Fine-Grained Image Categorization by Localizing Tiny Object Parts from Unannotated Images

Luming Zhang†, Yi Yang‡, and Roger Zimmermann†
†School of Computing, National University of Singapore, Singapore
‡Centre for Quantum Computation&Intelligent Systems,
the University of Technology, Sydney.

ABSTRACT

This paper proposes a novel fine-grained image categorization model where no object annotation is required in the training/testing stage. The key technique is a dense graph mining algorithm that localizes multiscale discriminative object parts in each image. In particular, to mimick human hierarchical perception mechanism, a superpixel pyramid is generated for each image, based on which graphlets from each layer are constructed to seamlessly describe object parts. We observe that graphlets representative to each category are densely distributed in the feature space. Therefore a dense graph mining algorithm is developed to discover graphlets representative to each sub-/super-category. Finally, the discovered graphlets from pairwise images are encoded into an image kernel for fine-grained recognition. Experiments on the UCB-200 [32] shown that our method performs competitively to many models relying on the annotated bird parts.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software

Keywords

Fine-grained; graph mining; hierarchical; perception; image kernel

1. INTRODUCTION

A large number of object recognition models have been developed in multimedia retrieval and analysis. Many of them focus on discriminatively learning to distinguish objects belonging to different basic-level categories. Inspired by applications in areas such as agriculture, medicine, and forestry, fine-grained domain recognition has become a hot research topic recently. For example, some Apps have been developed to recognize different species of pests, based on which suitable chemicals can be employed to eliminate them. Intuitively, successfully recognizing objects from multiple sub-categories is a difficult task. Even a knowledgeable person might be confused to distinguish poisonous/non-poisonous mushrooms. In summary, existing approaches suffer from the following challenges:

ICMR'15, June 23–26, 2015, Shanghai, China.

DOI: http://dx.doi.org/10.1145/2671188.2749299.

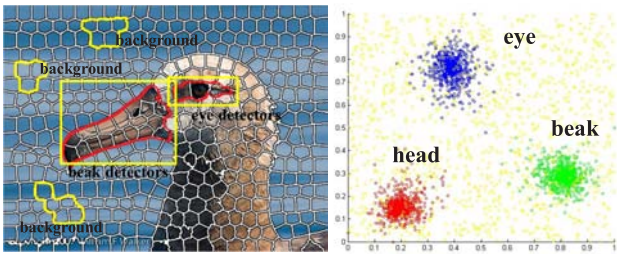

Figure 1: Left: bird beak/eye detectors built upon graphlets (red) and rectangular window (yellow) respectively; Right: distribution of graphlets from the sub-category "black foot albatross" (the yellow points denote the non-discriminative bird parts).

- Many fine-grained recognition models are part-based models, where object parts are either manually annotated or discovered by a set of object detectors. For the former, the annotation quality is good but the annotation process is labor-intensive. For the latter, a set of object component detectors are pre-defined for a specific data set, making the model difficult to be transferred from one data set to another.

- Fine-grained recognition discriminates similar objects with subtle differences. It requires the recognition model can discover the discriminative object details. Typically, a scanning window is used for part detection. However, the rectangular scanning window cannot well capture those arbitrarily shaped object components, such as the bird beak in Figure 1.

To solve the above problems, this paper presents hierarchical graphlet matching (HGM) for fine-grained image categorization. The key advantage is that discriminative object parts with different scales can be localized automatically from unannotated images. As shown in Figure 2, to mimick the coarse-to-fine visual perception of humans, a KL-divergence-based clustering is employed to construct the hierarchy of the super-/sub-categories. To seamlessly describe an object in each hierarchy layer, we construct a superpixel pyramid and further propose graphlets by connecting spatially connected superpixels from each pyramid layer. As the number of graphlets is huge, only those discriminative for fine-grained categorization should be preserved. As shown on the right of Figure 1, **highly discriminative graphlets to each super-/sub-category are densely distributed in the feature space, as these graphlets are similar**

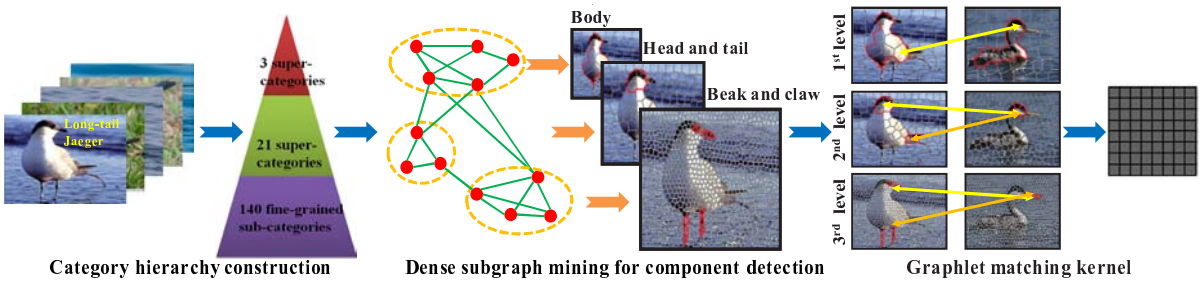

Figure 2: The pipeline of the proposed fine-grained image categorization model

in appearance. Therefore, an affinity graph is constructed to describe the similarity of graphlets to each super-/sub-category, based on which dense subgraphs containing the discriminative graphlets are selected. Finally, an image kernel is calculated by matching the selected graphlets from pairwise images hierarchically.

The contributions of this paper are two-fold: 1) the first fine-grained recognition model discovering multiscale discriminative object parts from unannotated training and testing images; and 2) a graphlet matching kernel that stimulates human hierarchial visual perception.

2. RELATED WORK

Our work is closely related to the graph-based image modeling in multimedia retrieval [38, 39, 40, 41]. As a natural binary relationship descriptor, graphical models are frequently used to exploit the geometric property of different regions in an image. In [5], Harchaoui *et al.* proposed walk-/tree-kernel that capture the walk/tree structures among local image regions using a finite sequence of neighboring regions. In [6], Duchenne *et al.* proposed a graph matching kernel for object categorization. That is, the graph vertices correspond to a set of image grids, and the edges reflect the grid structure, functioning as springs preserving the geometry of neighboring grids. Lin *et al.* [7] introduced an object categorization framework based on sketch graphs, *i.e.*, a learnable And-Or graph model. Further, in [8], Zhang *et al.* proposed to measure the similarity between aerial images by selectively matching their respective graphlets. Note that all these graphical models exploit geometric descriptors for basic-level categories, which fail to capture the detailed visual cues discriminative to sub-categories. Besides, many of them rely on the prior knowledge of a specific data set, limiting the application across different data sets.

