Image segmentation for large-scale subcategory flower recognition

Anelia Angelova
NEC Labs America
anelia@nec-labs.com

Shenghuo Zhu
NEC Labs America
zsh@nec-labs.com

Yuanqing Lin
NEC Labs America
ylin@nec-labs.com

Abstract

We propose a segmentation algorithm for the purposes of large-scale flower species recognition. Our approach is based on identifying potential object regions at the time of detection. We then apply a Laplacian-based segmentation, which is guided by these initially detected regions. More specifically, we show that 1) recognizing parts of the potential object helps the segmentation and makes it more robust to variabilities in both the background and the object appearances, 2) segmenting the object of interest at test time is beneficial for the subsequent recognition.

Here we consider a large-scale dataset containing 578 flower species and 250,000 images. This dataset is developed by our team for the purposes of providing a flower recognition application for general use and is the largest in its scale and scope. We tested the proposed segmentation algorithm on the well-known 102 Oxford flowers benchmark [11] and on the new challenging large-scale 578 flower dataset, that we have collected. We observed about 4% improvements in the recognition performance on both datasets compared to the baseline. The algorithm also improves all other known results on the Oxford 102 flower benchmark dataset.

Furthermore, our method is both simpler and faster than other related approaches, e.g. [3, 14], and can be potentially applicable to other subcategory recognition datasets.

1. Introduction

This paper considers the automatic recognition of different species of flowers. Such a task is referred to as subcategory recognition, or fine-grained classification [6], in which the base-level category is 'flower' and the classes to be recognized are different types of flowers. In the subcategory recognition setting, the main challenge lies in the very fine differences between possibly similar objects that belong to different classes. Only very well trained experts are able to discriminate between all of the categories properly. Naturally, an automatic recognition system in such a setting will provide much value to non-experts.

To this end, our team has built a large-scale flower dataset which contains 578 different species of flowers and about 250,000 images (Figure 1). This dataset has been collected and developed with the goal of providing a large-scale flower recognition system. This is the largest collection (with the largest number of species and images) that is available for flower recognition.

One of the main goals for any such system is improving the recognition performance. As mentioned, the main challenge in subcategory classification are the fine differences between classes. Other challenges, specific to an automatic recognition system, are also present, for example, scale variations, intra-class variabilities, inter-class similarities, image blur, etc. (Figure 1). Furthermore, in the case of flowers, photographs are often taken in natural settings with rich and challenging backgrounds. Although the background can generally provide useful context, it can sometimes serve as distractor to a classification algorithm. For example, background features can become prominent and be extracted as possibly good discriminators, or some background features may be matched across different categories and thus make it harder to discriminate among them (Figure 2). This can cause deteriorated performance of the classification algorithm.

One possible solution to this problem is to identify the object region and segment out the object, so as to discount the background during classification. It will obviously be of huge benefit if the object can be automatically segmented before being recognized, because the recognition system can focus on the relevant regions of the image. When given an image, a person has no problem segmenting the object of interest, so it is almost understood that when an expert classifies the image, their attention will be focused on the most informative foreground area.

In this work we focus on automatically segmenting the possible object of interest *prior* to doing classification. We use the Laplacian propagation as our optimization technique, which allows for fast convergence and contributes to significant decrease of the overall run-time. This segmentation is 5-6 times faster than previously known seg-

978-1-4673-5054-9/13/$31.00 ©2013 IEEE

Figure 1. Example images from the large-scale 578 flowers dataset. A large variety of flower classes, as well as, intra-class variabilities, inter-class similarities, and changes in flower scales are available in this dataset. Bottom row left three images demonstrate inter-class similarity, the images come from three different classes: bellis perennis, matricaria chamomilla and leucanthemum vulgare. Bottom row right three images demonstrate intra-class variability, the images come from the same class, torenia fournieri, but differ in appearance and in scale.

mentation algorithms in similar scenarios [2, 13]. Furthermore, the method is simpler and is applicable to a variety of datasets, unlike previous work on flowers [11] or other categories [13] whose segmentation methods are very specialized to the super-level category at hand.

