The Delicate Art of Flower Classification

Paul Vicol
Simon Fraser University University
Burnaby, BC
pvicol@sfu.ca

Note: The following is my contribution to a group project for a graduate machine learning course (CMPT 726). The other group members were Loong Chan, Walther Maciel, and Juan Sarria.

Abstract

In this paper, we study the feasibility of identifying flowers in real-world Flickr photos. Established datasets for flower recognition contain only close-up, centered images of flowers, which are not representative of the large variety found on Flickr. We introduce a new dataset of close-up images that has greater variation in the orientation, shape, colour, and lighting than existing datasets. We also introduce a new dataset that contains both close-up and far-away images of certain species of flowers. We show that it is possible to identify fields of flowers, as well as individual flowers, in images downloaded from Flickr. This could provide a way to mine Flickr for information about nature, that could impact our understanding of the consequences of climate change.

1 Introduction

The popularity of photo-sharing sites like Flickr and Instagram shows that people are keen to capture the many facets of their environment. As pointed out in [1] and [2], photos of the natural world can be used to assess the geographical distribution of plants and animals, as well as the timing of events such as blooming of flowers, animal migrations, or animal nesting. As the location and timing of various natural elements is affected by the climate, Flickr can be thought of as a repository of information about climate change. Mining Flickr images for spring flowers is challenging because it has to take into account the variability inherent in such pictures: the distance to the flower, the position of the flower, and the lighting conditions around the flower. In this paper, we test the feasibility of flower recognition on close-up images downloaded from Flickr, and present our results of flower identification in pictures taken from a distance, such as images of fields or bunches of flowers.

2 Previous Work

Previous research in flower classification has focused on recognizing up-close, centered images of individual flowers. A widely used dataset for this problem is the Oxford Flowers dataset, which contains 17 different species of flowers, with 80 images per species (for a total of 1,360 images).

In [3], the authors first segment images to separate a flower in the foreground from its background, and then they describe the foreground using bag-of-words representations over colour, shape, and texture vocabularies. They then perform classification using k-nearest-neighbours. Later work, in [4], uses SVMs with different kernels for classification, and introduces new features like SIFT.

3 Datasets

To address the problems related to the small size and low variability of the Oxford dataset, we collected two new datasets: 1) a larger dataset with the same 17 species found in the Oxford dataset, but with greater variation in appearance among the flowers in a class, and 2) a dataset containing up-close and far-away images for seven of the 17 species in the Oxford set.

3.1 Image Collection

We queried Flickr for images tagged with the names of the following 17 species of flowers from the Oxford dataset: bluebell, buttercup, coltsfoot, cowslip, crocus, daffodil, daisy, dandelion, fritillary, iris, lily of the valley, pansy, snowdrop, sunflower, tiger lily, tulip, and windflower. To increase the number of images, we also searched for these names in several other European languages. This yielded about 15,000 images.

3.2 A Larger 17-Species Dataset

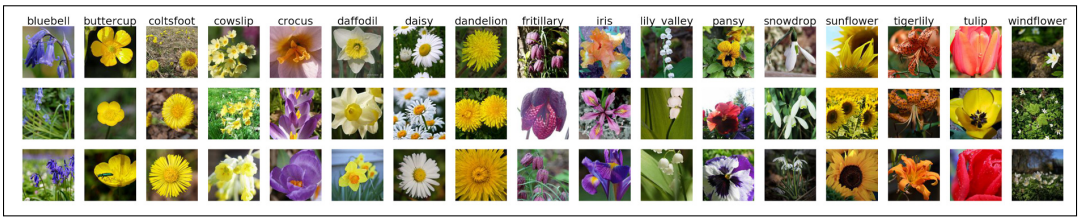

Figure 1: Sample of the 17 species in our close-up Flickr dataset

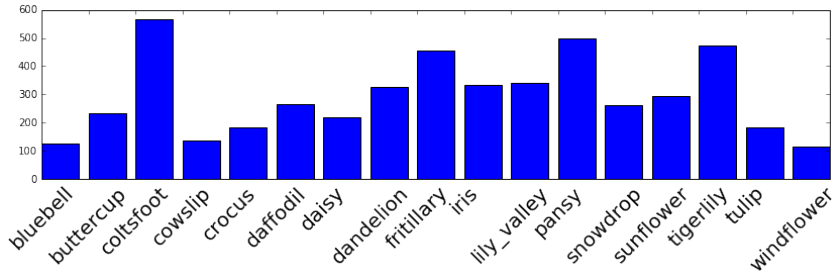

Figure 2: Class distribution for the larger 17-class dataset

For each species (Fig 1), we then eliminated the images obviously tagged wrong (e.g. bell peppers tagged as tulips), and then inspected the images that were left in order to eliminate all those that did not represent the proper species. The challenges here depended on the plant, and we tried, to the best of our ability, to eliminate the wrong flower species because we wanted to create a botanically correct dataset, which we could make publicly available. To differentiate between such species, we counted the number of petals of the flower and examined its leaves [5]. By following similar careful inspections of every image and eliminating incorrect species and duplicates, we collected about 7,000 images, containing hundreds of images for each of the 17 species. We then eliminated any images of fields of flowers, retaining only 5,111 close-up pictures for all 17 species. This is our 17-species dataset, which we built as an extension of the Oxford dataset. Our set contains the same species, but is larger (5,111 images compared to 1,360 in the Oxford dataset) and has a higher variability (Fig 2).

