Fine-grained Flower Classification

Leo Lou
qibinlou@stanford.edu

Abstract

In this paper, we tackle the task of fine-grained flower classification. Based on the earlier work of Maria et al.[6], we incorporate both traditional features like HOG, SIFT, HSV with CNN feature to better describe inter-class differences between flowers species. We extensively experiment on different ways to utilize these features to generate a high performance model, such as feature normalizing, feature scaling, feature fusing and multiple kernel learning.

Here we use the well-known 102 Oxford flowers benchmark dataset to do our experiment. Our approach has beaten the ground truth work by Maria [6] and the another CNN based work by [7], though the latter one contains another approach which yield better result than ours(but takes much longer time to do experiments).

1. Introduction

Nowadays image based classification systems are achieving better and better performance as a result of large datasets and neural networks. In this paper, instead of focusing on the task of classifying as many as different objects, we investigate the problem of recognizing a large number of classes within one category, in our case, flowers. Such task is called fine-grained classification.

There has been progress in expanding the set of fine-grained domains we have data for, which now includes e.g. birds, aircraft, cars, flowers, leaves, and dogs. In this paper, we focus on the flowers fine-grained classification task. For human beings, we can use different features of flowers to distinguish different species;for example, we can use color, shape, size, and smell information to help us make a better decision. But for computers, the only information they can get is from the input image, which requires us to well design visual features to describe the flowers.

In this paper, We first discuss several key components of our classification pipeline, including flowers segmentation in section 3, flowers representing features in section 4, and experiment details in section 5.

2. Related Work

There are several noticeable work done on fine-grained flower classification.

In [6], Maria 1 investigated on using multiple kernel learning for flower images acquired under fairly uncontrolled image situations the images are mainly downloaded from the web and vary considerably in scale, resolution, lighting, clutter, quality, etc. They combined different image features(HSV+SIFT+HOG) to describe a flower image and to their model. Their work is known as 102 Oxford flowers benchmark. Anelis *et al*. [3] designed a better segmentation algorithm to identify potential flower body regions and applied feature extraction on that. They also developed a much larger dataset than Oxford 102 flower dataset with 578 flower species and 250,000 images, which contributed to 4% classification performance improvement compared to Oxford 102 flower benchmark dataset. There are other ways to describe flower images. For example, [8] used generic features extracted from a convolutional neural network previously used to perform general object classification. They experimented the CNN features on plant classification task together with an extremely randomized forest.

3. Segmentation

Most flowers are recognizable only by their body. Performing a segmentation to separate the flower body from the background can help reduce the noise in feature extraction process.

Several papers have proposed methods explicitly for the automatic segmentation of a flower image into flower as foreground, and the rest as background. Here We use the segmentation scheme proposed by Nilsback and Zisserman [5]. Figure 1 shows several example segmentaions from this method. The problem with this schema is that there are some over-segmented images, where there is no foreground at all. See Figure 2.

4. Classification

There are several key features that we human use to differentiate various kinds of flowers. First of all, color is useful when discriminating a sunflower from a rose. But to differentiate a buttercup from a dandelion, shape would be much more useful, but color would not. Smell is also useful in some cases, though totally useless in our flowers images classification case. In this section, we discuss several features to represent a flower image, and then discuss pros and cons of multi-kernels SVM classifier and neural network classifier.

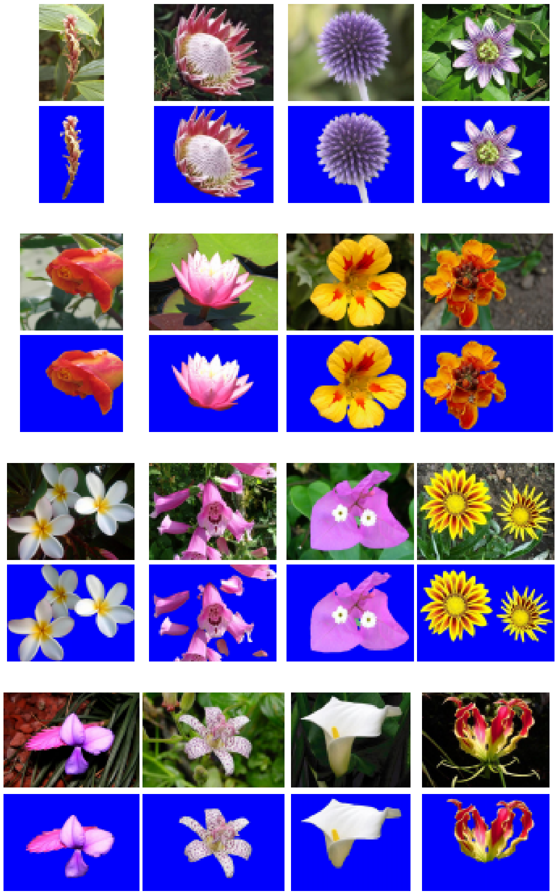

Figure 1. Example segmentations from [5]