Our experiments show that the proposed segmentation method, in addition to being simpler and faster, is beneficial to the classification performance. We tested the algorithm on a well established flower recognition dataset, the Oxford flower dataset containing 102 species of flowers [11], and on our large-scale 578-class flower dataset. The proposed algorithm improved the baseline performance for both datasets by at least 4%.

1.1. Overview of the approach

Our approach is based on identifying regions, specific of the categories of interest at the time of detection. We then apply a Laplacian-based propagation and segmentation approach to segment the object (or objects) based on low level cues. The propagation process is guided by the initially detected regions, as they are already good indicators of the presence of the possible object. The initial regions are identified by a learning model.

The key intuitions of our approach are that when segmenting an object, recognizing parts of the image that possibly belong to the object can help delineate object boundaries which may not be otherwise very prominent. A segmented object, in turn, is beneficial for the final recognition, as shown in our experiments. This is the case because the algorithm 1) manages to remove background areas which may be confusing for the classification algorithm, 2) it provides boundaries and shape information for the object. The algorithm is very simple, much faster than the previously used approaches for segmentation [3, 14], and outperforms previous classification methods on benchmark datasets.

We subsequently process the segmented image (in addition to the original image) by the standard feature extraction and classification pipeline, so both the elimination of the background and the extracted shape information can potentially affect the extracted features in a positive way. While object segmentation has been used in many object recognition contexts, our segmentation algorithm is robust, adap-

Figure 2. The background may sometimes serve as a distractor in subcategory recognition, since it can provide strong features common to different categories.

tive, and efficient enough to be applied at testing time.

2. Related work

A large body of segmentation work exists [1, 2], with the majority of the focus being on offline segmentations applied to the training data. For example [1] explored approaches in which increasingly better segmentations are used to learn improved models for recognizing the objects in the database. Similarly, in the co-segmentation body of works [2, 9], better models are trained by exploiting shared appearance features in images containing the same class of objects. These approaches are mainly focusing on segmentation during training.

Recent works have proposed segmentation done at the time of classification. In [13] the authors propose to detect some specific part of the object of interest (e.g. a cat's head), and then segment the object by extrapolating from the textures and colors observed. These methods may sometimes suffer from the assumption they make that the object has consistent texture. Recent work [2] proposes to do segmentation prior to recognition, but they used the iterative GrabCut algorithm [14] whose running time ($\tilde{3}0$ sec. per image until convergence for segmentation only) limits its application to offline settings. Another interesting work on semantic segmentation [3] has similar runtime. Those methods, although proposing viable segmentations, are still slow for the classification to be done in practical applications.

3. Object segmentation

3.1. Detecting object-specific regions

We start our method with an initial search for regions possibly belonging to a flower in the image. For simplicity we use the super-pixel segmentation method by Felzenszwalb and Huttenlocher [7] to over-segment the image into small coherent regions. Each super-pixel region is described by the following set of feature descriptors: average color (R,G,B) of all the pixels within the region, global pooling of all HOG features [4] in the region, after encoding them by the LLC method [15] (see section 5.1 for more details on these features), shape mask of the region obtained by normalizing the region's area bounding box to 6x6 pixels, and size and boundary features as in [2].

Some of the feature descriptors are inspired by other segmentation methods which used super-pixel descriptors: e.g. the use of shape masks and bit-maps denoting adjacency to the boundary of the region are proposed in [2]. Unlike previous methods, we use the encoded HOG features here, because we believe they have better generalization capabilities and because in our classification method (Section 5) these features are already precomputed in the image and can be reused.

Using the feature representation described above, we build a model which can discriminate if a region belongs to a flower or to the background (Section 4). We apply this model to each region and extract the high confidence regions for both background and foreground. We then perform the optimization, described in Section 3.2, to segment the image into foreground area and background area.

