Leaf Image Classification Using SIFT Features and Bag of Visual Words

Daniël van der Hall Joram Koiter Maarten van der Velde Olaf Visker

Department of Artificial Intelligence, University of Groningen, The Netherlands

Abstract

In the study of plant life, specifically trees, one important characteristic is the recognition of leaves. Due to the wide variety of shapes in which leaves appear, as well as their ease of transportation and photography, they have become one of the chief methods to visually determine the species of a tree or other plant. Modern software applications on smartphones provide a way of performing classification of a leaf, in the field, within seconds of it being photographed. We have developed a program with the goal of analysing leaf images that were obtained from existing datasets of digitized leaf photographs. Features were extracted from each image using the scale- and rotation-invariant SIFT method, after which the images were described using histograms obtained with a visual Bag-of-Words implementation. These histograms served as input to an artificial neural network which aimed to provide correct identification of the leaves' species. A comparison was made between the neural network and a K-Nearest Neighbour classifier that used the same input. These systems were gauged on accuracy and compared to similar implementations from the literature which differed in both classification method and dataset used.

1 Introduction

Despite the advent of more precise methods, the use of leaves to identify trees and other plants persists. The variance of shapes between species with a general uniformity of shape within one species, combined with the portability and size of the leaves, has ensured their popularity for this usage. Whereas leaf identification used to require manual comparison of the leaf (or a photograph of it) to images of previous leaves (often involving large decision trees), advances in computer vision have made fast, large scale comparison of images feasible [1]. One example of the use of computer image classification for leafs is the LeafSnap app [6]: a mobile app covering all 185 species of the northeast USA, it enables users to photograph leaves and immediately classify them, by transmitting the captures image to a server which houses the recognition system. After classification, the user is presented with a sorted list of identification results from which they can pick the one that most resembles their leaf. Total time to solution after uploading of the image is 5.4 seconds [6].

We decided to develop a system with a similar goal, to identify the species of a leaf by using Machine Learning techniques to compare it with a processed database of labelled examples. The resulting system is capable of performing the whole process: feature extraction from an image database and using the extracted features to train a classifier. We used a publicly accessible data set that was used in the ImageClef 2012 leaf classification challenge, earlier instalments of which have been used before in comparable efforts to make an image classification system [2]. These systems usually innovate most in the mechanism chosen for feature extraction; classification of the extracted features is often achieved using established methods. For example, earlier work [5] used a combination of geometric features such as contour information and moment invariants, among others, and a linear discriminant classifier. We decided to limit ourselves to one feature extraction method: a combination of Scale Invariant Feature Transform (SIFT) descriptors and Bag of Visual Words [12], and focus our efforts on the classification phase by using a custom Neural Network and comparing it to a simple K-Nearest Neighbour classifier. We expected the neural network to come out on top in this comparison.

Our expectations were modest, emphasizing the development of a working system that would run on a normal PC, and be capable of classifying leaf images at a rate substantially higher than that of a random guess. Other, more advanced, programs have been shown to be in the 90% accuracy range [11, 5]. However, the simpler approach taken in this study may have its benefits too. Speed of classification is of paramount importance, especially when the classifier is used in a real-world context, such as in a smartphone app. Users may be more than willing to trade in some accuracy if it means that they get an answer more quickly. The computational limitations of small devices such as smartphones may also tip the balance towards a classifier that uses its resources sparingly. The methods used in this study may be seen as an attempt to address these concerns, by exploring the effect on performance of a stronger emphasis on simplicity.

Finally, it is important to note that accuracies are not easily compared across the various systems mentioned here, as leaf classification systems such as LeafSnap tend to present a top 5 of solutions; in some cases, having the correct class appear in the top N list is counted as a successful classification [11].