Recently, many fine-grained categorization models have been developed. Most of them focus on learning part detectors from training annotated images [10, 11, 12, 14, 45, 34, 35, 36, 37], or localizing distinctive object details by human interaction [15, 17]. A few approaches are reviewed as follows. Yao *et al.* [12] represented an image by pooling template matching responses. The templates are sampled from $1.5\times$ the size of the object bounding box. In [14], Berg *et al.* proposed a grid-level saliency model for describing fine-grained image categories. Annotated object parts are aligned and cropped to indicate the discriminative parts. Deng *et al.* [15] designed a human interactive crowdsourcing system allowing users to localize discriminative object parts. Angelova *et al.* [18] "zoom in" the foreground objects segmented from an image for fine-grained recognition. Compared to the previous fine-grained models, our approach is free from annotated object parts in both the training and the testing stages. Zhang *et al.* [44] proposed a one-vs-all mid-level features for fine-grained recognition. The method is dimension friendly since the dimension of learned mid-level features is only related with the number of classes and far less than that of the low level ones.

3. CATEGORY HIERARCHY CONSTRUCTION

3.1 Topological Object Descriptors

As aforementioned, besides the large object components reflecting the basic-level category, fine-grained recognition depends on an accurate description of the tiny object parts. In our work, both the large objects and their tiny parts are constructed from superpixels, which align neatly with object boundaries. More specifically, objects and their parts are described by graphlets:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}), \tag{1}$$

where $\mathcal{V}$ is a set of vertices, each representing a superpixel; $\mathcal{E}$ is a set of edges, each connecting pairwise spatially adjacent superpixels. The number of constituent superpixels of a graphlet is called the graphlet size. As shown in Figure 3, graphlets well capture different object parts, which are descriptive to each sub-category.

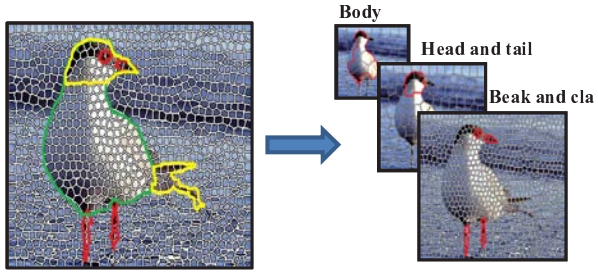

Figure 3: Left: A single superpixel mosaic describing a bird and its parts; Right: A superpixel pyramid where differently sized object parts are described in different layers.

3.2 KL-divergence-based Category Hierarchy

To mimick the hierarchical perception of humans, we cluster the sub-categories into super-categories. We first measure the similarity between images from two sub-categories. Unfortunately, global feature-based distance or unselectively local feature matching is not suitable here. We want to compare the object parts that are representative to each sub-category. For pairwise images, we first establish the matchings between their SIFT descriptors. Among all the SIFT descriptor matchings, some are between patches from the foreground objects, while the rest are between patches from the backgrounds. As shown in Figure 4, if we model the distribution

of all the pairwise matched SIFT descriptors from a sub-category, those from the foreground objects will closely distributed since they have similar appearance. While SIFT descriptors from the background will distribute dispersedly, since the patches can have arbitrary appearances. This reveals that the divergence between SIFT distribution from two sub-categories can well distinguish two sub-categories. In this work, the distribution of the matched SIFT descriptors from a sub-category is modeled by a GMM $\mathcal{N}$. Then, the similarity between sub-categories can be described by the KL-divergence between their SIFT distributions. Due to the non-symmetry of KL-divergence, it is difficult to integrate it into a semi-definite matrix for clustering. Instead, we use the square root of Jensen-Shannon divergence [22], a symmetric metric derived from KL-divergence.

Based on the Jensen-Shannon divergence, we encode it into

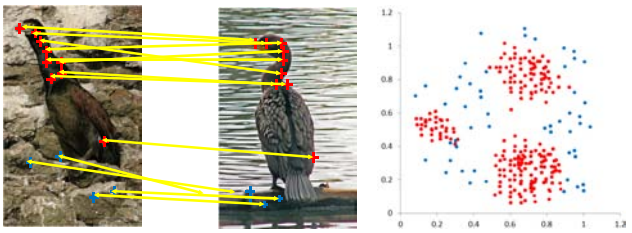

Figure 4: Left: SIFT descriptor matching between images; Right: distribution of matched SIFT descriptors (red: objects, blue: background).

spectral clustering for building the category hierarchy. More specifically, we construct an affinity matrix $\mathbf{W}$ wherein the ij-th element is computed as:

$$\mathbf{W}_{ij} = \exp\left(-\frac{D_{JS}(\mathcal{N}_i||\mathcal{N}_j)}{2\psi^2}\right), \quad (2)$$

where $\mathcal{N}_i$ and $\mathcal{N}_j$ denote the SIFT distributions of i-th and the j-th sub-categories respectively, and D_{JS} is the Jensen-Shannon divergence. Then, we calculate the Laplacian matrix accordingly and group the sub-categories into H parent-categories[1]. We empirically observe that a three layer hierarchy achieves a good performance. And the number of categories from the i-th layer is fixed to 1/7 to that from the $(i+1)$-th layer $(i = 1, 2)$.

4. DISCRIMINATIVE OBJECT PARTS DISCOVERY

Obviously, discriminative object parts from each sub-category (*e.g.*, the bird beaks) distribute densely in the feature space (can be quantified into an affinity graph). In contrast, non-discriminative object components distribute dispersedly. Thus, a dense graph mining framework is proposed to discover graphlets representative to different sub-/super-categories, as the pipeline shown in Figure 5.

