Automated flower classification over a large number of classes

Maria-Elena Nilsback and Andrew Zisserman
Visual Geometry Group, Department of Engineering Science
University of Oxford, United Kingdom
men,az@robots.ox.ac.uk

Abstract

We investigate to what extent combinations of features can improve classification performance on a large dataset of similar classes. To this end we introduce a 103 class flower dataset. We compute four different features for the flowers, each describing different aspects, namely the local shape/texture, the shape of the boundary, the overall spatial distribution of petals, and the colour. We combine the features using a multiple kernel framework with a SVM classifier. The weights for each class are learnt using the method of Varma and Ray [16], which has achieved state of the art performance on other large dataset, such as Caltech 101/256. Our dataset has a similar challenge in the number of classes, but with the added difficulty of large between class similarity and small within class similarity. Results show that learning the optimum kernel combination of multiple features vastly improves the performance, from 55.1% *for the best single feature to* 72.8% *for the combination of all features.*

1 Introduction

As image based classification systems are improving the task of classifying objects is moving onto datasets with far more categories, such as Caltech256 [8]. Recent work [2, 8, 9, 16, 17, 18] has seen much success in this area. In this paper, instead of recognizing a large number of disparate categories, we investigate the problem of recognizing a large number of classes *within* one category – that of flowers. Classifying flowers poses an extra challenge over categories such as bikes, cars and cats, because of the large similarity between classes. In addition, flowers are non-rigid objects that can deform in many ways, and consequently there is also a large variation within classes. Previous work on flower classification has dealt with a small number of classes [12, 14] ranging from 10 to 30. Here we introduce a 103 class dataset for flower classification. Images from the dataset are shown in figure 1.

What distinguishes one flower from another can sometimes be the colour, e.g. blue-bell vs sunflower, sometimes the shape, e.g. daffodil vs dandelion, and sometimes patterns on the petals, e.g. pansies vs tigerlilies etc. The difficulty lies in finding suitable features to represent colour, shape, patterns etc, and also for the classifier having the capacity to learn which feature or features to use.

In the case of the Caltech 101/256 image datasets [7, 8], state of the art performance has been achieved by using multiple features [18] and a linear combination of kernels in a SVM classifier [2, 3, 16]: a base kernel is computed for each feature (e.g. shape, appearance) and the final kernel is composed of a weighted linear combination of these base kernels [3] with a different set of weights learnt for each class. Varma and Ray [16] showed that the weights could be learnt by solving a convex optimization problem.

In this paper we investigate this multiple kernel learning approach 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. The link we make is to automatically segment each image (section 2) so that the flower is isolated in the image. This makes the recognition challenge somewhat similar in nature to that of Caltech 101/256 – in that there is only a single (or very few) instances of the object in each image (excepting fields of flowers like bluebells) – i.e. the background clutter has been removed, and there are a similar number (103 vs 101) of classes to be classified. On the other hand flowers have the additional challenges (compared to Caltech101) of scale variation, pose variation and also greater between class similarity.

We design features, and corresponding kernels, suited to the flower class which capture the colour, texture, and shape (local and global) of the petals and their arrangement. This is presented in section 3. The image dataset and experimental procedure are described in section 4, and results on the test set given in section 5.