2 Methods

2.1 Data set

Images from the ImageCLEF2012 Plant Identification Task were used in this study [4]. The data set consists of high resolution colour images that are labelled with metadata such as the full taxon name of the plant, its common name, and the GPS coordinates of the observation. The images vary in resolution and aspect ratio but are scaled so that their longest axis is 800 pixels. The data set originally contains three types of images: *scans*, scan-like photographs (*pseudoscans*), and normal *photographs*. Examples of these three types are shown in Figure 1. They differ in the complexity of their backgrounds: *scans* have a purely white background and few shadows, whereas *pseudoscans*, while maintaining a uniform background, are more variable in the colour of their background and the lighting conditions. Finally, *photographs* have very diverse backgrounds, often with other plants visible, and vary strongly in their lighting conditions.

Figure 1: Examples of the three image types in the original ImageCLEF2012 data set. From left to right: *photograph*, *pseudoscan*, and *scan*.

Because of the complexity associated with extracting the leaf from its background, only *scan* images (57% of all images) were used in this study. This subset consists of 6630 images of leaves from 75 classes. The number of images varies strongly per class, from a single image (*Xanthium strumarium*), to 367 (*Ulmus minor*). To resolve this imbalance only images from the 32 most frequent classes were used in this study. The least frequent of these 32 classes (*Punica granatum*) still appeared in 134 images.

These images were divided into a training set of 2550 images ($\pm 73\%$) and a test set of 965 images ($\pm 27\%$).

2.2 Feature extraction

Several image classification studies [10, 7], including at least one study into leaf image classification [11], have made use of Scale-Invariant Feature Transform (SIFT) descriptors. These descriptors are invariant to changes in scale and rotation, and somewhat robust to variations in viewpoint and illumination. This makes them suitable for this task, since the leaves are photographed at varying scales and rotations.

The first step in classifying leaves is extracting utilizable information from the processed image, this is called feature extraction. This procedure tries to find informative values in the image and converts them into a feature vector. To assist the feature extractor the image is first converted to a greyscale. This is done to reduce the dimensionality of the RGB-image from three dimensions (Red Green and Blue), to one dimension. This significantly reduces the processing cost and therefore speeds up the process. The parameters used for the weighted greyscale conversion are $Y = 0.299 * R + 0.587 * G + 0.114 * B$. These are the values provided by the openCV library, an open-source computer vision library. After this a mask is applied to inform the feature extractor of the relevant part of the image. This mask is an inverted threshold to zero mask, this means that every pixel with a grey value above a threshold is set to a value of 0 (black). This threshold, established empirically, was set to 200. After this preprocessing the image is ready for the feature extractor. As mentioned in the introduction, the feature extraction algorithm applied in this research is the Scale-invariant feature transform algorithm, or in short: SIFT. This algorithm was chosen for its ability to swiftly localize keypoints at areas with a high variation, while being scale- and rotation invariant. This invariance is important for the classification of leaves because of the various ways photos can be taken by the users. Also SIFT proves to be a proper algorithm for different viewpoints and varying illumination. To acquire keypoints SIFT uses Difference of Gaussians, DoG finds points by subtracting Gaussian blurred versions of the image from each other, this is done at various scales. This finds local extrema (maxima or minima) in the image, the keypoints. The Difference of Gaussians is calculated by the substraction of the image by the convolution of the image with the Gaussian of variance:

$$\delta DoG = I * \frac{1}{\sigma_1\sqrt{2\pi}} e^{-(x^2)/(2\sigma^2)} - I * \frac{1}{\sigma\sqrt{2\pi}} e^{-(x^2)/(2\sigma^2)}$$

This gives keypoints like the onees depicted in Figure 2 on the next page.

2.3 Bag of visual words

In spite of the SIFT descriptors' robustness to scale and rotation changes, it is unrealistic to expect features to be exactly identical between leaves, because of natural variation between leaves, and variation in the images that cannot be accounted for by SIFT. Nevertheless we expect there to be certain natural features for each class that are recurrent in images of leaves of that class. A good way to discover these features from the 'noisy' SIFT descriptors is to use a clustering approach. Each of the clusters that are found can then be considered to represent a certain feature that occurs multiple times in the data, albeit with some variation.

This study uses K-Means clustering with Euclidean distance as its distance metric [8]. A random subset of the total set of extracted SIFT descriptors is used for this process to mitigate the high computational cost caused by the high dimensionality of the SIFT descriptors (128 dimensions) and the large number of these descriptors.