3.2. Segmentation algorithm

Here we describe the optimization done using the Laplacian operator for the purposes of segmentation. Let I_j denote the j-th pixel in an image and f_j denotes its feature representation. The goal of the segmentation task is to find the label X_j for each pixel I_j, where $X_j = 1$ when the pixel belongs to the object and $X_j = 0$, otherwise. For the optimization, we relax the requirement on X_j and allow them to be real-valued. We form the affinity matrix W, using the feature representations f_i of each pixel:

$$W_{ij} = exp\left(-\frac{|f_i - f_j|^2}{2\sigma^2}\right) \quad (1)$$

The terms W_{ij} are nonzero for only neighbouring pixels, e.g. in our case we use the 8-connected component neighborhood for each pixel. The goal is to minimize the cost function $C(X)$ with respect to all pixel labels X:

$$C(X) = \frac{1}{2}X^T(I - S)X + \frac{\lambda}{2}|X - Y|^2 \quad (2)$$

where $S = D^{-1/2}WD^{-1/2}$, $D_{ii} = \sum_{j=1}^{N} W_{ij}$, and Y_i are the desired labels for some (or all) the pixels. Those label constraints impose prior knowledge of what is an object and background (Section 3.1 described our approach of how we assign them). This is a standard Laplacian label propagation formulation [16].

After differentiation of Equation 2, we obtain the optimal X, which is the solution of the system of linear equations:

$$((1 + \lambda)I - S)X = \lambda Y$$

$$X = \lambda((1 + \lambda)I - S)^{-1}Y.$$

In our implementation we use the Conjugate Gradient method and achieve very fast convergence. Figure 3 shows example segmented images.

3.3. Implementation details

In this section we include the details of our Laplacian segmentation implementation.

To perform the Laplacian segmentation, a feature representation f_i per each pixel is needed. Obviously, the goal is for similar pixels (or pixel neighbourhoods) to have very close feature representations, but at the same time the time of computation of these features has to be very fast. Here we set f_i to be the (R,G,B) color values of the pixel, but other choices are possible too.

To make the optimization feasible, we resized the original image to approximately 120 by 120 pixels per image area (which is typically a 4 to 5 times rescaling, preserving the original aspect ratio). This is needed in order to have a tractable optimization procedure. We did not observe significant improvements in performance when using the full-scale segmentations. The parameter λ is selected as in [16].

We also note here that the initial super-pixel segmentation of Felzenszwalb and Huttenlocher [7] is not sufficient to do the segmentation for our purposes well. This is because some of the super-pixel regions may not be very informative or their boundaries may not be as smooth as desired for our purposes.

Another thing that is notably different in this work from a standard Laplacian propagation implementation is that, instead of using isolated foreground and background pixels, we use all the pixels in a region, that is detected as confident, and set their initial values to the confidence value of the region. The reason is that the regions may vary in size and texture and may have very different diffusion properties, so in our case we observed better convergence and subsequently better segmentations. For the same reasons we considered separate segmentations with respect to the foreground and the background. This resulted in having segmentations that are more stable and adaptive to variabilities in both foreground and background appearances.

4. Training the region model

This section describes how to train the model which discriminates between a region belonging to the super-class (i.e. any flower) and to the background.

Each training image is decomposed into super-pixels using the method proposed by Felzenszwalb and Huttenlocher [7]. Each super-pixel region is represented by the set of feature descriptors, already described in Section 3.1.

Given ground truth segmentation, we consider regions with a specific overlap to the background or the foreground. Regions which are in-between are ignored. We then trained a standard linear SVM algorithm to learn the decision boundary. When no ground truth is available, we use approximate segmentation given by an automatic algorithm and then iteratively improve our segmentation by applying the trained model. For example, for the case of Oxford 102 flowers datasets we used the segmentation images provided here [11] and iteratively improved the segmentation. For the large-scale 578-flower dataset, described in this paper, we used the same model that has been trained on Oxford 102 flowers dataset.

The training of the model is done offline. A potential advantage of this model is that it is general, i.e. not specialized to characteristics of one super-class, and can be applicable to different types of species, whereas previous subcategory classification approaches are more specific [6, 11].

5. Subcategory recognition with segmentation

As mentioned, the input image will be segmented at recognition time. This section describes how we use the segmented image in the final flower recognition task. For simplicity, we first describe the baseline algorithm.