978-0-7695-3476-3/08 $25.00 © 2008 IEEE
DOI 10.1109/ICVGIP.2008.47

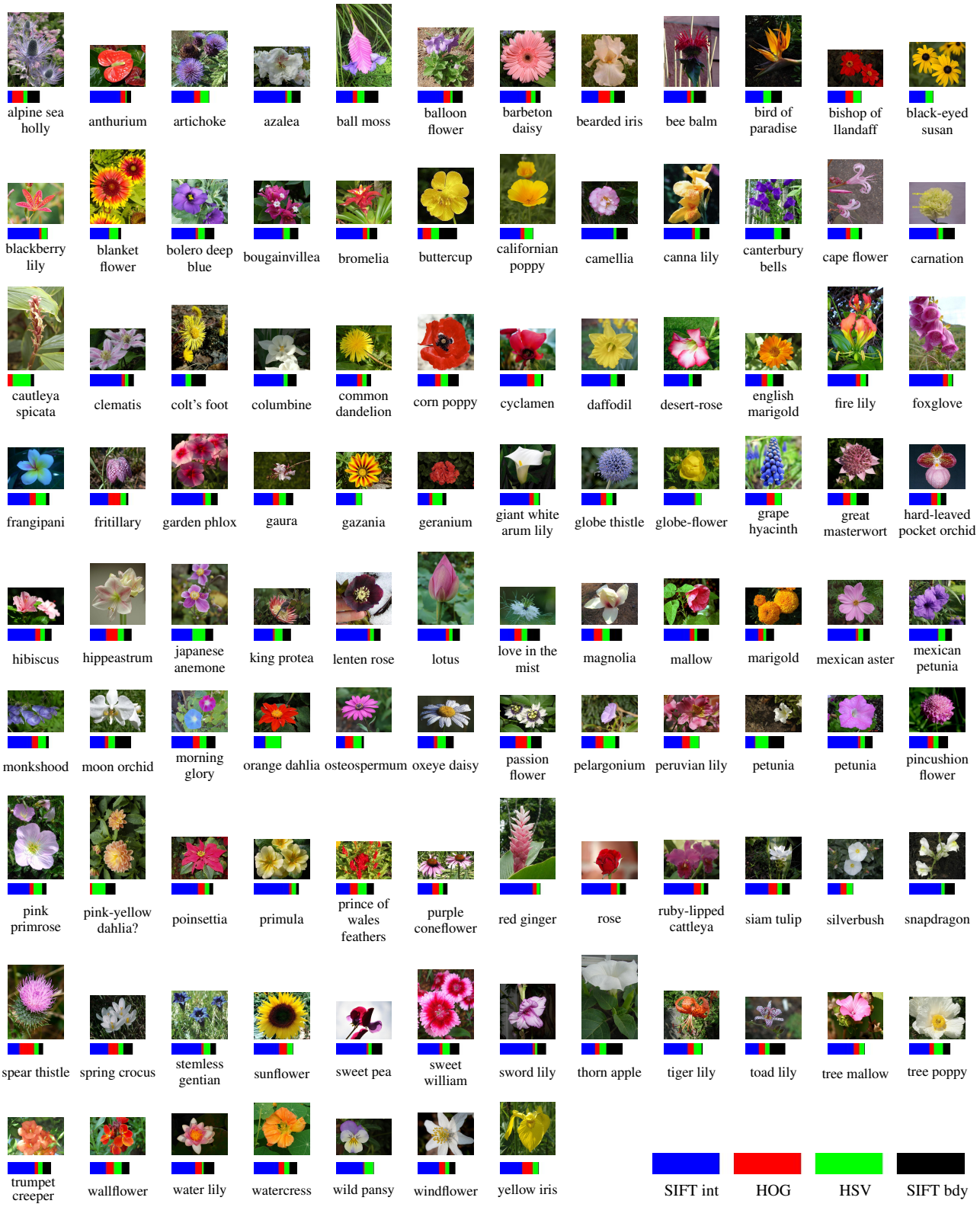

Figure 1: The 103 class flower dataset. Each image is an instance of a different class. They are sorted alphabetically. The colour bar below each image shows the magnitude of the features weights learnt for each class, as described in section 4.

2 Segmentation

Several papers [6, 14, 12, 13] have proposed methods explicitly for the automatic segmentation of a flower image into flower as foreground, and the rest as background. We use here the segmentation scheme proposed by Nilsback and Zisserman [13].

The scheme of [13] proceeds in an iterative manner: first an initial flower segmentation is obtained using *general* (non-class specific) foreground and background colour distributions. These distributions are learnt by labelling pixels in a few training images of each class in the dataset as foreground (i.e. part of the flower), or background (i.e. part of the greenery), and then averaging the distributions across all classes. Given these general foreground and background distributions, the initial binary segmentation is obtained using the contrast dependent prior MRF cost function of [4], optimized with graph cuts. This segmentation may not be perfect, but is often sufficient to extract at least part of the external boundary of the flower. A generic flower shape model is then fitted to this initial segmentation in order to detect petals. The model selects petals which have a loose geometric consistency using an affine invariant Hough like procedure. The image regions for the petals deemed to be geometrically consistent are used to obtain a new *image specific* foreground colour model. The foreground colour model is then updated by blending the image specific foreground model with the general foreground model. The MRF segmentation is repeated using this new colour model. In cases where the initial segmentation was not perfect, the use of the image specific foreground often harvests more of the flower. The steps of shape model fitting and image specific foreground learning can then be iterated until convergence, when no or very little change has occurred between two consecutive iterations.

