); 3) some categories are not similar to all the other categories, e.g., the Cardinal and Chuck will Widow (i.e., outlier categories).

Motivated by the above observations, we formulate a generic MTL (GMTL) framework without any specific structure assumption: the categories are related in a mixture manner with unknown clusters and outliers. As illustrated in Figure 2(c), the categories consist of multiple clusters and outliers. To capture such a complex task structure for FGVC applications, the proposed GMTL model explores a balanced mixture of the category clusters and outliers as

$$\min_{\mathbf{W}} \sum_{l=1}^{L} \sum_{i=1}^{n} \mathcal{V}(\mathbf{w}_l^\top \mathbf{x}_i, y_{li}) + \lambda \left(\alpha \|\mathbf{W}\|_{2,1} + (1-\alpha)\|\mathbf{W}\|_{1,1}\right). \tag{3}$$

The first term is the empirical loss of fitting the training data, and the regularization term consists of two parts: the feature sharing part for category grouping and the outlier detection part for category outliers. Accordingly, the $\|\mathbf{W}\|_{2,1}$ penalty reflects a grouping structure, and encourages feature sharing among tasks within each task group. The $\|\mathbf{W}\|_{1,1}$ penalty reflects an element-wise sparse structure, highlighting the outlier categories. The coefficient λ weighs the contribution of the total penalties, and the parameter α balances the two regularization terms.

In this paper, we particularly choose the squared hinge loss as our empirical loss for the FGVC problem:

$$\mathcal{V}(\mathbf{w}^\top \mathbf{x}, y) = \left[max(0, 1 - y\mathbf{w}^\top \mathbf{x})\right]^2. \tag{4}$$

Compared with the standard hinge loss, the above squared hinge loss penalizes less when the sign of prediction is correct but within the margin, and penalizes more when the sign of prediction is wrong. In addition, the squared hinge loss provides a better computational efficiency, which is extremely important when we need to solve the primal problem directly [15,29]. Note that a similar objective function with mixture norms and the hinge loss was investigated in a recent work [46] for other purposes using a different optimization strategy. However, the non-smoothness of the hinge loss may affect the convergence speed of their method. Another work used similar structural regularization forms with the mixture norms for regression problems [35], which is also different from the classification scenario of FGVC.

To further explore the role of these two penalties and better understand the above formulation, we rewrite Eq. 3 as:

$$\min_{\mathbf{W}} \sum_{l=1}^{L} \left(\sum_{i=1}^{n} \mathcal{V}(\mathbf{w}_l^\top \mathbf{x}_i, y_{li}) + \lambda(1-\alpha)\|\mathbf{w}_l\|_1 \right) + \lambda\alpha\|\mathbf{W}\|_{2,1}. \tag{5}$$

In contrast to the basic MTL formulation in Eq. 2, although we have the same MTL penalty $\mathbf{W}_{2,1}$ to encourage feature sharing among tasks, the term in the parentheses is different: we are solving an ℓ_1 regularized data fitting instead of the unregularized data fitting for each category. This ℓ_1 regularized term can shrink the small values of $\mathbf{w}_l$ to zero. Combining the effects of the two penalties, the optimization of Eq. 3 can satisfy all the three situations in fine-grained categorization: 1) the $\ell_{2,1}$ norm generally enforces categories to share features, including those categories from different groups; 2) since the categories in different groups are less relevant, the magnitude of feature sharing should be small; the $\ell_{1,1}$ penalty tends to shrink the corresponding weights to zero; 3) for the outliers, the mixture of the $\ell_{2,1}$ and $\ell_{1,1}$ penalties shrinks the unrelated features to be zero weighted. In summary, the above mixed structure regularization encourages category grouping, as well as identifies outlier categories.

3.3 Optimization Strategy

Solving the optimization problem in Eq. 3 is nontrivial due to the coupling of all the classifiers and the discontinuity of the regularization penalty. One way to optimize this problem is to employ the proximal gradient method [7]. However, for such a mixture norm penalty, it usually requires two shrinkage operations in the projection step [32], which are inefficient. Another way to optimize the mixture norm penalty is to iteratively solve a series of reweighted ℓ_2 problems [38]. With some derivations, the original problem in Eq. 3 can be rewritten as

$$\min_{\mathbf{W}} \sum_{l=1}^{L} \left(\sum_{i=1}^{n} \mathcal{V}(\mathbf{w}_l^\top \mathbf{x}_i, y_{li}) + \frac{\lambda}{2} \|\mathbf{C}_l \mathbf{w}_l\|_2 \right). \tag{6}$$