3.3 A Near/Far Dataset

For humans, recognizing flowers from images is fairly easy if the images are taken up close. Because of this, most of the Flickr photos consist of close-ups of one or two flowers. Photographers seem

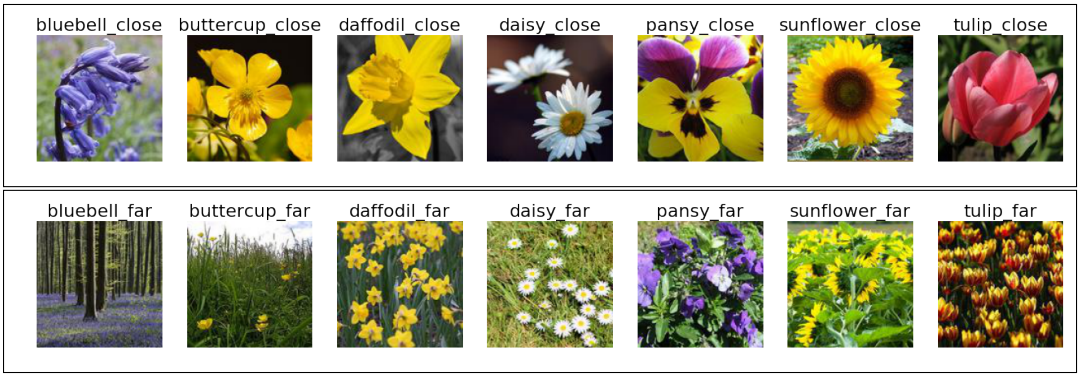

Figure 3: Sample close-up and far-away images of the seven species in the close/far dataset

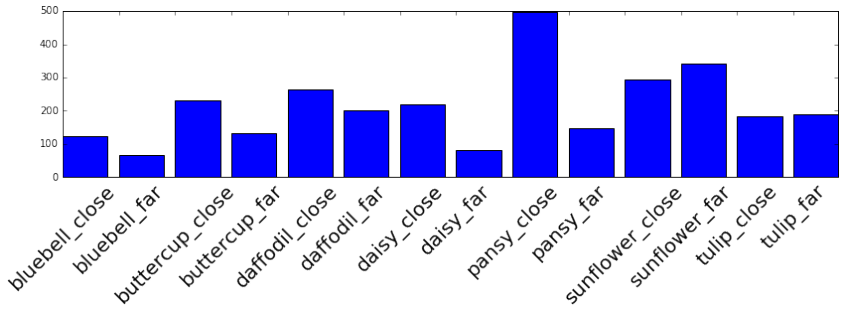

Figure 4: Class distribution for the near/far dataset

to want the flowers in their photos to be recognized, and for that they often need to go down to the flower level and snap close-ups.

However, sometimes a whole field consisting of a single species of flower has a distinct appearance, and photographers realize that these flowers could be recognized even from a distance. For example, a person can easily recognize an image of a field of sunflowers or dandelions. Identifying flowers in distant images is an interesting task. For example, some Flickr photos of fields of flowers can be ascribed to the most likely species from cues given by the surrounding environment: bluebell fields are always on the forest floor, so trees are usually visible in the picture.

To build a near/far dataset (Figure 3), we divided the 7,000 curated images of the 17 species into those that show close-ups (the near category), and those that show fields or bunches of flowers (the far category). We found that, for some species, there were very few far images (e.g. for fritillary, tiger lily, coltsfoot). We selected seven species (from the original 17) for which we had the largest number of far images, and included these far images together with those species near images in our near/far dataset (Figure 4).

4 Experiments

We focus on the most exciting problem first: classifying near and far images of the seven different species in our near/far dataset. Then, we use the same methods on our large 17-class dataset of close-ups, to see if we can classify more varied, real-world images from Flicker. We also evaluate our classification methods on the original Oxford dataset to compare performance against the techniques used by the authors in [3].

4.1 Features

We use the following global and local descriptors as features to train our models:

Tiny Images: We subsample images to 16×16 pixels, giving 256 pixels per RGB color plane and yielding a 768 dimensional feature vector. The resized images are less sensitive to alignment, and can be used to represent colour distributions in broad regions of the image.