The scheme was introduced using a 13 class flower dataset, a subset of the 17 class flower dataset of [12]. Figure 2 shows example segmentations obtained using this scheme on our 103 class dataset. It can be seen that it also works well for flowers very different in shape to those used in [13].

Figure 2: Example segmentations.

3 Classification

The aim is to obtain a classifier which is discriminative enough between classes, but also is able to classify correctly all instances of the same class. It needs to to be able to represent and learn that to discriminate a sunflower from a daisy, colour is a useful cue but shape would be quite poor. Conversely, to differentiate a buttercup from a dandelion, shape would be much more useful, but colour would not. In this section we first describe four features designed to represent the foreground flower regions, and then the linear combination of kernels used for the one-vs-rest SVM classification, where each kernel corresponds to one feature.

3.1 Features

Different features are chosen to describe different properties of the flower. The low level features we use are colour, histogram of gradient orientations (HOG) [5], and SIFT [10] sampled on both the foreground region and its boundary.

Colour: 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 on the foreground region: SIFT [10] 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 colour), 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 $n(w^f|I)$ through vector quantization in the same way as for the colour features.

SIFT on the foreground boundary: The boundary of the segmentation gives us the boundary of the flower. The difficulty of describing the shape is increased by the natural deformations of a flower. The petals are often very soft and flexible and can bend, curl, twist etc. By sampling SIFT features on the boundary of the flower we can can give greater emphasis (over the internal features) to the local shape of the boundary. A similar boundary feature was used in [11]. The 128 dimensional SIFT descriptors with radii R pixels are computed at each step S along the boundary. In a similar manner to the SIFT features for the internal region, only the grey value is used. $n(w^b|I)$ is obtained by clustering only the boundary SIFTs, i.e. separate vocabularies are used for the boundary and internal SIFT features.

Histogram of Gradients: HOG features [5], 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 $n(w^h|I)$ through vector quantization in the same way as for the previous features.

3.2 Linear combination of kernels classifier

The classifier is a SVM [15] using multiple kernels [1]. A weighted linear combination of kernels is used, one kernel corresponding to each feature. The final kernel has the following form for two data points i and j:

$$K(i,j) = \sum_{f \epsilon F} \beta_f \exp\left(-\mu_f \chi_f^2(x_f(i), x_f(j))\right) \quad (1)$$

where x_f is the feature vector for descriptor f (e.g. the normalized bag-of-visual-words histogram in the case of local shape), and β_f is the weight for feature f. $\chi^2(x,y)$ is the symmetric Chi-squared distance between histograms x and y. Note, $K(x,y) = \exp(-\mu\chi^2(x,y))$ is a Mercer kernel, and consequently (1) is a Mercer kernel by the sum rule. The parameter μ_f is set to 1/mean value of the χ^2 distances between the vectors x_f over all the training images [18].

Following [3], a set of parameters β_f is learnt for each class as a one-vs-rest classifier, as described in the following section. The final classification of a test image is then determined by the classifier with the most positive response over all the flower classes.

4 The dataset and experimental procedure

In this paper we introduce a dataset consisting of 8189 images divided into 103 flower classes. Figure 1 shows one example of each class. These are chosen to be flowers commonly occurring in the United Kingdom. Most of the images were collected from the web. A small number of images were acquired by taking the pictures ourselves. Each class consists of between 40 and 250 images. Figure 3 shows the distribution of the number of images over all the classes. Passion flower has the greatest number of images and eustoma, mexican aster, celosia, moon orchid, canterbury bells and primrose have the least, i.e. 40 per class. The images are rescaled so that the smallest dimension is 500 pixels.