4.1 Affinity Graph Construction

To construct an affinity graph that describes the similarity between graphlets, a similarity measure is required. In the color channel, the appearance similarity between graphlets can be measured by a Gaussian kernel, *i.e.*, $\mathbf{S}^c_{i,j} \propto \exp\left(-||x^c_i - x^c_j||^2/\sigma^2\right)$, where x^c_i describes the color channel graphlets by a 9-dimensional color moment (CM) [25], and σ^2 is the empirical variance.

[1]In this work, parent-categories are super-categories one layer higher than the those in the current layer.

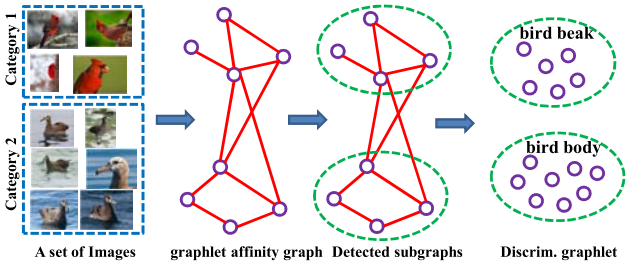

Figure 5: Detecting discriminative object parts using the dense graph mining technique. Each purple vertex denotes a graphlet and each edge connects pairwise similar graphlets.

Though it is common to use color channel as the similarity metric for object classification. It is not sufficiently descriptive for our object parts detection. HOG [24] based sliding window is standard for object detection. Thus, the similarity between graphlets is calculated by combining both the CM and the HOG descriptors:

$$\mathbf{S}_{i,j} = \mathbf{S}^c_{i,j} \cdot \mathbf{S}^t_{i,j}, \quad (3)$$

where $\mathbf{S}^t_{i,j}$ denotes the similarity in the textural channel, which is computed similarly to $\mathbf{S}^c_{i,j}$.

4.2 Mining Dense Subgraphs by Graph Shift

To effectively discover dense subgraphs from an affinity graph, two conditions are required. First, *compatibility with graph representation*: many similarity metrics are defined based on binary relation, such as our color+texture-based similarity. Only graph-based clustering can utilize this pairwise relation directly. Second, *robustness to outliers*: many graphlets, such as those from the background and highly occluded, may not belong to any sub-category. Methods insisting on partitioning all input data into coherent groups without explicit outliers may fail to preserve the structure of sub-categories. Conventional clustering algorithms, *e.g.*, k-means, are not suitable here as they insist on partitioning all the input data. Comparatively, graph shift [26], which is efficient and robust for graph mode seeking, is particularly suitable for the discriminative graphlet mining. It directly works on graph, supports an arbitrary number of clusters, and leaves the outlier points ungrouped.

Formally, we define an individual graph $\mathbf{G} = (\mathbf{V}, \mathbf{A})$ for each category, $\mathbf{V} = \{v_1, v_2, \cdots, v_{N_i}\}$ is a set of vertices denoting the graphlets extracted from images in category i. $\mathbf{A}$ is a symmetric matrix with non-negative elements. The diagonal elements of $\mathbf{A}$ are one while the non-diagonal element measures the similarity between graphelts, as detailed in (3). The modes of a graph $\mathbf{G}$ are defined as local maximizers of graph density function $g(y) = y^T\mathbf{A}y$, $y \in \Delta^{N_i}$, where $\Delta^{N_i} = \{y \in \mathbb{R}^{N_i} : y \geq 0 \text{ and } ||y||_{l_1} = 1\}$. More specifically, the similarity between graphlets is expressed as the edge weights of graph $\mathbf{G}$. The vertices represent the graphlets corresponding to a category. Therefore, discriminative object parts correspond to vertices of those strongly connected subgraphs. It is worth emphasizing that those strongly connected subgraphs correspond to large local maxima of $g(y)$ over simplex, which is an approximation of the average affinity score of these subgraphs.

The target patterns are the local maximizers of $g(y)$, which are detected by solving the quadratic optimization problem as follows:

$$\max_y g(y) = y^T\mathbf{A}y, \quad s.t. \ y \in \Delta^n, \quad (4)$$

Obtaining an analytic solution of (4) is difficult. Therefore, we employ replicator dynamics to find the local maxima of (4). Given

an initialization $y(0)$, the corresponding local solution y^* can be iteratively computed by the discrete-time version of the first-order replicator equation:

$$y_i(t+1) = y_i(t)\frac{(\mathbf{A}y(t))_i}{y(t)^T\mathbf{A}y(t)}. \quad (5)$$

5. HIERARCHICAL GRAPHLET MATCHING KERNEL

After discovering the discriminative graphlets for each sub-/super-category, each image can be represented by a collection of planar visual features in $\mathbb{R}^2$. Unfortunately, conventional classifiers such as SVM can only handle 1-D vector form features. Moreover, the number of discovered graphlets from each image varies. Thus, it would be impractical for a conventional classifier such as SVM to carry out classification directly. To tackle this problem, a hierarchical graphlet kernel is calculated by quantizing the extracted graphlets from an image into a 1-D vector.

The proposed quantization method is built upon the Euclidean

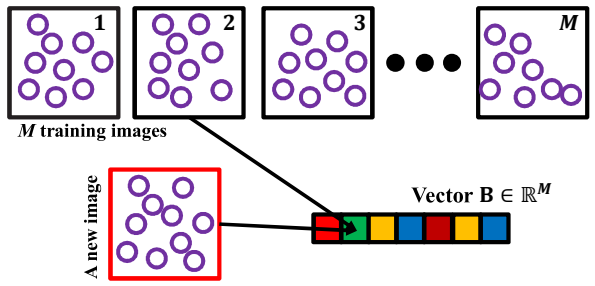

Figure 6: Hierarchical graphlet matching kernel calculation.

distance between images, which is computed based on the extracted graphlets. As shown in Figure 6, given an image, discriminative graphlets corresponding to sub- and super-categories are extracted. The extracted graphlets are then converted into a vector $\mathbf{B} = [\beta_1, \beta_2, \cdots, \beta_M]$, where each element of $\mathbf{B}$ is computed as:

$$\beta_i \propto \exp\left(-\frac{1}{T\cdot T'}\sum_{i=1}^{3}\sum_{\substack{G\in\mathcal{H}_i \\ G'\in\mathcal{H}_i'}} d_E(f(G), f(G'))\right), \quad (6)$$

where $d_E(\cdot,\cdot)$ is the Euclidean distance; M is the number of training images; T and T' denote the number of discovered graphlets in image I and I'; $\mathcal{H}_i$ and $\mathcal{H}_i'$ contains the discovered graphlets in the i-th level category, from image I and I' respectively; $f(G)$ and $f(G')$ are the combined color and texture descriptors of graphlet G and G'.