Colour Histograms: We create colour histograms using the hue channel in HSV color space, using 200 bins, to yield a 200-dimensional feature vector. We also tried 100 and 300 bins, but we found that 200 bins yielded the best cross-validation results.

SIFT: SIFT features represent shapes within local regions. We extract SIFT features using a SIFT keypoint detector and cluster them into 200 clusters using k-means to create a 200-word shape vocabulary. This essentially finds the 200 "most important" local features across all training images. To construct a bag-of-words for an image over this vocabulary, we quantize each SIFT feature for the image into one of the 200 clusters, and construct a histogram over the number of occurrences of each visual word in the image. Then, each image can be represented by a 200-dimensional histogram over this vocabulary.

GIST: GIST features capture coarse texture and scene layout. This feature yields a 960-dimensional vector.

We normalize each feature to have zero mean and unit variance. In our experiments, we also use combinations of features, which are formed by concatenating normalized feature vectors.

4.2 Identifying Flowers in Near/Far Images

We experimented with three different classification methods: 1) k-nearest-neighbours; 2) SVMs; and 3) convolutional neural networks. We try to classify flowers from our 7-species dataset, containing 3,078 images of near and far flowers. We split this set into a training set of 2,313 images and a test set of 765 images.

4.2.1 K-Nearest-Neighbour Classifiers

We start with one of the simplest models for classification: k-nearest-neighbours. We experimented with using individual features for classification, as well as all features together. For each feature, we cross-validated a KNN classifier on three parameters: 1) the number of neighbours, k; 2) the type of distance, which can be L1 or L2 (Euclidean); and 3) the *weighting* of the nearest neighbours in making a classification decision. There are two possibilities for the weighting: with *uniform* weighting, each of the k nearest neighbours is given an equal vote in deciding the class of a new example, while with *distance* weighting, closer neighbours are given more voting power. We found that using L1 distance combined with distance-based weighting yielded the best cross-validation results for every feature (Table 1).

Table 1: KNN classifiers trained on different features

Feature	Best K	Best CV Accuracy	Test Accuracy
Random Baseline	-	-	7.14%
Tiny 16x16	3	37.4%	36.1%
HSV Histograms	9	47.8%	47.1%
GIST	7	59.5%	60.9%
Tiny + HSV + GIST	7	64.0%	66.0%
SIFT	19	46.3%	43.7%
All Features	**9**	**66.4%**	**68.2%**

4.2.2 Support Vector Machines

We use the Support Vector Classifier from scikit-learn, which constructs $k(k-1)/2$ one-vs-one classifiers, each between two specific classes, and classifies an example by considering the predictions of each classifier. We trained SVMs on our features, using linear and RBF kernels. We present the best accuracies obtained with either kernel (Table 2). The confusion matrix over all features is shown in Figure 5.

Table 2: SVMs trained on different features

Feature	Kernel	Average CV Accuracy	Test Accuracy
Random Baseline	-	-	7.14%
Tiny 16x16	RBF	48.2%	50.6%
HSV Histograms	RBF	41.8%	41.4%
GIST	RBF	68.9%	70.5%
SIFT	RBF	53.5%	49.4%
Tiny + HSV + GIST	RBF	70.9%	73.6%
All Features	**Linear**	**76.2%**	**77.6%**

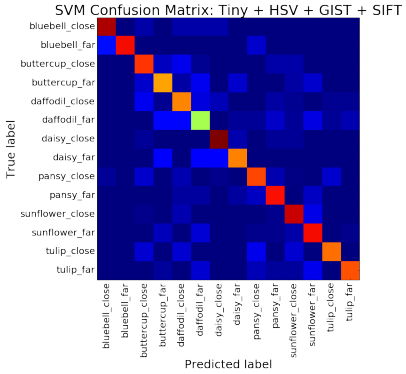

Figure 5: Confusion matrix for an SVM trained on all features

4.2.3 Convolutional Neural Networks

We also trained a deep network from scratch using Keras, using real-time data augmentation that rotated images randomly, and flipped images horizontally. This yielded a test accuracy of **78.6%** after 60 epochs (Figure 7).

4.3 Delving Deeper into Near vs Far

In the classification problem described in the previous section, the computer has to decide both the species of a flower and whether it is close up or far away. A number of questions naturally arise, including: Is it harder to identify species in close-up images or far-away images?

4.4 Classifying Near and Far Images

The near/far dataset has separate classes for up-close and far-away images of 7 different flower species; thus, there are 14 classes: bluebell_close, bluebell_far, buttercup_close, buttercup_far, daffodil_close, daffodil_far, daisy_close, daisy_far, pansy_close, pansy_far, sunflower_close, sunflower_far, tulip_close, and tulip_far.