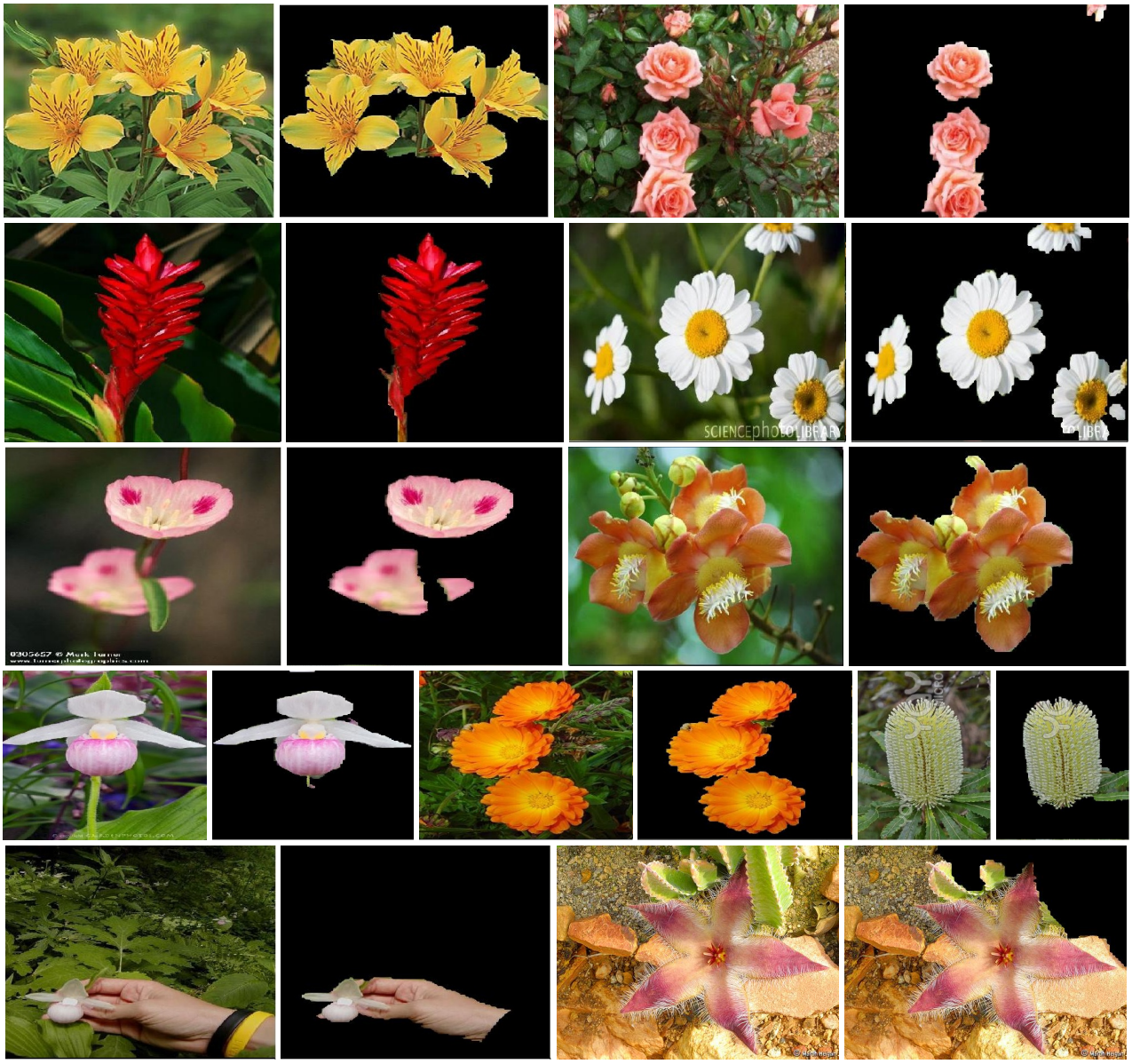

Figure 3. Example segmented images from the 578 flowers dataset. Although not necessarily perfect, these segmentations are sufficient to remove most of the background. Examples of failed segmentations are shown in the bottom row.