Extracting clusters from the SIFT descriptors in this way allows for a much simpler description of the data. Each cluster centre represents a *visual word* (or code word). Together the set of cluster centres forms a dictionary (or bag) of visual words. Using this dictionary any image in the data set can be described in terms of its visual words. This process starts by extracting the SIFT descriptors from the image. These descriptors are then individually assigned to the existing clusters nearest to them using K-Nearest Neighbour. This step translates the descriptors from ones that are specific to the image to more general descriptors that represent certain visual words. Now that the image's features are expressed in terms of the visual words in the dictionary, a histogram can be made depicting the frequency of each visual word in the image. This histogram has a fixed length K — the number of visual words in the dictionary.

A challenge of this approach is to find the optimal value of K. If K is too small, the vocabulary will be too limited to accurately describe all features, since the clusters that form will be too general.

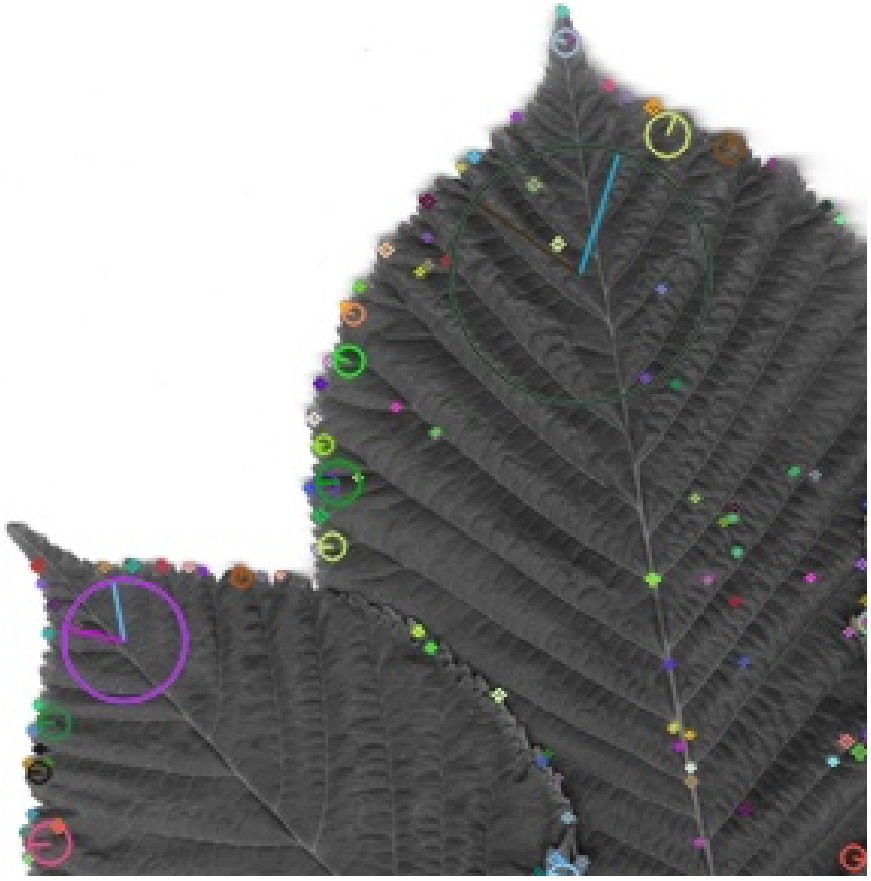

Figure 2: Keypoints found by the SIFT algorithm

Conversely, if K is too large, too much of the original noise will still be present in the clusters, as there will be many small clusters that describe small variations of the same underlying feature. This defeats the purpose of the bag-of-words approach by not really describing the images in more general terms.

In this study K was selected through a procedure in which classifier performance was compared for a range of values of K, between 50 and 250 (based on [9]).