Here $\mathbf{C}_l = diag\left[(\mathbf{C}_l)_1, \cdots (\mathbf{C}_l)_d, \cdots, (\mathbf{C}_l)_D\right]$ is a diagonal weight matrix with the elements computed as:

$$(\mathbf{C}_l)_d = \alpha \|\mathbf{w}_{d\cdot}\|_2^{-1} + (1-\alpha)|w_{dl}|^{-1}, \tag{7}$$

where $\mathbf{w}_{d\cdot}$ represents the d-th row vector of $\mathbf{W}$. Note that $(\mathbf{C}_l)_d$ consists of two components: a group impact term $\|\mathbf{w}_{d\cdot}\|_2^{-1}$ imposing the global effect across all categories, and an individual term $|w_{dl}|^{-1}$ denoting the impact of the d-th feature on the l-th category.

Algorithm 1. Training Procedure of GMTL

```
Require: X: feature representation of all the samples; $\{\mathbf{y}_l\}_{l=1}^L$: binary label vector for
    each category;
Initialize $\{\mathbf{C}_l\}_{l=1}^L$ with the identity matrix;
while not converged do
    for $l = 1$ to $L$ do
        Reweight the training data:
        $\tilde{\mathbf{x}}_{li} = (\mathbf{C}_l)^{-1}\mathbf{x}_i$;
        Solve an $\ell_2$ regularized minimization problem:
        $\tilde{\mathbf{w}}_l = \min_{\tilde{\mathbf{w}}_l} \left(\sum_{i=1}^n \mathcal{V}(\tilde{\mathbf{w}}_l^\top \tilde{\mathbf{x}}_{li}, y_{li}) + \frac{\lambda}{2}\|\tilde{\mathbf{w}}_l\|_2\right)$;
        Compute the coefficient $\mathbf{w}_l$ for each classifier:
        $\mathbf{w}_l = (\mathbf{C}_l)^{-1}\tilde{\mathbf{w}}_l$;
        Update the weight matrix:
        $\mathbf{C}_l = diag\left(\alpha\|\mathbf{w}_{d\cdot}\|_2^{-1} + (1-\alpha)|w_{dl}|^{-1}\right)$;
    end for
end while
```

Compared with the formulation of basic MTL in Eq. 1, the key difference lies in the regularization term, where a diagonal matrix $\mathbf{C}_l$ is applied to weight the importance of the individual feature dimensions for each classification model $\mathbf{w}_l$. To further simplify such a weighted ℓ_2 form in Eq. 6, we can transform the problem to reweighted data instead of reweighted classifiers. Denoting $\tilde{\mathbf{w}}_l = \mathbf{C}_l \mathbf{w}_l$ and $\tilde{\mathbf{x}}_{li} = \mathbf{C}_l^{-1} \mathbf{x}_i$, we can derive a standard ℓ_2 regularized optimization problem with reweighted data as:

$$\min_{\tilde{\mathbf{W}}} \sum_{l=1}^{L} \left(\sum_{i=1}^{n} \mathcal{V}(\tilde{\mathbf{w}}_l^\top \tilde{\mathbf{x}}_{li}, y_{li}) + \frac{\lambda}{2} \|\tilde{\mathbf{w}}_l\|_2 \right). \tag{8}$$

Starting from any reasonable initialization, we can solve the above minimization problem iteratively, where during each iteration we need to update the weight matrices $\{\mathbf{C}_l\}_{l=1}^L$ using Eq. 7 and reweight the data. Algorithm 1 summarizes the training procedure of the proposed GMTL through the iterative reweighted ℓ_2 method. In each iteration, to learn a classifier, there are four