Based on the feature vector $\mathbf{B}$, a multi-class SVM is trained for fine-grained label prediciton. For the training images from the p-th and the q-th sub-categories, we construct a binary SVM classifier as:

$$\max_{\alpha\in\mathbb{R}^{M_{pq}}} W(\alpha) = \sum_{i=1}^{M_{pq}}\alpha_i - \frac{1}{2}\sum_{i=1}^{M_{pq}}\alpha_i\alpha_j l_i l_j k(\mathbf{B}_i, \mathbf{B}_j)$$
$$s.t. \quad 0 \le \alpha_i \le C, \quad \sum_{i=1}^{M_{pq}}\alpha_i l_i = 0, \quad (7)$$

where $\mathbf{B}_i \in \mathbb{R}^M$ is the quantized vector from the i-th training image; l_i is the category label (+1 for the p-th sub-category and -1 for the q-th sub-category) to the i-th training image; α determines the hyper-plane that separates images in the p-th sub-category from those in the q-th sub-category; $C > 0$ trades the complexity of the machine off the number of nonseparable images; and M_{pq} is the number of training images either from the p-th or from the q-th sub-category.

We summarize the procedure of our fine-grained image categorization framework in Algorithm 1.

Algorithm 1 Fine-grained Image Categorization based on Dense Subgraph Mining

//**Training stage**:
Input: Training images $\{I_1, I_2, \cdots, I_M\}$ and their sub-category labels;
Output:Discovered object parts, and the trained SVM classifier.
1) Extract graphlets from the M training images, and build a category hierarchy based on KL-divergence;
2) Construct an affinity graph to model the similarity between graphlets, based on (3). Then apply dense graph mining technique to discover discriminative graphlets to each sub-/super-category;
3) Calculate an image kernel using the discovered graphlets from the training data using (6); learn a multi-class SVM.
//**Testing stage**:
Input: A test image I_{test}; **Output**: the sub-category label of I_{test};
1) Extract graphlets from the image I_{test}, discover the discriminative object parts by dense graph mining;
2) Compute the vector representation of I_{test} based on (6), and use the trained SVM to predicted its sub-category label.

6. EXPERIMENTS

In this section, we evaluate our fine-grained recognition model based on four experiments. The first experiment compares our approach with some previous categorization models. Then, we testify the important components in our model. Third, we evaluate the influence of different parameter settings. Finally, we visualize the detected discriminative graphlets, which can explain the impressive performance of our approach.

We experiment on four popular fine-grained data sets: the CUB-200 [32], the Standford dogs [20], the Oxford flowers [21], and the Leeds butterflies [33]. Different from the other fine-grained categorization models we compared, our approach **does not** require the object or object part annotations in the four data sets.

6.1 A Comparative Study

This section compares our method with 1) a series of SPM-based generic classification models, and 2) several well-known fine-grained recognition models.

We first compare our method with SPM [27] and its two variants: SC-SPM [28] and LLC-SPM [29]. The parameter settings are as follows. For SPM, SC-SPM, and LLC-SPM, we construct a three layer spatial pyramid. Then we extract nearly one million SIFT descriptors from 16×16 patches computed over a grid with spacing of eight pixels for all the training images. Finally, a codebook with size 256 is generated by k-means clustering on the one million SIFT descriptors. For our method, a three layer superpixel pyramid is constructed. Each image is represented by nearly 150 (*i.e.*, 110 (the 3rd layer)+ 30 (the 2nd layer)+10 (the 1st layer)) graphlets mined from the superpixel hierarchy. For each of the three data sets, 30% and 50% images are used for training respectively, while the rest are for testing. We report the categorization accuracy in Table 1. As can be seen, the proposed method outperforms SPM and its two variants. This is because the discovered graphlets from multiple pyramid layers are more descriptive to object parts than the grids in the SPM. Further, our algorithm performs better than Russakovsky *et al.* [4]. This reflects the advantage of mining object-shape parts for fine-grained categorization.

Besides, we compare our method with five recent fine-grained categorization models that are proposed by Yao *et al.* [12], Berg *et al.* [14], Duan *et al.* [17], Angelova *et al.* [18], and Zhang *et al.* [31], respectively. The parameters are the same as in the publication.

Table 1: Comparison of of our approach with SPM-based models

	Method	CUB-200	Leeds	SF dogs	OF flowers
30% train	SPM	35.4%	31.6%	17.65%	62.23%
	SC-SPM	38.9%	32.4%	18.23%	63.35%
	LLC-SPM	38.7%	31.9%	18.01%	64.11%
	Russakovsky	41.9%	36.4%	19.26%	65.58%
	Our	**44.1%**	**39.6%**	**20.6%**	**68.87%**
50% train	SPM	40.1%	36.4%	17.65%	63.2%
	SC-SPM	44.3%	39.1%	18.42%	70.34%
	LLC-SPM	44.1%	37.7%	19.68%	72.13%
	Russakovsky	47.8%	74.32%	41.3%	19.38%
	Our	**49.1%**	**47.7%**	**25.78%**	**76.47%**

Different proportion of training images are adopted by selecting 30%, 50%, and 70% training images respectively. As shown in Figure 7, our approach outperforms its competitors. Noticeably, the first three approaches depend on the bird parts annotation during training. However, our method can detect tiny object parts from unannotated images automatically.

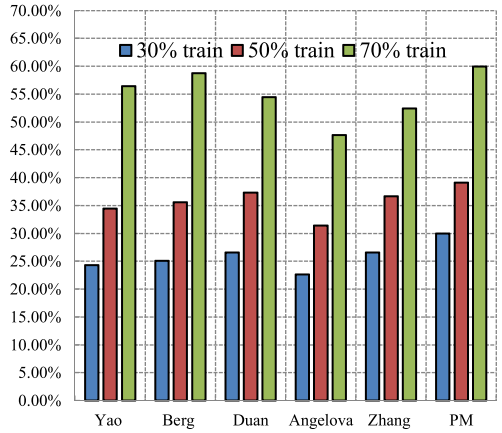

Figure 7: Comparison of the five state-of-the-art fine-grained categorization models on the CUB-200

6.2 Important Components Evaluation

In this experiment, we evaluate the two key components in our method: 1) the KL-divergence-based category hierarchy, and 2) the dense-graph-mining-based discriminative graphlets discovery.