5.1. Baseline algorithm

We apply a feature extraction and classification pipeline which is very similar to the one of Lin et al. [10] to the input image. In our feature extraction pipeline we first extract HOG [4] features at 4 different levels, then those features are encoded in 8K dimensional global feature dictionary using the LLC method [15]. After that, a global max pooling of the encoded features in the image is done, as well as, max poolings in a 3 by 3 grid of the image. Our classification pipeline uses the 1-vs-all strategy of linear SVM classification and we used the Liblinear SVM implementation [5]. For the very large 578-flowers dataset, we used a Stochastic Gradient Descent algorithm, e.g. [10].

5.2. Classification with segmentation

The segmented image is processed through the same feature extraction pipeline as the original image. We then combine the two sets of extracted features (from the original image and from the segmented image). One thing to note here is that, because of our decision to apply HOG type features and pooling to the segmented image, the segmentation helps with both providing shape of the contour of the flower to be

recognized, as well as, ignoring features in the background that can be distractors. On the other hand, by keeping both sets of features from the original and the segmented image, we can avoid losing precision due to mis-segmentation.

In our experiments we found that it is sufficient to keep a global pooling of the segmented image and this has shown to be very useful for improving performance without increasing the dimensionality too much.

In terms of computation, our algorithm performs much better compared to competitors [2, 13]. Still, some improvements are needed to be real-time. The segmentation algorithm itself takes about 4-5 seconds seconds (the features for classification also take some time but they are the same from the baseline method and could be reused). Our baseline algorithm runs within 1-2 seconds. In total, the segmentation procedure adds an overhead of 5-6 seconds, which is still relatively slow for real-time performance, but is 5 to 6 times faster than state-of-the-art segmentation algorithms, e.g. Grabcut [14], or [3], both of which take at least 30 seconds.

6. Experiments

In this section we show experimental results of our proposed algorithm on the flower datasets: the Oxford 102 flowers [11] and our large-scale 578 class dataset.

6.1. Oxford 102 flower species dataset

Oxford 102 flowers dataset is a well established dataset for subcategory recognition proposed by Nilsback and Zisserman [11]. The dataset contains 102 species of flowers and a total of 8189 images, each category containing between 40 and 200 images. It has established protocols for training and testing, which we have adopted in this paper too.

A lot of methods have been tested on this dataset [2, 8, 11, 12], including some segmentation-based [2, 11]. Some of the segmentation methods are designed to be very specific to the appearance of flowers [11] (with the assumption that a single flower is in the center of the image and takes most of the image), while others [2] are more general and can also be applied to other types of datasets. Our method is closer to the latter, since we are proposing a general method that does not make assumptions about the set of categories for classification or the initial location or size of the objects in the image. In Section 6.2 we test the algorithm on a much larger and more diverse dataset, in which the flowers are not necessarily in the center of the image, can contain multiple small cluster flowers, can vary in scale, and have a lot more within-class variability (Figure 1).

The performance of our approach on this dataset (see Table 1) is 80.66% which outperforms all previous known methods in the literature (some by as much as 4 to 8%) [2, 8, 11, 12]. One important thing to note is that the improvement of our algorithm over our baseline is about 4%, and the only difference between the two is the addition of the proposed segmentation algorithm and the features extracted from the segmented image.

Method	Accuracy (in %)
Our baseline (no segmentation)	76.7
Nilsback and Zisserman [11]	72.8
Ito and Cubota [8]	74.8
Nilsback and Zisserman [12]	76.3
Chai et al., Bicos method [2]	79.4
Chai et al., BicosMT method [2]	80.0
Ours	**80.66**
Ours: improvement over our baseline	+3.94

Table 1. Classification performance on Oxford 102 flower dataset.

Method	Accuracy (in %)
Our baseline (no segmentation)	52.35
Ours	**56.76**
Ours, improvement over our baseline	+4.41

Table 2. Classification performance on the 578 flowers dataset for the top returned result.