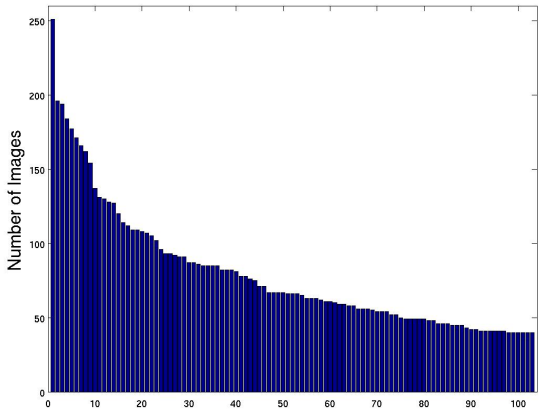

Figure 3: The distribution of the number of images over the 103 classes.

The dataset is divided into a training set, a validation set and a test set. The training set and validation set each consist of 10 images per class (totalling 1030 images each). The test set consist of the remaining 6129 images (minimum 20 per class). The validation set is used to optimize the number of visual words for each of the features, and the radius and spacing of the SIFT features.

For both the validation and test sets, performance is measured per class (in the same manner as for Caltech101/256), i.e. the final performance is the classification averaged over all classes (not over all images).

4.1 Optimization on the validation set

The optimum value for parameters such as the grid spacing (for the internal SIFT features) or the k in the k-means clustering for each feature type are learnt by optimizing the performance on the validation set in the standard manner.

For example for the internal SIFT features, the classifier is trained on the training data using only the kernel for this one feature. The optimum number of words in the vocabulary is determined by searching over a range between 1000 and 10000 and finding the maximum classification performance on the validation set. Each k-means clustering is repeated 3 times, and the best results kept. For colour the search is over the range 100 to 5000, because colour features have much fewer dimensions than SIFT, so we expect to be able to use fewer words to describe them. Both the grid spacing M and the circular support patches R, are searched over a range of 5 to 50 pixels. Maximization of the performance on the validation set is carried out separately for each variable.

The optimum number of words is 1000 for the colour features, 8000 for the SIFT over the entire foreground region, 3000 for the SIFT on the boundary region only, and 1500 for the HOG features. The optimum size radius is 15 and 10, for the internal and boundary SIFT respectively. The optimum spacing is 10 for the internal SIFT and 10 for the boundary SIFT.

Once the parameters have been determined on the validation set, the classifiers are retrained using all the available training data (both the training and validation sets). At this stage the weights β_f are determined using the optimization method of Varma and Ray [16].

5 Results on the test set

Table 1 shows the classification performance for different feature sets. It can be seen that combining all the features results in a far better performance than using the single best features (SIFT internal). Both the internal SIFT and the boundary SIFT contribute to the performance. The internal SIFT individual performance is however much better, which is to be expected as non-rigid deformations of the petals affect the boundary more than the inside. Figure 1 shows the weights learnt for each one-vs-rest classifier. Each class is represented by one image and the bar below each image show the distribution of weights for the different features. Blue represent the weight for the internal SIFT, red for the HOG features, green for the colour features and black for the boundary SIFT. The bar results show that overall the internal SIFT features are the most discriminative, and have the largest weight for most classifications. However, there are some classes (shown in figure 4) for example Cautleya, that have zero weights on the internal SIFT. These have much more weight put on both the HOG features and the SIFT sampled along the boundary, i.e. they are much better distinguished by the boundary of the petals and the overall layout of the petals than the internal texture. It also shows that some classes like Snapdragon, are not well distinguished by colour, but for the orange Dahlia the colour weight is relatively strong. This is because the Snapdragon occurs in many different colours, whereas the orange dahlia only occurs in one colour. Figure 5 shows examples of images that are classified correctly by the combination of features, but would be misclassified by some of the features independently. Each row shows the classification for an image (green indicates correct). The last row, for example, shows a case where no single features are classified correctly but the combination of all three results in a correct classification. Figure 6 shows examples of misclassifications.