Component 1 We calculate the distribution of the matched SIFT descriptors from images in each sub-category. For the testing images in the CUB-200, we manually annotate the distinctive object parts, *i.e.*, bird beak, head, body, tail and claw. Then, graphlets corresponding to the five parts are represented as differently colored points in the scatter plots. As shown in the first two rows of Figure 8, we select 10 scatter plots from the 200 bird sub-categories. As can be seen, in each sub-category, SIFT descriptors from discriminative object parts are densely distributed. And the SIFT distribution of each sub-category is unique. Therefore, the KL-divergence between SIFT distribution of two sub-categories can quantify their similarity. In addition, we visualize SIFT distribution by combining sub-categories randomly. As shown in the last row of Figure 8, SIFT descriptors from discriminative bird parts are dispersedly scattered, since the appearance of the discriminative bird parts are different across sub-categories.

Component 2 To evaluate the second component, we compare the affinity graph constructed based on different local descriptors in each image: $10 \times 10/20 \times 20$ grids respectively, and 100/400 superpixels respectively. As the scatter plots shown in Figure 9, different discriminative object parts are densely distributed in the constructed affinity graph. These discriminative object patterns can be easily discovered by graph shift [26]. Comparatively, affinity graphs generated using the above four schemes are sub-optimal, as different object parts are mixed. Besides the qualitative results, we further calculate the ratio of accumulated distance within and between sub-categories as in LDA [19]. As shown in Table 2, the lowest ratio is achieved by our graphlet-based affinity graph.

6.3 Parameters and Time Consumption

This subsection evaluates the performance of our approach under different parameter settings: 1) the structure of the category hierarchy; and 2) the number of outliers (*i.e.*, non-discriminative graphlets) in graph shift [26].

First, we evaluate our approach under different ratios between

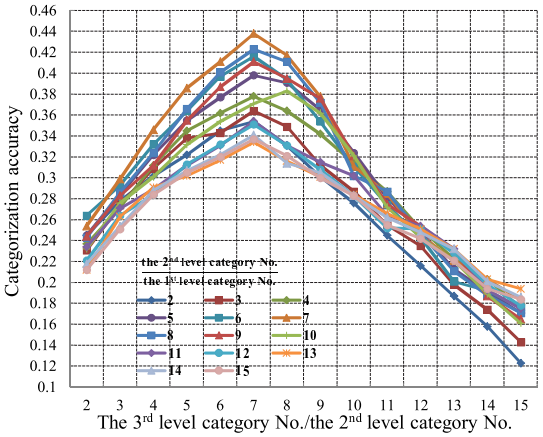

Figure 10: Influence of fine-grained categorization under the influence of the structure of the three layer category hierarchy.

the number of categories from the $(i+1)$-th layer and that from the i-th layer ($i = 1, 2$). As shown in Figure 10, when the ratio between the third and the second layer is tuned from 2 to 15, the categorization accuracy increases and peaks when the ratio is seven or eight. Then, the category accuracy decreases to a low level. Further, we notice that the best performance is achieved when the ratio between the second and the first layer is seven. Based on this, we set the ratio between the $(i+1)$-th layer and that from the i-th layer to seven in our experiment.

Second, we evaluate our approach when different numbers of outliers are adopted in the graph shifting [26] algorithm. As shown in Figure 11, when a small or a moderate number of outliers are used, our graph-shifting mining outperforms the conventional clustering algorithms such as k-means [23] and spectral clustering [30]. This is consistent with the theoretical analysis in Sec 4.2. When a large number of outliers are used, some discriminative graphlets are abandoned in the fine-grained categorization, which hampers the fine-grained categorization.

The time consumption analysis of the proposed method is as follows. All experiments are carried out on a computer equipped with an Intel Xeon X5482 CPU and 8GB RAM. All the comparative algorithms are implemented on the Matlab 2011 platform. Take the CUB-200 [32] for example, there are about 12,000 images in total. We use a half images for training. For each image, it takes about 0.12 seconds to extract all the graphlets (1330 graphlets in each image on average) based on the depth-first-search [43]. Then, it takes

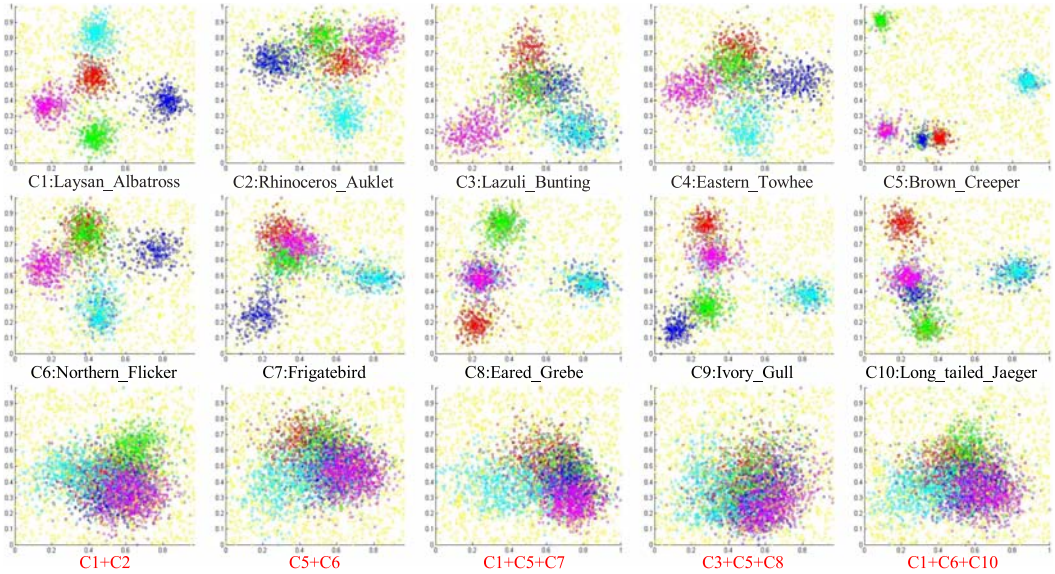

Figure 8: The first two rows: Distribution of SIFT descriptors from images belonging to the same sub-category. The last row: Distribution of SIFT descriptors from images from two and three sub-category (red points: beak, pink points: head, blue points: body, green points: tail, and cyan points: claw).