2.4 Classifiers

2.4.1 Neural network

For this research a neural network, with backward propagation of errors, was used as the main classifier for the extracted image data. As our optimization method we used gradient descent. The network consisted of an input layer, one hidden layer and an output layer. As our classifier used supervised learning we needed to specify target output and compare it to the output given by the network. For the output we used N_{output} output nodes where $N_{output} = N_{leaftypes}$. The target consisted of a vector $V = [.., 0, 0, 0, ..]$ of length $N_{leaftypes}$ where $V[I_{leaftype}] = 1$. The index $I_{leaftype}$ was unique for every leaf type and corresponded to its index in the leaf type dictionary. The amount of input nodes was $N_{input} = Length_{histogram}$ The optimal amount of hidden nodes was $N_{hidden} = 100$ and were randomly initialized from a continuous uniform distribution of the interval $[-1, 1]$. As the activation function the sigmoid function was chosen, defined by

$$f(x) = \frac{1}{1 + e^{-x}}$$

Where its derivative is as follows:

$$f'(x) = f(x)(1 - f(x))$$

A learning value of 0.01 was used for all weights including those of the bias nodes of which there were 2, one connected to the hidden layer and one connected to the output layer. The option for regularization, that discourages large changes in weights, was also added, but later discarded as no improvements were shown for various values of the regularization lambda. Another change made was the removal of mini-batch learning in favour of online learning, although improving the time till convergence, it reduced accuracy and was therefore dropped from the network. In order to avoid the program to learn meaning from the order of input data we presented the data in a randomized order every epoch. This counteracts repeating update cycles.

2.4.2 KNN

A K-Nearest Neighbour classifier was made to serve as a comparison for the neural network's performance. The classifier finds the K nearest neighbours of a test image's bag-of-words histogram (using Euclidean distance) and assigns it the majority class. In case of a tie the class assignment is chosen randomly from the tied competitors. The classifier returns a list of the top T classes. Following [11, 1], a classification is considered to be correct if the ground truth class appears in the top 5 results.

3 Results

3.1 Neural network classifier

SIFT descriptors were extracted from each image, with a maximum of 100 descriptors per image. A random subset of 100 of the files containing these descriptors was used for the clustering step in the bag-of-words process, in which the descriptors were mapped onto 100 clusters. This number of clusters was empirically found to provide a good trade-off between accuracy and computation time. A subset of all SIFT descriptors was used to ensure clustering completed within a reasonable time span. The neural network, containing 100 hidden nodes, was run for 1,000 epochs with a learn rate of 0.1 and no regularisation. Classifications were considered correct if the ground truth class appeared in the top 5 results. Using this metric, the network's classification accuracy was 77.78%.

Figure 3 shows how the accuracy of the network improves quickly for the first 20 epochs, and then stabilises.

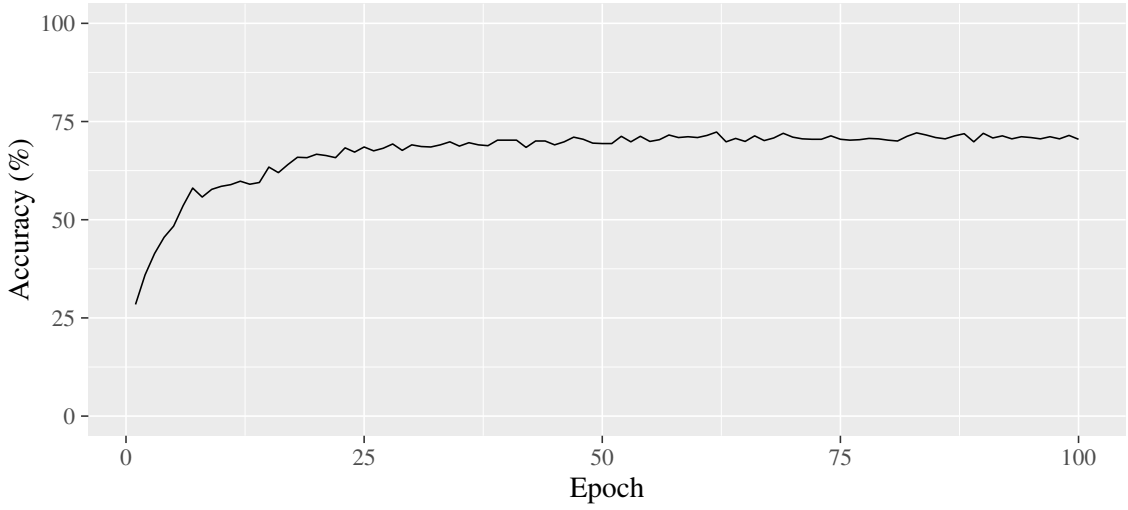

Figure 3: Classification accuracy of the neural network at various stages during training.