In general, near and far images of the same species are visually distinct, but there are still some similarities (i.e., the characteristic colour). Some features may cause the classifier to misclassify distance while correctly classifying the species; for example, it may classify a far bluebell as a close bluebell, or vice versa, probably because both are blue.

In many situations it is sufficient to identify species without caring about distance. Thus, we measure performance on this dataset in several different ways: 1) the *overall accuracy* is the accuracy on the original 14-class problem – the percentage of testing examples that are correctly classified with respect to species and distance; 2) the *species accuracy* is defined as the percentage of testing examples for which the system correctly identified the species (e.g. buttercup) regardless of whether distance was correct – we compute the species accuracy by pooling the output of the 14-class classifier such that near and far classes for a particular species are both considered correct, so

Figure 6: Sample of Incorrectly Classified Images

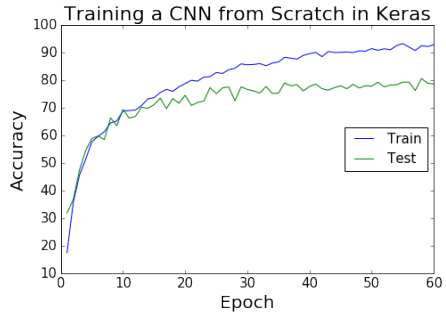

Figure 7: Accuracies while training a CNN

this essentially converts the problems into a 7-class classification over species; and 3) out of interest more than for practical significance, we compute the *distance accuracy*, defined as the percentage of examples that were correctly classified as near or far, regardless of the species. We calculate this accuracy by pooling the output of the 14-class classifier such that all near and all far images of any species are the same; this essentially converts the problem into 2-class classification over distance.

Another way to obtain the species and distance accuracies is to group images into different classes from the start, and train classifiers on these classes. We tried this approach as well, and found that this gives very similar results to the pooling method described above (Table 4).

Table 3: Comparing species and distance accuracies

	Species Accuracy		**Distance Accuracy**	
Feature	**Pooled**	**7-Class**	**Pooled**	**2-Class**
Random Baseline	14.3%	14.3%	50%	50%
Tiny 16x16	59.5%	60.5%	80.7%	82.1%
HSV	55.2%	56.2%	68.9%	69.0%
GIST	75.4%	74.8%	91.5%	90.5%
Tiny + HSV + GIST	78.8%	78.7%	**92.0%**	**91.1%**
SIFT	54.6%	56.7%	85.8%	86.1%
All Features	**83.1%**	**80.5%**	90.1%	86.5%

4.5 Identifying Diverse Close-Ups from our 17-Class Dataset

We also tested these approaches to identify flowers from our 17-species dataset, containing 5,111 close-up images of flowers. We split this set into a training set of 3,833 images and a test set of 1,278 images. Table 5 presents the results. We note that all the features perform significantly better than the random baseline.

On our expanded 17-class dataset, we obtain **top-1**, **top-3**, and **top-5** classification accuracies of **77.9%**, **92.0%**, and **96.9%**, respectively. Using the same techniques to classify the Oxford dataset, we obtain **top-1** and **top-5** accuracies of **73.2%** and **95.3%**, respectively.

Table 4: SVM classifiers for our 17-class Flickr dataset

Feature	Kernel	Average CV Accuracy	Test Accuracy
Random Baseline	-	-	5.88%
Tiny 16x16	RBF	51.0%	51.3%
HSV	RBF	46.4%	47.5%
GIST	RBF	65.5%	66.6%
Tiny + HSV + GIST	RBF	69.8%	71.9%
SIFT	RBF	42.4%	44.5%
All Features	**Linear**	**74.3%**	**77.9%**

5 Conclusion

We have shown that it is possible to classify both up-close and distant flowers from real-world photos from Flickr. Future work can address a few limitations of our study: 1) labeling flowers as 'far' or 'near' was at times inconsistent, so the 'far' set contains a few images very similar to those we placed in the 'near' set; and 2) the size of our classes is not uniform.

Acknowledgements

We would like to thank Walther Maciel for downloading some of the images that we included in the datasets.

References

[1] Wang, J., Korayem, M. & Crandall, D. J. (2013). Observing the natural world with Flickr. ICCVW 2013, 452-459.

[2] Zhang, H. Korayem, M. & Crandall D. J. (2012). Mining photo-sharing websites to study ecological phenomena. In WWW.

[3] Nilsback, M.E. & Zisserman, A. (2006). A visual vocabulary for flower classification. CVPR.

[4] Nilsback, M.E. & Zisserman, A. (2008). Automated flower classification over a large number of classes. ICVGIP.

[5] Ontario Wildflower Database. http://www.wildflowersofontario.ca