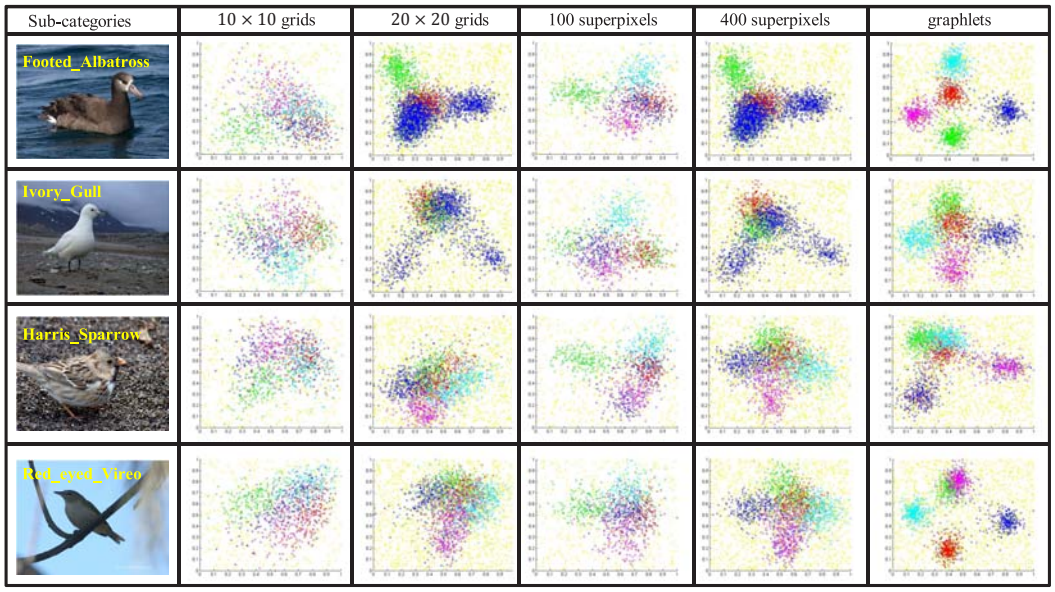

Figure 9: Affinity graphs generated using different schemes.

Table 2: The ratio $\phi = S_w/S_b$ of affinity graphs generated using different schemes

Data set	10×10 grids	20×20 grids	100 superpixels	400 superpixels	graphlets
CUB-200	0.0121	0.0171	0.008	0.0076	**0.0042**

Table 3: Comparison of Time Consumption of Different Categorization Models

Model	SPM	SC-SPM	LLC-SPM	Yao *et al.*	Berg *et al.*	Duan *et al.*
Time	0.76 seconds	1.12 seconds	0.87 seconds	2.34 seconds	1.05 seconds	1.37 seconds
Model	Algelova *et al.*	Zhang *et al.* [31]	Walk kernel	Tree kernel	PM	
Time	2.13 seconds	1.34 seconds	2.87 seconds	3.34 seconds	**0.51 seconds**	

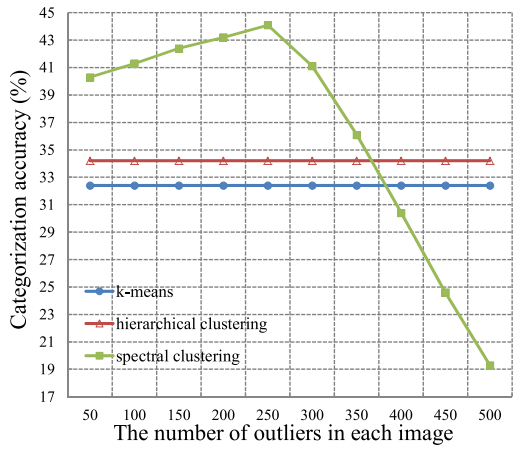

Figure 11: Performance under different number of outliers in the graph-shift-based discriminative graphlet discovery.

about 0.16 seconds to construct the KL-divergence-based category hierarchy. The dense graph mining for all training graphlets takes 5.23 minutes. Finally, the graphlet-based kernel construction and SVM learning take 1.21 minutes. Therefore, our training stage consumes about 20 minutes. Comparatively, the test stage is carried out rapidly. Given a test image, it takes 0.12 seconds to extract the graphlets and 0.31 seconds to detect the discriminative graphlets. Then, it consumes 0.06 seconds to calculate the feature vector as shown in Figure 6. Finally, the SVM classification takes 0.02 seconds. In total, it takes 0.51 seconds to predict the category label of a test image. In Table 3, we present the average time consumption of categorizing an image using different models. Our model requires the lowest time to predict the fine-grained category label for an image.

6.4 Visualization of the Discovered Graphlets

This subsection visualizes the discovered discriminative object parts from the CUB-200 [32]. As shown in Figure 12, by utilizing the dense graph mining technique, our approach seamlessly constructs the discriminative/salient objects in each image. Discriminative object parts with multiple sizes are captured by superpixel hierarchy from multiple layers. Our approach can accurately localize tiny object parts and then seamlessly capture them by graphlets. This is the underlying reason why the fine-grained categorization accuracy can be greatly improved by our approach.

7. CONCLUSIONS

This paper presents a novel fine-grained image categorization algorithm. The key is a graph mining algorithm that detects discriminative object parts with multiple scales. By extracting graphlets from the training images, an affinity graph was constructed to describe their similarity. Since discriminative object parts are densely distributed in each sub-category, we used graph shift [26] to mine them efficiently. Finally, these discovered object parts were encoded into an image kernel for categorization.

8. APPENDIX

Theoretically, our hierarchical graphlet-based image kernel is an extension of many existing image kernels. This can uncover the good performance of our method.