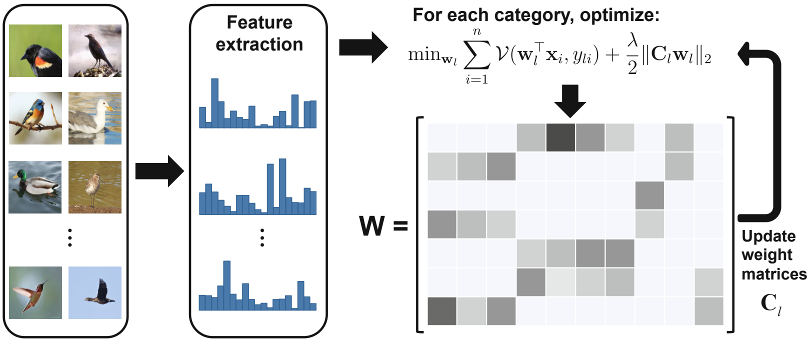

Feature extraction
For each category, optimize:
$\min_{\mathbf{w}_l} \sum_{i=1}^{n} \mathcal{V}(\mathbf{w}_l^\top \mathbf{x}_i, y_{li}) + \frac{\lambda}{2}\|\mathbf{C}_l\mathbf{w}_l\|_2$
W =
Update weight matrices
$\mathbf{C}_l$

Fig. 3. Illustration of the iterative training process in GMTL

steps: *a*) reweight the data to obtain $\tilde{\mathbf{x}}_{li}$; *b*) solve an ℓ_2 regularized minimization problem; *c*) compute the coefficient vector of the classifier $\mathbf{w}_l$; *d*) update the weight matrix $\mathbf{C}_l$ (referring to step 4-7 in Algorithm 1). The first three steps are equivalent to solving an ℓ_2 reweighted classifier, and the last step updates the weight matrix $\mathbf{C}_l$. Therefore, the GMTL optimization is formed in nested loops, where the outer loop (while-loop) is for pursuing local convergence and the inner loop (for-loop) is for updating the classifier of each category. Detailed proof of the convergence of the GMTL is omitted due to space limitation. In our experiments, we have empirically observed that the convergence can be often achieved after just a few iterations. The conceptual pipeline of using the proposed GMTL for learning fine-grained categorization models is also demonstrated in Figure 3.

4 Experiments

In this section, we first introduce the used datasets and experimental settings. Then we discuss our results and present comparison studies with state-of-the-art methods. After that we visualize the automatically identified category groups.

4.1 Datasets and Settings

We evaluate our approach using mainly two public datasets: the Stanford Dog [27] and the Caltech-UCSD Bird-200-2010 [37], which are widely used in the FGVC literature. The Stanford Dog dataset contains 20, 580 images from 120 breeds of dogs. Following the standard setup of [27], 100 images from each category are used for training and the rest are used for testing. The Caltech-UCSD Bird-200-2010 (CUB-200-2010) contains 6, 033 images from 200 bird species in North America, with about 30 images per class and 15 of them for training [37]. Exemplar images from Dog and Bird datasets are shown in Figure 4. Following the standard procedure in existing FGVC works [27,37], all the images are cropped according to the provided dog/bird bounding boxes before feature extraction. The images are then

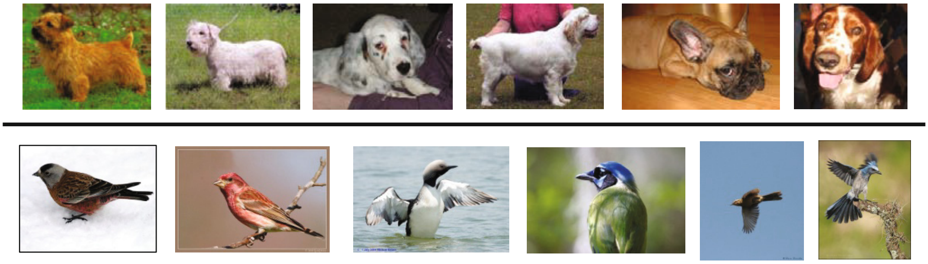


Fig. 4. Image examples from the Stanford Dog Dataset (top) and the CUB-200 Dataset (bottom)

resized to be no larger than 300 × 300 with the original aspect ratio preserved. At the end of the experiments, we also briefly discuss our results on the newer 2011 version of the bird dataset, which has more training and testing images per class.

We adopt the kernel descriptors (KDES) [10,40] to represent each image as a feature vector. Following [10,40], we use four types of the kernel descriptors: color-based, normalized color-based, gradient-based, and local-binary-pattern-based. The color and normalized color kernel descriptors are extracted from the original RGB images, and the other descriptors are extracted from converted gray scale images. All the kernel descriptors are computed on 16×16 image patches over dense regular grids with a step size of 8 pixels. Combining all the descriptors, we receive the final image representation as a $120,000$-dimension feature vector.

