

# 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 minutes of it being photographed. For this assignment, we have developed a program with the goal of analyzing 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, and were then filtered and clustered using a visual Bag of Words algorithm. The resulting feature sets were then classified using an artificial neural network to provide correct identification of the leaf's species. The performance of this system was gauged on accuracy, and then compared with other such implementations from the literature which differed in both classification method and dataset used.

This is the abstract. It is imported from the file `abstract.tex`.

## 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 leaves is the Leaf Snap app [5]: 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 [5].

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 labeled 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 two available data-sets, one of which was part of the ImageClef 2012 leaf classification challenge, earlier installments 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. [4] use a combination of geometric features such as contour information and moment invariants, among others, and use a linear discriminant classifier.

We decided to limit ourselves to one feature extraction method: Scale Invariant Feature Transform (SIFT), and focus our efforts on the classification phase by using visual Bag of Words and a custom Neural Network for classification.

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 substantially better than a random guess. Other programs have been shown to be in the 90 % accuracy range [10, 4]. It is important to note here that the accuracies are not easily compared across systems, as leaf-classification systems such as LeafSnap tend to present a top 5 of solutions; in some cases, having the correct class in the top N list is counted as a successful classification [10]. This is the introduction. It is imported from the file `introduction.tex`.

## 1.1 Instructions

### 1.1.1 Making the PDF

Run `makePDF.sh` (Linux, OS X) or `makePDF.bat` (Windows) in the terminal to make a pdf of the whole paper. This script also takes care of changes to the bibliography, so you only need to run it once.

### 1.1.2 Editing

Each section of the paper has its own `.tex` file that is imported into the main `ML_leaf_report.tex` file, so the sections can be worked on simultaneously without messing up the main file (hopefully).

### 1.1.3 Preamble

Finally, the file `preamble.sty` contains all package imports in one place.

## 1.2 Another subsection

More text goes here.

# 2 Methods

## 2.1 Data set

Images from the ImageCLEF2012 Plant Identification Task were used in this study [3]. 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 on the following page. 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.

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 had 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 [9, 6], including at least one study into leaf image classification [10], 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.



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

## 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 [7]. 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. 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 [8]).

## 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  $N_{output} = N_{leaf types}$ . The target consisted of a vector  $V = [.., 0, 0, 0, ..]$  of length  $N_{leaf types}$  where  $V[I_{leaf type}] = 1$ . The index  $I_{leaf type}$  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 [10, 1], a classification is considered to be correct if the ground truth class appears in the top 5 results.

### 3 Results

#### 3.1 Parameters

The optimal number of visual words used in the bag-of-words process was found to be 100.

#### 3.2 Neural network classifier

#### 3.3 KNN classifier

As with the neural network classifier, a maximum of 100 SIFT descriptors were extracted from each 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. 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 2 on the next page 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.

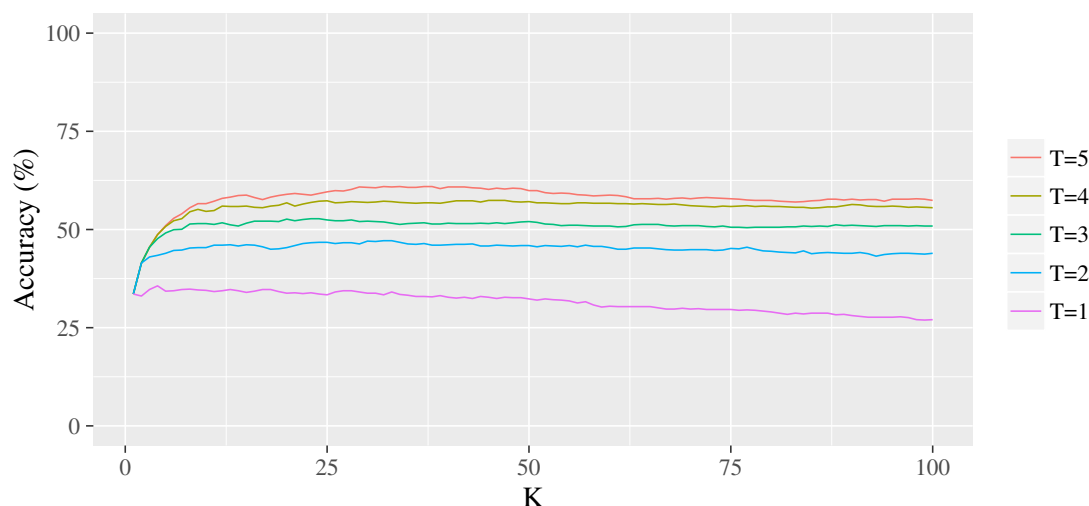


Figure 2: Accuracy of the KNN classifier at various values of  $T$ , for  $K$  from 1 to 100. A classification is considered correct if the ground truth class appears in the top  $T$  answers.

## 4 Discussion

This is the discussion section. It is imported from the file `discussion.tex`.

Elements of the discussion:

Were the results as expected? Comparison with other literature.

Is this useful in a practical sense? Did we make any unrealistic assumptions?

Problems we encountered and possible solutions or ways of mitigating their effects.

Final thoughts, suggestions for further improvement.

## 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] ImageCLEF. Plant Identification 2012 — ImageCLEF, 2012. [Online; accessed 4 February 2016].
- [4] Cem Kalyoncu and Önsen Toygar. Geometric leaf classification. *Computer Vision and Image Understanding*, 133:102–109, 2015.
- [5] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, and Joao 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.
- [6] David G. Lowe. Object recognition from local scale-invariant features. In *Proc. of the International Conference on Computer Vision, Corfu*, 1999.

- [7] 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.
- [8] 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.
- [9] Yu Wang, Xiaojuan Ban, Jie Chen, Bo Hu, and Xing Yang. License plate recognition. *Optik*, (126):2895–2901, 2015.
- [10] 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.