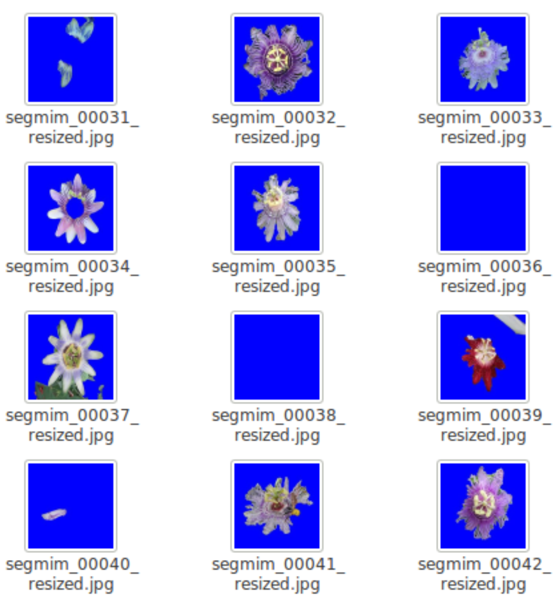

Figure 2. Bad segmentation examples

4.1. Features

Color: Colour is described by taken the HSV values of the pixels. The HSV space is chosen because it is less sensitive to variations in illumination and should be able to cope better with pictures of flowers taken in different weather conditions and at different time of the day. The HSV values for each pixel in an image are clustered using k-means. Given a set of cluster centres (visual words) $w_i^c, i = 1, 2, ..., V_c$, each pixel in the image I is then assigned to the nearest cluster centre, and the frequency of assignments recorded in a a V_c dimensional normalized frequency histogram $n(w^c|I)$.

SIFT: SIFT descriptors are computed at points on a regular grid with spacing M pixels over the foreground flower region. At each grid point the descriptors are computed over circular support patches with radii R pixels. Only the grey value is used (not color), and the resulting SIFT descriptor is a 128 vector. To cope with empty patches, all SIFT descriptors with L2 norm below a threshold (200) are zeroed. Note, we use rotationally invariant features. The SIFT features describe both the texture and the local shape of the flower (e.g. fine petal structures (such as a sunflower) vs spikes (such as a globe thistle). We obtain $n(wf|I)$ through vector quantization in the same way as for the color features.

Histogram of Gradients: HOG features, are similar to SIFT features, except that they use an overlapping local contrast normalization between cells in a grid. However, instead of being applied to local regions (of radius R in the case of SIFT), the HOG is applied here over the entire flower region (and it is not made rotation invariant). In this manner it captures the more global spatial distribution of the flower, such as the overall arrangement of petals. The segmentation is used to guide the computation of the HOG features. We find the smallest bounding box enclosing the foreground segmentation and compute the HOG feature for the region inside the bounding box. We then obtain $n(wh|I)$

CNN Feature: Convolutional neural networks (CNNs) were proposed in 1989 by LeCun et al. [4]. Recently, it has shown its power with the availability of large datasets, performance improvement of GPUs and efficient algorithms. It has been shown that combining CNN features with a simple classifier such as SVM can outperform classical approaches for various classification and detection tasks, which also require hand-crafted features.

4.2. Classifier

Currently we adopt multiple kernels SVM as our classifier to train our final model. It is reasonable to have a weighted linear combination of SVM kernels, each corresponding to each feature. The weights vary for different flower species, some of which might have a high weight on Color feature while some might have a high weight on SIFT feature. For individual feature, we simply use one-vs-other SVM classifier to get their corresponding model.

5. Experiment

5.1. Dataset

Oxford 102 flowers dataset is a well established dataset for subcategory recognition proposed by Nilsback *et al*. [6]. 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 our work too.