In addition to comparing our results with the state-of-the-art FGVC methods, we also compare with two representative MTL algorithms: Joint Feature Selection (JFS) [2] and Clustered MTL (CMTL) [45]. The formulation of JFS can be treated as a special case of our generic solution by setting $\alpha = 1$, and the CMTL method is based on the spectral-relaxed k-means clustering. We use the source codes provided by the authors of [45]. For all the MTL methods including ours, we use cross validation to estimate suitable parameters.

Table 1. Comparison of the classification accuracies on the Stanford Dog dataset

Approach		Accuracy (%)
State-of-the-art FGVC methods	SIFT [27]	22.0
	KDES [10]	36.0
	UTL [40]	38.0
	Symb+DPM [14]	45.6
MTL methods	JFS [2]	29.9
	CMTL [45]	30.4
	Our GMTL	39.3

4.2 Results and Discussions

Dog Categorization. We now discuss results on the Stanford Dog dataset. We compare with a SIFT-based method [27], KDES [10], an approach using unsupervised template learning (UTL) [40] and a recent work [14]. For all the approaches, the classification accuracies are reported using the same settings. In addition, the performance of JFS and CMTL are also reported, using the same KDES features. Table 1 gives the classification accuracies of various approaches. It is easy to see that the GMTL approach significantly outperforms the other two MTL methods. It is worth noting that both JFS and CMTL have worse performance than the KDES baseline. This is due to the existence of both subtle and drastic appearance variations among categories in FGVC datasets, which result in negative transfer or improper feature sharing in the JFS and CMTL methods. As discussed earlier in Section 3, our proposed GMTL is able to cope with the existence of both category clusters and outliers, which enables a more appropriate exploration of class relationships, and thus offers better performance. Our GMTL improves the KDES baseline by 3.3%, which is a significant gain considering the difficulty of the problem. Note that the recently developed approach [14] exploits symbiotic segmentation and part localization techniques to achieve strong performance on this dataset. However, this approach is computationally more expensive and the performance highly relies on the quality of segmentation and localization.

14 Bird Species Categorization. Next we experiment with the CUB-200-2010 dataset, which has more categories, less training data, and even more significant appearance variations across categories. Since this dataset is very challenging, in many existing works, a subset of 14 species was frequently used for evaluation. For the ease of comparison we also report results on this subset.

The subset contains two families of birds: Vireos and Woodpeckers [19]. Following [19,41], we produce a left-right mirrored image for each training and test image, which forms a total of 420 training images and 508 testing images. We compare with the following published approaches: multiple kernel learning (MKL) [11], Birdlet [19], a random template method [41], and the KDES [10]. A few very recent approaches are excluded from the comparison since they are designed under different settings and require additional human inputs like [17].

Following [41], we report the performance on this dataset using mean average precision (mAP) in Table 2. Again, we find that JFS and CMTL fail to improve the KDES baseline, and our GMTL significantly outperforms these two MTL methods. Compared with the state-of-the-art FGVC methods, our GMTL performs better than all of them. Table 3 further gives the per-class results for the three compared MTL approaches on this subset. For most of the bird subcategories, the proposed GMTL provides a visible performance gain over the compared JFS and CMTL methods.

All Bird Species Categorization. Finally, we test our method on the full bird dataset of 200 species. Results are summarized in Table 4. We compare with a multiple kernel learning (MKL) method [11], a bag-of-features approach using the LLC [36], a randomization based method [42], a multi-cue representation [26],

Table 2. Performance comparison on the bird subset of 14 species, measured by mean average precision (mAP)

Approach		mAP (%)
State-of-the-art FGVC methods	MKL [11]	37.0
	Birdlet [19]	40.3
	Random template [41]	44.7
	KDES [10]	42.5
MTL methods	JFS [2]	38.9
	CMTL [45]	40.6
	Our GMTL	45.7

Table 3. Per-class average precision (%) of the three MTL-based methods on the bird subset of 14 species. The best results of each row are shown in bold. We list abbreviated names of the bird species due to space limitation.