3.2 KNN classifier

As with the neural network classifier, a maximum of 100 SIFT descriptors were extracted from each image. Once again a random subset of 100 of the files containing these descriptors was used for creating

the 100 clusters of the visual bag-of-words. The classifier was run on the test data for values of K ranging from 1 to 100. The highest accuracy of 60.9% was achieved with $K = 32$.

Classifications were considered correct if the ground truth class appeared in the top T results. Figure 4 shows the accuracy of the KNN classifier at various values of T, with K ranging from 1 to 100. Performance remains more or less steady once K is above 10. The marginal gain in accuracy for larger values of T decreases rapidly with T — the step from $T = 4$ to $T = 5$ only improves the classifier's performance very slightly.

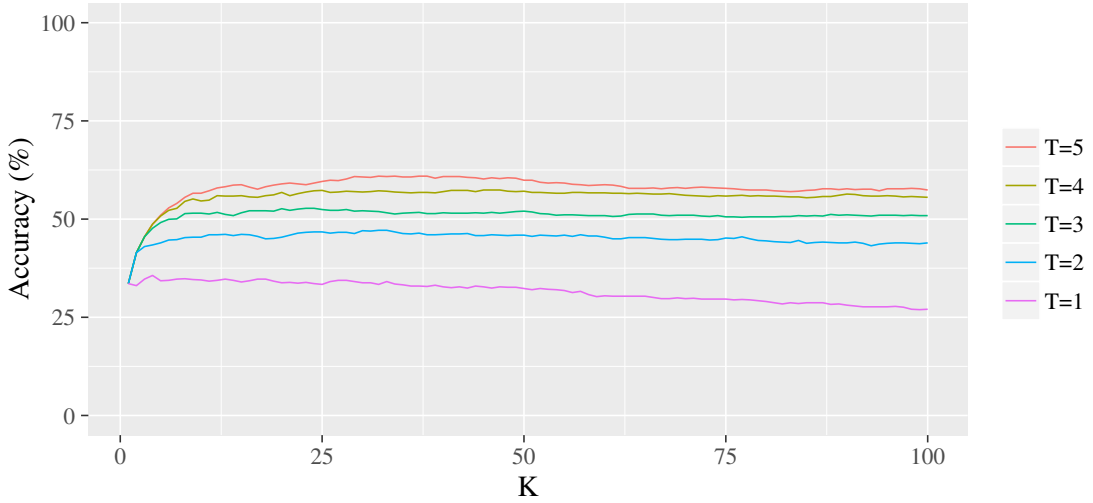

Figure 4: Accuracy of the KNN classifier at various values of T, for K from 1 to 100. A classification is considered correct if the ground truth class appears in the top T answers.

4 Discussion

The methods described in this study were an attempt at a computationally simple, yet effective classifier of leaf images. As expected, the neural network performed at a higher accuracy than the relatively much simpler K-Nearest Neighbour classifier to which it was compared. This difference of almost 20 percentage points shows that the additional complexity of the neural network does pay off with a marked improvement in performance. Of course, the higher accuracy comes at the cost of much higher computational demands. For these classifiers to be trainable and usable on low-powered devices such as smartphones, it is imperative to keep their limitations in mind. Which of these classifiers is considered superior depends on how much emphasis is placed on computational efficiency versus classification accuracy. However, when we limit ourselves to desktop computers, the more powerful neural network is the obvious choice.

Further improvement of the classifier's performance may be found in a more careful feature extraction process. Previous work has shown that the inclusion of shape context, in which a set of points on the edge of the leaf is selected, and each point's relative distance to all other points is recorded, can improve classifier performance [11]. In addition, it may be beneficial to take into account higher order relationships that may exist between the visual words in the leaf images [3].