6.2. Large-scale 578 flower dataset

This dataset consists of 578 species of flowers and contains about 250,000 images. The categories have been identified by a team of expert botanists to approximately cover 90 percent of the most common flower species in the world and the data has been painstakingly checked by the same team of experts to make sure the class labels are correct. The goal of developing this data is to build a recognition application which can recognize and/or provide top K suggestions (e.g., for K=5, 10, etc.) for an input flower image, and be available for general use. This is the largest dataset of its kind. Comparing to previous known flower datasets, the largest one has been the Oxford 102 dataset [11], which contains 102 flower species. We believe that such a flower-recognition application will be of great value to people who are not botanist or flower experts.

We tested our baseline algorithm vs the proposed segmentation-based algorithm on this data, see Table 2. The improvement provided by our segmentation method is 4.41 percent for the top 1 returned result. Here we used the Stochastic Gradient Descent algorithm for learning, for both the baseline and the segmentation-based algorithm, instead of the Liblinear SVM implementation [5], because the dataset is too large and Liblinear fails to load it into memory.

Note that this large-scale data has no segmentation ground truth or bounding box information, since it contains

250,000 images and obtaining those would be prohibitive or at least very expensive. Thus, here the advantage that an automatic segmentation algorithm can give in terms of improving the final classification performance is really important. Another interesting fact is that here we have used the same initial region detection model that was trained on the Oxford 102 flowers dataset, which contains fewer species of flowers (102 instead of 578). This was motivated again by the lack of good ground truth for such a large volume of data. We believe that the performance of the segmentation algorithm can be further improved after adapting the segmentation model for this data, in particular.

7. Summary and future work

We propose a novel segmentation algorithm which is robust and adaptive to variety of object appearances and backgrounds. Our algorithm uses learning to guide the segmentation process and is based on the intuition that recognizing (even imperfectly) some regions of the object can help delineate its boundaries and thus segment the potential object of interest.

We show that the proposed segmentation of objects is very useful for recognition by improving the classification performance on the Oxford 102 flowers dataset [11] and on a large-scale 578 flowers dataset. The improvements in performance are about 4% for both datasets and are due to the automatic segmentation done at test time. This is important since the large-scale datasets contain hundreds of thousands of images and no manual segmentation for them is practical. The algorithm also improves all other known benchmark results on the Oxford 102 flower dataset.

Our algorithm is simpler and faster than previously used segmentation algorithms in similar scenarios, e.g. [3, 14]. It is also more general and not specific to the appearance of flowers, so it can potentially be applied to other types of categories in natural images.

Although the speed is at least 5 times better than previously known segmentation algorithms, one major focus of our work is on improving the computational time further. Other improvements can be done to the feature model, e.g. it can be represented as a mixture of submodels, each one responsible for a subset of flowers that are very similar to each other but different as a group from the rest. This, we hope, can improve the precision of the model, and subsequently the segmentation, as well. Additionally, the feature representations used can be further enhanced with more powerful and discriminative features.

Acknowledgements

We would like to thank the team lead by Prof. Chelsea Specht at the Department of Plant and Microbial Biology at UC Berkeley, who selected the flower species to be recognized and provided labeling information. This work would not have been possible without their critical expertise.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Classcut for unsupervised class segmentation. *ECCV*, 2010.

[2] Y. Chai, V. Lempitsky, and A. Zisserman. Bicos: A bi-level co-segmentation method for image classification. *ICCV*, 2011.

[3] G. Csurka and F. Perronnin. An efficient approach to semantic segmentation. *IJCV*, 2011.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005.

[5] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 2008.

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

[7] P. Felzenszwalb and D. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004.

[8] S. Ito and S. Kubota. Object classfication using heterogeneous co-occurrence features. *ECCV*, 2010.

[9] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. *CVPR*, 2010.

[10] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang. Large-scale image classification: fast feature extraction and svm training. *CVPR*, 2011.

[11] M.-E. Nilsback and A. Zisserman. A. automated flower classification over a large number of classes. *ICVGIP*, 2008.

[12] M.-E. Nilsback and A. Zisserman. An automatic visual flora - segmentation and classification of flower images. *DPhil Thesis, University of Oxford, UK*, 2009.

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

[14] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graphics*, 2004.

[15] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. *CVPR*, 2010.

[16] D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. *NIPS*, 2004.