	JFS [2]	CMTL [45]	Our GMTL
BC Vireo	33.0	33.9	**39.5**
BH Vireo	**22.3**	20.9	**22.3**
P Vireo	33.1	35.6	**45.7**
RE Vireo	14.5	**14.8**	13.3
W Vireo	14.0	**18.1**	17.2
WE Vireo	49.1	48.4	**54.7**
YT Vireo	23.9	25.0	**28.3**
N Flicker	66.3	65.2	**76.0**
ATT Woodpecker	58.1	57.9	**63.6**
P Woodpecker	50.0	53.5	**67.6**
RB Woodpecker	41.1	41.2	**45.2**
RC Woodpecker	19.5	30.8	**33.9**
RH Woodpecker	89.7	92.7	**95.8**
D Woodpecker	29.4	30.6	**36.8**
mAP (%)	38.9	40.6	**45.7**

the TriCoS[13], the UTL [40], the KDES [10] and two recent approaches [1,14]. Results of most methods are from the corresponding references, except that the performances of LLC and KDES were reported in [42] and [40] respectively.

As shown in the table, GMTL outperforms most of the compared FGVC approaches, which again confirms the effectiveness of our method. Compared with UTL, the gain is marginal. However, UTL focuses on feature representation, while our method emphasizes on the use of the class relationships during the learning phase. Since the UTL and KDES results are based the same SVM classification pipeline, we expect that similar improvement can be attained using our GMTL over the UTL feature, which however is difficult to validate as the source codes of UTL are not available online. Several recent works [1,14,17] reported better performance on this dataset. However, the approaches of [1,14] include computationally expensive segmentation/detection, and [17] requires additional human inputs, which is therefore excluded from the table. In addition, similar to

Table 4. Performance comparison on the entire Caltech-UCSD Bird-200-2010 dataset

Approach		Accuracy (%)
State-of-the-art FGVC methods	MKL [11]	19.0
	LLC [42]	18.0
	Randomization [42]	19.2
	Multi-Cue [26]	22.4
	TriCoS [13]	25.5
	UTL [40]	28.2
	KDES [10]	26.4
	Detection+Segmentation [1]	30.2
	Symb+DPM [14]	47.3
MTL methods	JFS [2]	21.7
	CMTL [45]	22.0
	Our GMTL	28.4

the observations from the experiment on the subset, JFS and CMTL fail again for the same reason as discussed earlier.

We also evaluate our method on the newer 2011 version of the Bird dataset, which contains more samples per category. Comparing to the baseline KDES using SVM (43.0%), our GMTL method achieves an accuracy of 44.2%. The improvement on this dataset (1.2%) is less significant than that on the 2010 version

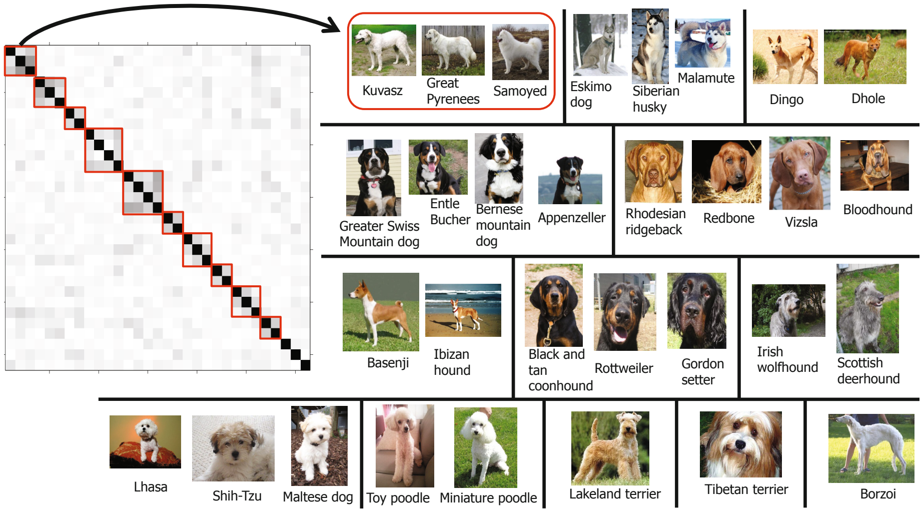


Fig. 5. Left: Similarity matrix of a subset of the learned tasks on the Stanford Dog dataset, where the red boxes indicate the automatically generated category groups. **Right:** Visual examples of the category groups (and the three outliers in the lower right corner) indicated on the similarity matrix, ordered from left to right and top to bottom. For instance, the first red box on the matrix corresponds to the upper left group of the example images.

(2.0%), indicating that our method is more effective when there is insufficient training data.