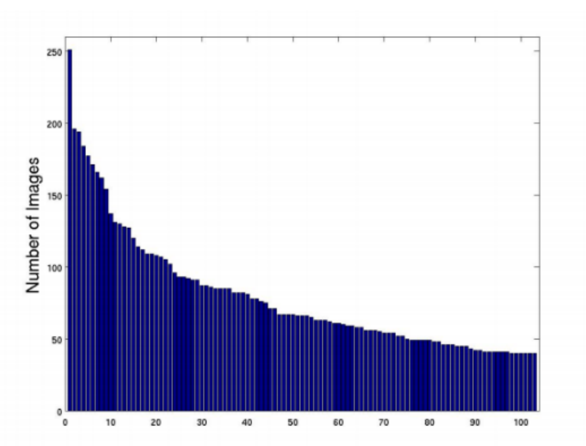

Figure 3. The distribution of numbers of images over 102 classes

5.2. Setup

Our experiment is based on Python environment and various Python packages, numpy, scipy, scikit-learn, etc. The complete package list can be obtained from the import sections of our scripts. It's worth mentioning that we use OpenCV 2.4.8 [1] for Python package to process our images and extract different features. Due to some unknown issues, our OpenCV version doesn't support to extract SIFT features. Instead, we use the open sourced Octave and VLFeat [9] to extract SIFT features. It's a little bit troublesome to run VLFeat library on Octave because it's still in experiments. The Overfeat network feature extractor can be installed from Github source [2] or the pre-build version.

5.3. Procedure

In our experiments, we combine the training set and the validation set in the dataset since we skip fine tuning the parameters for different features experiments. Instead, we directly reuse the reported optimal parameters from [6]. Therefore, the optimum numbers of words is 1000 for HSV, 3000 for SIFT, 1500 for HOG. These optimum numbers might not really be the optimal ones for our experiments and other reproducing experiments because of libraries and implementation differences. We train our classifiers on the combined training data(both the training and validation sets) and test our classifiers on the testing sets.

For features like HSV, HOG and SIFT, we use KMeans to cluster the features to get our visual words. We use MiniBatchKMeans in scikit-learn toolbox to overcome the bottleneck that some feature sets are too large to fit into the entire memory. MiniBatchKMeans can cluster the centers batch by batch on our split feature sets.

5.4. Result

We report our experiments result in Table 1. It can be seen that combining all the features contributes to a far better performance than using single feature. CNN feature outperforms all the other features. Though we observe that CNN features extracted from segmented flowers images are less descriptive than from images with background. There are probably two reasons for this; one is that our segmentation algorithm is still not perfect that it filters some important parts of the flowers images when segmenting; another is that background information actually helps to better classify a flower image.

Features	mAP
HSV	42.3%
HOG	49.1%
SIFT	53.0%
CNN w/o segmentation	73.9%
CNN w/ segmentation	54.1%
HSV+HOG+SIFT+CNN	84.0%

Table 1. Classification performance on the test set. mAP refers to classification performance averaged over all classes(not over all images)

Table 2 shows the comparison result between our work and others. Note that in [7], the best performance is achieved by further augment the training set by adding

Method	mAP
Nilsback *et al*. [6]	76.3%
Anelia *et al*. [3]	80.66%
Ours	84.0%
Ali *et al*. [7]	86.8%

Table 2. Comparison between our work and others

cropped and rotated samples and doing component wise power transform, which makes the training model invariant to scale and rotation. We don't implement this step because of priorities and time, but we believe it will also benefit our model and increase our final performance.

6. Summery and future work

So far we have researched and experimented on available image features and segmentation algorithms for fine-grained flower classification task. We observe that multiple features empower the classifier to train a better model and achieve a better classification accurate on test sets. For the fine-grained flowers classification task, the learning of different weights for different classes enables us to use an optimum feature combination for each classification.

We also realize there are flaws in current pipeline, e.g. computation time cost is not good enough to be in the realtime scale. It poses a barrier to build an usable flowers classification service for users. Future work should include using current method to train on a larger dataset and fine tuning the neural network which we use to extract CNN features on flowers dataset.

References

[1] *OpenCV*. https://github.com/itseez/opencv.

[2] *Overfeat*. https://github.com/sermanet/OverFeat.

[3] A. Angelova, S. Zhu, and Y. Lin. Image segmentation for large-scale subcategory flower recognition. In *Applications of Computer Vision (WACV), 2013 IEEE Workshop on*, pages 39–45. IEEE, 2013.

[4] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[5] M.-E. Nilsback and A. Zisserman. Delving into the whorl of flower segmentation. In *BMVC*, pages 1–10, 2007.

[6] M.-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Computer Vision, Graphics & Image Processing, 2008. ICVGIP'08. Sixth Indian Conference on*, pages 722–729. IEEE, 2008.

[7] A. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson. Cnn features off-the-shelf: an astounding baseline for recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 806–813, 2014.

[8] N. Sünderhauf, C. McCool, B. Upcroft, and T. Perez. Fine-grained plant classification using convolutional neural networks for feature extraction. In *CLEF (Working Notes)*, pages 756–762, 2014.

[9] A. Vedaldi and B. Fulkerson. *VLFeat: An Open and Portable Library of Computer Vision Algorithms*, 2008.