The proposed image kernel is closely related to several graph-based image kernels: walk-/path-/tree-kernels [5] and the two graphlet kernels [8, 42]. As elaborated in Figure 13, graphlet can generalize topologies like path, walk, and tree. Each of the three topologies have certain constraints (*e.g.*, the hierarchical structure of the tree). In addition, in Zhang *et al.* [42]'s model, all the graphlets from an image are integrated into the kernel. Some graphlets might be non-discriminative or even noisy. Moreover, Zhang *et al.* [8] selects graphlets in the topology level. This scheme is less effective as graphlets with the same topology may have different discrimination. In summary, the graphlet kernel in our work either generalizes or improves the previous graph-based image kernels.

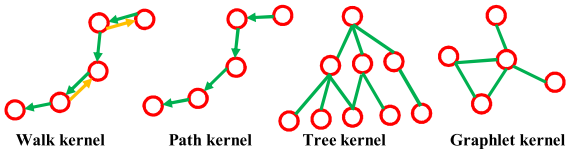

Figure 13: Topologies in different image kernels (The yellow arrows in the walk kernel indicate that each vertex can be revisited.)

9. ACKNOWLEDGMENTS

This research has been supported by the Singapore National Research Foundation under its International Research Centre@ Singapore Funding Initiative and administered by the IDM Programme Office through the Centre of Social Media Innovations for Communities (COSMIC).

10. REFERENCES

[1] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, Yihong Gong, Locality-constrained Linear Coding for Image Classification, *CVPR*, 2010.

[2] Li-Jia Li, Hao Su, Eric P. Xing, Li Fei-Fei, Object Bank: A High-Level Image Representation for Scene Classification and Semantic Feature Sparsification, *NIPS*, 2010.

[3] Yangqing Jia, Chang Huang, Trevor Darrell, Beyond Spatial Pyramids: Receptive Field Learning for Pooled Image Features, *CVPR*, 2012.

[4] Olga Russakovsky, Yuanqing Lin, Kai Yu, Li Fei-Fei, Object-Centric Spatial Pooling for Image Classification, *ECCV*, 2012.

[5] Zaïd Harchaoui, Francis Bach, Image Classification with Segmentation Graph Kernels, *CVPR*, 2007.

[6] Olivier Duchenne, Armand Joulin, Jean Ponce, A Graph-Matching Kernel for Object Categorization, *ICCV*, 2011.

[7] Liang Lin, Xiaobai Liu, Shaowu Peng, Hongyang Chaoa, Yongtian Wang, Bo Jiang, Object Categorization with Sketch Representation and Generalized Samples, *PR*, 45(10): 3648–3660, 2012.

[8] Luming Zhang, Yahong Han, Yi Yang, Mingli Song, Shuicheng Yan, Qi Tian, Discovering Discriminative Graphlets for Aerial Image Categories Recognition, *IEEE T-IP*, 22(12): 5071–5084, 2013.

[9] Honglak Lee, Alexis Battle, Rajat Raina, Andrew Y. Ng, Efficient Sparse Coding Algorithms, *NIPS*, 2006.

[10] Ryan Farrell, Om Oza1, Ning Zhang, Vlad I. Morariu, Trevor Darrell, Larry S. Davis, Birdlets: Subordinate Categorization Using Volumetric Primitives and Pose-Normalized Appearance, *ICCV*, 2011.

[11] Ning Zhang, Ryan Farrell, Trever Darrell, Pose Pooling Kernels for Sub-category Recognition, *CVPR*, 2012.

[12] Bangpeng Yao, Gary Bradski, Li Fei-Fei, A Codebook-Free and Annotation-Free Approach for Fine-Grained Image Categorization, *CVPR*, 2012.

[13] Asma Rejeb Sfar, Nozha Boujemaa, Donald Geman, Vantage Feature Frames For Fine-Grained Categorization, *CVPR*, 2013.

[14] Thomas Berg, Peter N. Belhumeur, POOF: Part-Based One-vs.-One Features for Fine-Grained Categorization, Face Verification, and Attribute Estimation, *CVPR*, 2013.

[15] Jia Deng, Jonathan Krause, Li Fei-Fei, Fine-Grained Crowdsourcing for Fine-Grained Recognition, *CVPR*, 2013.

[16] Jiefeng Cheng, Jeffrey Xu Yu, Philip S. Yu, Graph Pattern Matching: A Join/Semijoin Approach, *IEEE T-KDE*, 2(7): 1006–1021, 2011.

[17] Kun Duan, Devi Parikh, David Crandall, Kristen Grauman, Discovering Localized Attributes for Fine-grained Recognition, *CVPR*, 2013.

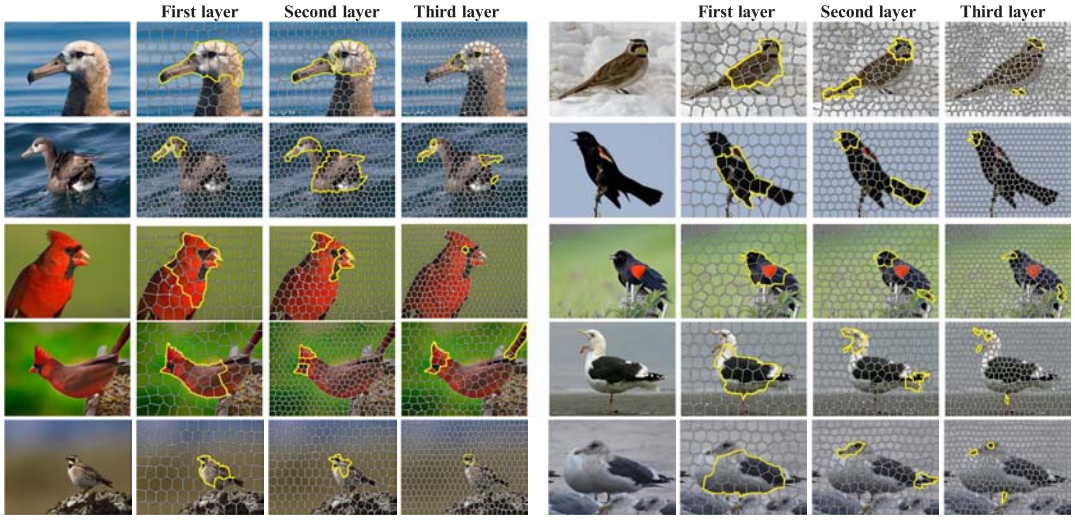

Figure 12: An example of the detected discriminative object parts from the CUB-200

[18] Anelia Angelova, Shenghuo Zhu, Efficient Object Detection and Segmentation for Fine-grained Recognition, *CVPR*, 2013.