Visualization of Category Groups and Outliers. Finally, we analyze the power of our GMTL in identifying the inter-class relationships, including both the category groups and the outliers. The Stanford Dog dataset is adopted in this study for the ease of visualization as it contains less categories. We use $\mathbf{W}^\top\mathbf{W}$ to represent the similarity matrix of all the categories, and adopt the Normalized Cut [31] to group the categories, which are then reordered for visualization. A subset of the similarity matrix is displayed in Figure 5, which shows ten category groups and three outliers. Within each category group, the different species of dogs share similar color, shape and texture, while there exist significant differences across different groups.

5 Conclusion

We have presented a generic MTL method to explore inter-class relationships for improved fine-grained visual categorization. Different from the existing MTL algorithms that often rely on certain assumptions of the task structure, the proposed GMTL imposes no structural assumptions, making it more flexible to handle complex category relationships in the FGVC applications. We have shown that the training of our GMTL can be efficiently achieved using an iterative reweighted ℓ_2 method. The learned classification models enforce feature sharing within each automatically discovered category group, which leads to better discriminative power. Results on two standard benchmarks have clearly demonstrated the effectiveness of our proposed method. One promising future direction is to jointly learn visual representations and classification models under the GMTL framework for further performance improvements.

Acknowledgments. This work was supported in part by a National 863 Program (#2014AA015101), a Key Technologies Research and Development Program (#2013BAH09F01), a grant from the National Natural Science Foundation of China (#61201387), four grants from the Science and Technology Commission of Shanghai Municipality (#13PJ1400400, #13511504503, #12511501602 and #14511106900), and a grant from Ministry of Education (#20120071120026), China.

References

1. Angelova, A., Zhu, S.: Efficient object detection and segmentation for fine-grained recognition. In: CVPR (2013)
2. Argyriou, A., Evgeniou, T., Pontil, M.: Multi-task feature learning. In: NIPS (2007)
3. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Mach. Learn. 73(3), 243–272 (2008)
4. Babenko, B., Branson, S., Belongie, S.: Similarity metrics for categorization: From monolithic to category specific. In: ICCV (2009)

5. Bar-Hillel, A., Weinshall, D.: Subordinate class recognition using relational object models. In: NIPS (2006)
6. Bart, E., Porteous, I., Perona, P., Welling, M.: Unsupervised learning of visual taxonomies. In: CVPR (2008)
7. Beck, A., Teboulle, M.: A fast iterative shrinkage-thresholding algorithm for linear inverse problems. SIAM J. Img. Sci. 2(1), 183–202 (2009)
8. Berg, T., Belhumeur, P.N.: How do you tell a blackbird from a crow? In: ICCV (2013)
9. Berg, T., Liu, J., Lee, S.W., Alexander, M.L., Jacobs, D.W., Belhumeur, P.N.: Birdsnap: Large-scale fine-grained visual categorization of birds. In: CVPR (2014)
10. Bo, L., Ren, X., Fox, D.: Kernel Descriptors for Visual Recognition. In: NIPS (2010)
11. Branson, S., Wah, C., Schroff, F., Babenko, B., Welinder, P., Perona, P., Belongie, S.: Visual recognition with humans in the loop. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part IV. LNCS, vol. 6314, pp. 438–451. Springer, Heidelberg (2010)
12. Caruana, R.: Multitask learning. Mach. Learn. 28(1), 41–75 (1997)
13. Chai, Y., Rahtu, E., Lempitsky, V., Van Gool, L., Zisserman, A.: TriCoS: A tri-level class-discriminative co-segmentation method for image classification. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part I. LNCS, vol. 7572, pp. 794–807. Springer, Heidelberg (2012)
14. Chai, Y., Lempitsky, V., Zisserman, A.: Symbiotic segmentation and part localization for fine-grained categorization. In: ICCV (2013)
15. Chapelle, O.: Training a support vector machine in the primal. Neural. Comput. 19(5), 1155–1178 (2007)
16. Chen, J., Zhou, J., Ye, J.: Integrating low-rank and group-sparse structures for robust multi-task learning. In: KDD (2011)
17. Deng, J., Krause, J., Fei-Fei, L.: Fine-grained crowdsourcing for fine-grained recognition. In: CVPR (2013)
18. Duan, K., Parikh, D., Crandall, D., Grauman, K.: Discovering localized attributes for fine-grained recognition. In: CVPR (2012)
19. Farrell, R., Oza, O., Zhang, N., Morariu, V.I., Darrell, T., Davis, L.S.: Birdlets: Subordinate categorization using volumetric primitives and pose-normalized appearance. In: ICCV (2011)
20. Fergus, R., Bernal, H., Weiss, Y., Torralba, A.: Semantic label sharing for learning with many categories. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 762–775. Springer, Heidelberg (2010)
21. Gavves, E., Fernando, B., Snoek, C.G.M., Smeulders, A.W.M., Tuytelaars, T.: Fine-grained categorization by alignments. In: ICCV (2013)
22. Gong, P., Ye, J., Zhang, C.: Robust multi-task feature learning. In: KDD (2012)
23. Griffin, G., Perona, P.: Learning and using taxonomies for fast visual categorization. In: CVPR (2008)
24. Jalali, A., Ravikumar, P.D., Sanghavi, S., Ruan, C.: A dirty model for multi-task learning. In: NIPS (2010)
25. Kang, Z., Grauman, K., Sha, F.: Learning with whom to share in multi-task feature learning. In: ICML (2011)
26. Khan, F.S., Van De Weijer, J., Bagdanov, A.D., Vanrell, M.: Portmanteau vocabularies for multi-cue image representation. In: NIPS (2011)
27. Khosla, A., Jayadevaprakash, N., Yao, B., Fei-Fei, L.: Novel dataset for fine-grained image categorization. In: First Workshop on FGVC, CVPR (2011)
28. Kumar, A., Daumé III, H.: Learning task grouping and overlap in multi-task learning. In: ICML (2012)