6 Comparison with previous work

In this section we compare the performance of our method on the publicly available 17 class flower dataset

Figure 5: Example classifications. Each column shows the classifications for each feature given a test image(top row). The classification is indicated by a random image from that class (i.e. not the "closest" image to the test image). A green/thick border indicates correct classification and a red/thin border incorrect classification. The first column, for example, shows an image where colour is the only single feature that classifies it correctly, but the combination of feature still recognizes it. The last column shows an example where none of the features gets it right, but the combination of all features still gets it right. Note how each feature captures the aspect of the flower it was designed for, such as local shape with SIFT, global shape with HOG.

Features	Recognition rate
HSV	43.0%
SIFT internal	55.1%
SIFT boundary	32.0%
HOG	49.6%
HSV + SIFT int	66.4%
HSV + SIFT bdy	57.0%
HSV + HOG	62.1%
SIFT int + SIFT bdy	58.6%
SIFT int + HOG	66.4%
SIFT bdy + HOG	55.3%
HSV + SIFT int + HOG	71.8%
HSV + SIFT int + SIFT bd + HOG	72.8%

Table 1: Recognition performance on the test set. It can be seen that combining the features within the kernel framework improves the performance.

Figure 4: Example images from different classes.

`http://www.robots.ox.ac.uk/~vgg/data/flowers/index.html`, which was introduced in [12], with the two previous publications that have classified this dataset: Nilsback and Zisserman [12], and Varma and Ray [16]. Again, we report a class average classification for the overall recognition performance. Note, in [12] the performance measure was a weighted rank, aimed at retrieval. We report here instead the results of a first nearest neighbour classifier using χ^2 as the distance function.

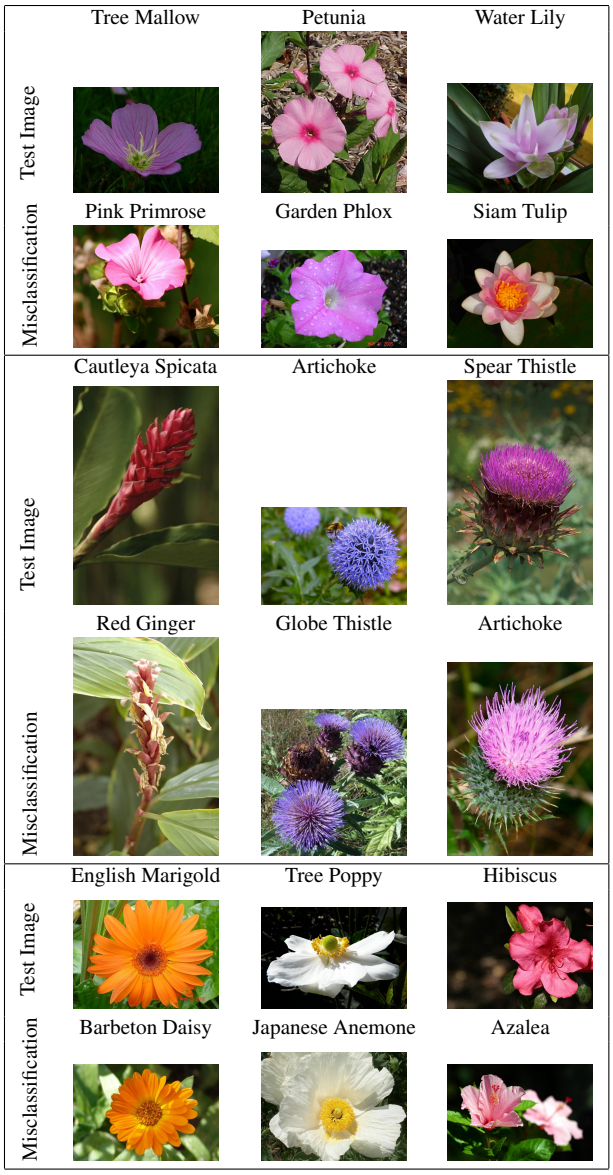

Figure 6: Examples of misclassifications. Note, the image shown for the misclassification is only a representative image for that class and not the "closest" image.