Many of the issues that were encountered in this study had to do with aspects that we initially did not expect would become problematic. One problem was the initially long time needed for the training of the neural network, which could require multiple hours to complete. Although eventual improvements severely reduced the time needed, it still prevented us from performing exhaustive parameter tests early on. Despite this, we have proved that a neural network can classify images with a rate of accuracy far better than guessing, and in fact comparable with other efforts reported in the ImageClef 2012 results [4]. SIFT and Bag of Words have proved to provide a good pre-processing strategy, given the favourable results.

Aside from the details of the implementation, we also encountered unexpected decisions we had to make; those regarding the criteria for a successful classification for example, or the nature of the input. The latter point involved the choice of what types of images to include; just scans, or natural photographs as well?

In its current state, our program would probably require significant development to make it suitable for a mobile platform, but the system is functional. One step that should be added if functionality in the real world is to be achieved, is a mechanism to distinguish leaves from non-leaves. The next step could then be the creation of a desktop application that could train a neural network, which could then be brought into the field on a mobile app. The app itself would only need the functionality to pre-process a photograph taken with the device, and classify it. A final version would perhaps include the option to have classification done through Internet upload to a remote server, a method used by LeafSnap. In short, while the software is not very portable, it serves as a definite proof of viability for the concept.

References

[1] Peter N. Belhumeur, Daozheng Chen, Steven Feiner, David W. Jacobs, W. John Kress, Haibin Ling, Ida C. Lopez, Ravi Ramamoorthi, Sameer Sheorey, Sean White, and Ling Zhang. Searching the world's herbaria: A system for visual identification of plant species. In David A. Forsyth, Philip H. S. Torr, and Andrew Zisserman, editors, *ECCV (4)*, volume 5305 of *Lecture Notes in Computer Science*, pages 116–129. Springer, 2008.

[2] Hervé Goëau, Pierre Bonnet, Alexis Joly, Nozha Boujemaa, Daniel Barthelemy, Jean-François Molino, Philippe Birnbaum, Elise Mouysset, and Marie Picard. The clef 2011 plant images classification task. In Vivien Petras, Pamela Forner, and Paul D. Clough, editors, *CLEF (Notebook Papers/Labs/Workshop)*, volume 1177 of *CEUR Workshop Proceedings*. CEUR-WS.org, 2011.

[3] Kaiqi Huang, Chong Wang, and Dacheng Tao. High-order topology modeling of visual words for image classification. *IEEE Transactions on Image Processing*, 24(11):3598–3608, 2015.

[4] ImageCLEF. Plant Identification 2012 — ImageCLEF, 2012. [Online; accessed 4 February 2016].

[5] Cem Kalyoncu and Önsen Toygar. Geometric leaf classification. *Computer Vision and Image Understanding*, 133:102–109, 2015.

[6] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, and Jo ao V. B. Soares. Leafsnap: A computer vision system for automatic plant species identification. In Andrew W. Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *ECCV (2)*, volume 7573 of *Lecture Notes in Computer Science*, pages 502–516. Springer, 2012.

[7] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision, Corfu*, 1999.

[8] James MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA., 1967.

[9] Quang-Khue Nguyen, Thi-Lan Le, and Ngoc-Hai Pham. Leaf based plant identification system for Android using SURF features in combination with Bag of Words model and supervised learning. In *Advanced Technologies for Communications (ATC), 2013 International Conference on*, pages 404–407. IEEE, 2013.

[10] Yu Wang, Xiaojuan Ban, Jie Chen, Bo Hu, and Xing Yang. License plate recognition. *Optik*, (126):2895–2901, 2015.

[11] Zhiyong Wang, Bin Lu, Zheru Chi, and David Dagan Feng. Leaf image classification with shape context and sift descriptors. In Andrew P. Bradley and Paul T. Jackway, editors, *DICTA*, pages 650–654. IEEE Computer Society, 2011.

[12] Yin Zhang, Rong Jin, and Zhi-Hua Zhou. Understanding bag-of-words model: a statistical framework. *Int. J. Machine Learning & Cybernetics*, 1(1-4):43–52, 2010.