29. Melacci, S., Belkin, M.: Laplacian Support Vector Machines Trained in the Primal. JMLR 12, 1149–1184 (2011)
30. Salakhutdinov, R., Torralba, A., Tenenbaum, J.: Learning to share visual appearance for multiclass object detection. In: CVPR (2011)
31. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Trans. Pattern Anal. Mach. Intell. 22(8), 888–905 (2000)
32. Su, H., Yu, A.W., Fei-Fei, L.: Efficient euclidean projections onto the intersection of norm balls. In: ICML (2012)
33. Todorovic, S., Ahuja, N.: Learning subcategory relevances for category recognition. In: CVPR (2008)
34. Wah, C., Branson, S., Perona, P., Belongie, S.: Multiclass recognition and part localization with humans in the loop. In: ICCV (2011)
35. Wang, H., Nie, F., Huang, H., Risacher, S.L., Ding, C.H.Q., Saykin, A.J., Shen, L.: Adni: Sparse multi-task regression and feature selection to identify brain imaging predictors for memory performance. In: ICCV (2011)
36. Wang, J., Yang, J., Yu, K., Lv, F., Huang, T.S., Gong, Y.: Locality-constrained linear coding for image classification. In: CVPR (2010)
37. Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., Perona, P.: Caltech-UCSD Birds 200. Tech. Rep. CNS-TR-2010-001, California Institute of Technology (2010)
38. Wipf, D.P., Nagarajan, S.S.: Iterative reweighted ℓ_1 and ℓ_2 methods for finding sparse solutions. J. Sel. Topics Signal Processing 4(2), 317–329 (2010)
39. Xie, L., Tian, Q., Hong, R., Yan, S., Zhang, B.: Hierarchical Part Matching for Fine-Grained Visual Categorization. In: ICCV (2013)
40. Yang, S., Bo, L., Wang, J., Shapiro, L.: Unsupervised Template Learning for Fine-Grained Object Recognition. In: NIPS (2012)
41. Yao, B., Bradski, G., Fei-Fei, L.: A codebook-free and annotation-free approach for fine-grained image categorization. In: CVPR (2012)
42. Yao, B., Khosla, A., Fei-Fei, L.: Combining randomization and discrimination for fine-grained image categorization. In: CVPR (2011)
43. Zhang, N., Farrell, R., Darrell, T.: Pose pooling kernels for sub-category recognition. In: CVPR (2012)
44. Zhang, N., Farrell, R., Iandola, F., Darrell, T.: Deformable part descriptors for fine-grained recognition and attribute prediction. In: ICCV (2013)
45. Zhou, J., Chen, J., Ye, J.: Clustered multi-task learning via alternating structure optimization. In: NIPS (2011)
46. Zweig, A., Weinshall, D.: Hierarchical regularization cascade for joint learning. In: ICML (2013)