For the comparison we first use the segmentation method (graph cuts), described in [12], to obtain the foreground flower regions. In [12], the features are visual word histograms for colour, shape and texture. The nearest neighbour classifier using a weighted distance on the three histograms (as described in [12]) gives a recognition performance of $71.76 \pm 1.76\%$. Using the same features, but a multiple kernel classifier, [16] achieves a recognition performance of $82.55 \pm 0.34\%$, showing that this is a superior

classifier. Using the features computed in this paper and again with a multiple kernel classifier, increases the performance to $85.1 \pm 1.19\%$, demonstrating the benefit of the additional features introduced here.

We also do a comparison using the iterative segmentation scheme used in this paper (from [13]). We recompute the shape, colour and texture descriptors of [12] on the new segmentations. The weights are again optimized as in [12]. This gives a recognition performance of $73.14 \pm 1.76\%$. Again performance is improved by using a multiple kernel classifier, which gives a recognition performance of $83.33 \pm 1.39\%$. Finally, using the features computed in this paper and the multiple kernel classifier leads to a performance of $88.33 \pm 0.3\%$. This is the best performance todate reported on the 17 class flower dataset.

Although using the features computed in this paper leads to improved classification with both the graph cut segmentation and the iterative segmentation scheme, the improvement is more evident using the segmentation scheme of [13]. This is mainly because the SIFT features computed from the boundary are sensitive to the segmentations.

7 Conclusion

We have shown that by combining features in an optimized kernel framework we can improve the classification performance of a large dataset of very similar classes. The learning of different weights for different classes enables us to use an optimum feature combination for each classification. This allows us to incorporate, for example, that some classes are very similar in shape but different in colour and that some classes are better distinguished by the overall shape than the internal shape and vice versa. The principal challenge now lies in the large variations within a class and the relatively few samples of images. Future work should include using visually similar classes to jointly train the classifier.

Acknowledgements

We are grateful for expert assistance in ground truth labelling the flower classes from Radhika Desikan, Liz Hodgson and Kath Steward. This work was funded by the EC Marie-Curie Training Network VISIONTRAIN, Microsoft Research and the Royal Academy of Engineering.

References

[1] F. R. Bach, G. R. G. Lanckriet, and M. I. Jordan. Multiple kernel learning, conic duality, and the smo algorithm. In *Proc. ICML*, 2004.

[2] A. Bosch, A. Zisserman, and X. Munoz. Image classification using random forests and ferns. In *Proc. ICCV*, 2007.

[3] A. Bosch, A. Zisserman, and X. Munoz. Representing shape with a spatial pyramid kernel. In *Proc. CIVR*, 2007.

[4] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images. In *Proc. ICCV*, volume 2, pages 105–112, 2001.

[5] N. Dalal and B. Triggs. Histogram of oriented gradients for human detection. In *Proc. CVPR*, volume 2, pages 886–893, 2005.

[6] M. Das, R. Manmatha, and E. M. Riseman. Indexing flower patent images using domain knowledge. *IEEE Intelligent Systems*, 14(5):24–33, 1999.

[7] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. In *IEEE CVPR Workshop of Generative Model Based Vision*, 2004.

[8] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical Report 7694, California Institute of Technology, 2007.

[9] S. Lazebnik, C. Schmid, and J. Ponce. Beyound Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proc. CVPR*, 2006.

[10] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[11] T. Malisiewicz and A. Efros. Recognition by association via learning per-exemplar distances. In *Proc. CVPR*, 2008.

[12] M.-E. Nilsback and A. Zisserman. A visual vocabulary for flower classification. In *Proc. CVPR*, volume 2, pages 1447–1454, 2006.

[13] M.-E. Nilsback and A. Zisserman. Delving into the whorl of flower segmentation. In *Proc. BMVC.*, volume 1, pages 570–579, 2007.

[14] T. Saitoh, K. Aoki, and T. Kaneko. Automatic recognition of blooming flowers. In *Proc. ICPR*, volume 1, pages 27–30, 2004.

[15] B. Scholkopf and A. Smola. *Learning with Kernels*. MIT Press, 2002.

[16] M. Varma and D. Ray. Learning the discriminative power-invariance trade-off. In *Proc. ICCV*, 2007.

[17] H. Zhang, A. C. Berg, M. Maire, and J. Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Proc. CVPR*, volume 2, pages 2126–2136, 2006.

[18] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: a comprehensive study. *IJCV*, 73(2):213–238, Jun 2007.