[19] Dacheng Tao, Xuelong Li, Xindong Wu, Stephen J Maybank, Geometric mean for subspace selection, *IEEE T-PAMI*, 31(2): 260–274, 2009.

[20] Aditya Khosla, Nityananda Jayadevaprakash, Bangpeng Yao, Li Fei-Fei, Novel dataset for Fine-Grained Image Categorization, *CVPR Workshop*, 2011.

[21] Maria-Elena Nilsback, Andrew Zisserman, A Visual Vocabulary for Flower Classification, *CVPR*, 2006.

[22] Ferdinand Ôsterreicher, Igor Vajda, A New Class of Metric Divergences on Probability Spaces and its Statistical Applications, *Annals of the Institute of Statistical Mathematics*, 55(3): 639–653, 2003.

[23] Inderjit Dhillon, Yuqiang Guan, Brian Kulis, A Unified View of Kernel K-means, Spectral Clustering and Graph Cuts, *UTCS Technical Report*, 2005.

[24] N. Dalal, B.Triggs. Histograms of Oriented Gradients for Human Detection, *CVPR*, 2005.

[25] Markus Stricker, Markus Orengo. Similarity of Color Images, *Storage and Retrieval of Image and Video Databases*, 1995.

[26] Hairong Liu, Shuicheng Yan, Robust Graph Mode Seeking by Graph Shift, *ICML*, 2010.

[27] Svetlana Lazebnik, Cordelia Schmid, Jean Ponce, Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories, *CVPR*, 2006.

[28] Jianchao Yang, Kai Yu, Yihong Gong, Thomas Huang, Linear Spatial Pyramid Matching using Sparse Coding for Image Classification. *CVPR*, 2009.

[29] Jinjun Wang, Jianchao Yang, Kai Yu, Fengjun Lv, Thomas Huang, Yihong Gong, Locality-constrained Linear Coding for Image Classification, *CVPR*, 2010.

[30] Yang Yang, Yi Yang, Heng Tao Shen, Yanchun Zhang, Xiaoyong Du, Xiaofang Zhou, Discriminative Nonnegative Spectral Clustering with Out-of-Sample Extension, *IEEE T-KDE*, 25(8): 1760–1771, 2013.

[31] Luming Zhang, Yi Yang, Roger Zimmermann, Discriminative Cellets Discovery for Fine-Grained Image Categories Retrieval, *ICMR*, 2014.

[32] Peter Welinder, Steve Branson, Takeshi Mita, Catherine Wah, Florian Schroff, Caltech-UCSD Birds 200, *California Institute of Technology*, CNS-TR-2010-001, 2010.

[33] Josiah Wang, Katja Markert, Mark Everingham, Object-Centric Spatial Pooling for Image Classification, *BMVC*, 2009.

[34] Luming Zhang, Yue Gao, Yingjie Xia, Qionghai Dai, Xuelong Li, A Fine-Grained Image Categorization System by Cellet-Encoded Spatial Pyramid Modeling, *IEEE Transcations on Industrial Electronics* (T-IE), 62(1), pages: 564-571, 2015

[35] Luming Zhang, Yue Gao, Chaoqun Hong, Yinfu Feng, Jianke Zhu, Deng Cai, Feature Correlation Hypergraph: Exploiting High-order Potentials for Multimodal Recognition, *IEEE Transcations on Cybernetics* (T-CYB),), 44(8), pages: 1408-1419, 2013.

[36] Luming Zhang, Yue Gao, Rongrong Ji, Lv Ke, Jiale Shen, Representative Discovery of Structure Cues for Weakly-Supervised Image Segmentation, *IEEE Transcations on Multimedia* (T-MM), 16(2): 470–479, 2014.

[37] Luming Zhang, Mingli Song, Yi Yang, Qi Zhao, Zhao Chen, Nicu Sebe, , Weakly Supervised Photo Cropping, *IEEE Transcations on Multimedia* (T-MM), 16(1): 94–107, 2014.

[38] Luming Zhang, Mingli Song, Zicheng Liu, Xiao Liu, Jiajun Bu, Chun Chen, Probabilistic Graphlet Cut: Exploring Spatial Structure Cue for Weakly Supervised Image Segmentation, *IEEE Computer Vision and Pattern Recognition* (CVPR), pages: 1908–1915, 2013.

[39] Luming Zhang, Yue Gao, Rongrong Ji, Qionghai Dai, Xuelong Li, Actively Learning Human Gaze Shifting Paths for Photo Cropping, *IEEE Transcations on Image Processing* (T-IP), 23(5), pages: 2235–2245, 2014.

[40] Luming Zhang, Yue Gao, Roger Zimmermann, Qi Tian, Xuelong Li, Fusion of Multi-Channel Local and Global Structural Cues for Photo Aesthetics Evaluation, *IEEE Transcations on Image Processing* (T-IP), 23(3): 1419–1429, 2014.

[41] Luming Zhang, Yi Yang, Yue Gao, Changbo Wang, Yi Yu, Xuelong Li, A Probabilistic Associative Model for Segmenting Weakly- Supervised Images, *IEEE Transcations on Image Processing* (T-IP), 23(9), pages: 4150-4159, 2014.

[42] Luming Zhang, Mingli Song, Li Sun, Xiao Liu, Yinting Wang, Dacheng Tao, Jiajun Bu, Chun Chen, Spatial Graphlet Matching Kernel for Recognizing Aerial Image Categories, *ICPR*, 2012.

[43] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein, Introduction to Algorithms, *MIT Press and McGraw-Hill*, pages: 540-549,2001.

[44] Xiaopeng Zhang, Hongkai Xiong, Wengang Zhou, Qi Tian, Fused One-vs-All Mid-Level Features for Fine-Grained Visual Categorization, *ACM Multimedia*, 2014.

[45] Hanwang Zhang, Zheng-Jun Zha, Yang Yang, Shuicheng Yan, Yue Gao and Tat-Seng Chua, Attributes-augmented Semantic Hierarchy for Image Retrieval, *ACM Multimedia*, 2013.