Automatic Plant Classification by Leaf Recognition System

Ricardo Aaron Pillay

University of KwaZulu-Natal, South Africa
218009114@stu.ukzn.ac.za

Abstract. Plants have a wide range of application in agricultural fields. There have been multiple attempts to classify leaves using different types of features with the incorporation of machine learning. In this study we propose a plant leaf classification system that uses the shape features of a leaf to classify which species of plant the leaf belongs to, using Random Forest Classification (RF). Several different methods for Image Pre-processing, Segmentation, Feature Extraction and Classification of plant leaves are also explored and contrasted in this study. The proposed system has an accuracy rate of 80 percent.

Keywords: Machine Learning · Plant Leaf Classification System · Shape Features · Random Forest Classification ·

1 Introduction

Classification of a plant species is crucial to the agricultural sector, where farmers need to estimate crop yields, based on species, identify weeds from crop species, etc. The recent urbanization and biodiversity loss has made plant identification an important problem for other professionals such as foresters, environmental conservationists, botany scientists and agronomists.

Plant classification is conventionally done by their floral parts, fruits and leaves by botanist [7]. Selection of flowers and fruits is not a suitable choice of selection for plant identification as they do not last long. Leaves on the other hand are available for longer duration and are available in abundance and thus are a suitable choice for automatic classification of plants. Considering the huge amount of species, plant identification is a fairly difficult task even for botanists. Discrimination of different plant species is based on their unique features. Plant leaves contain several discriminative and unique geometric and vein features, based on the species. The idea that plants can be classified based on their unique leaf features has spawned curiosity in the minds of researchers to devise machine based plant identification. In literature different shape or geometric leaf features are used for classification like aspect ratio, circularity, eccentricity, roundness and others [2]. Furthermore vein features [3] and colour feature moments such as mean, standard deviation, skewness, and kurtosis [4] are also used for classifying plant leaves.

The organization of the rest of this paper is as follows. Section 2 provides a literature survey of contemporary research in the area of leaf-based plant identification. Section 3 provides the methods used in this investigation for each of the image processing steps: pre-processing, segmentation, feature extraction and classification as well as empirical experiment comparisons between different methods that were considered for each step. Section 4 shows and discusses the results of this study. Section 5 is the conclusion and will also provide an explanation on future work and improvements that could be made to this system.

2 Literature Review

Liu et al. [1] proposed an automated classification of stems and leaves of potted plants based on point cloud data. A specialised scanner was used to extract the point cloud data representation of images of potted plants. The 3D convex hull algorithm was used to extract leaf feature points, while the point density of a 2D projection was used to extract stem feature points. The Support Machine Vector (SVM) was used for classification of the potted plants. This study tested the point cloud data method against random selection and artificial selection methods, on 3 potted plants and the result was that the point cloud method performed more accurately and efficiently with an overall average Kappa Coefficient of 0.68. The point cloud representation captures distinct parts of a plant that regular segmentation methods like thresholding or edge detection is not able to. However, this method is not robust, requires very complex processing and good accuracy for different plants is not guaranteed.

Suniara [2] suggested a plant classifier based on leaf shape and texture features. Segmented(Thesholded) images of leaves were used to extract both shape and texture features. 3 classification techniques, namely Probabilistic Neural Network (PNN), k Nearest Neighbour (k-NN) and Support Machine Vector (SVM) were contrasted on classifying both shape and texture features. PNN had the highest accuracy (88.6 percent) for classifying shape features while SVM had the highest accuracy (79.33 percent average) for classifying texture features. The PNN shape based classifier gave better precision, had a faster preparing rate and a simple architecture. The SVM texture based classifier also had a fast performance rate, simple structure and high accuracy (but not as high as PNN shape classifier). The division/segmentation method used in this study resulted in the production of false regions of interest that affected performance of the models.

Sunny [3] carried out a review on deep learning for plant species classification using leaf veins. The images were segmented using Sobel edge detection and skeletonization, then 62 vein features were extracted using the CapsNet convoluted neural network (CNN). The images were then classified using the Probablistic Neural Network (PNN) approach. Since CapsNet has been proven to be an effective CNN, it ensures high accuracy regardless of input test data. This review concluded that CapsNet can provide very suitable and standardized vein feature extraction of leaves, that allows for effective classification of plants.

El-hariri et al [4] undertook an investigation on their implemented plant classification system based on leaf features. They used features relating to Hue, Saturation, Value (HSV) colour moments, veins, shapes and texture features (first order texture and Gray Level Co-occurence Matrix), to classify leaf images. Random Forests (RF) and Linear Discriminant Analysis (LDA) were used to classify the plants, and their results were contrasted. An issue regarding this study is high computational complexity as pre-processing, segmentation and feature extraction must be performed on both the colour and grey level images. A major advantage of using both colour and grey level features is the high accuracy rates. LDA produced a higher classification accuracy of 92.65 percent, while RF produced an accuracy of 88.82 percent.

Bhardwaj, Kaur and Kunar [5] suggested that plants could be recognised by their leaves using Moment Invariant and Texture Analysis. The texture representation of the images were obtained by using frequency domain processing (Fourier Transform) and the shapes of the images were obtained using thesholding. Morphological opening was used to segment the leaves and morphological closing was used to segment the stem. Both texture (GLCM features) and shape features (including Hu's 7 invariant moments) were extracted and comparatively used to classify the images, using the k-Nearest Neighbour method. The shape features produced higher accuracy of 91.5 percent. Suggested improvements include use of machine learning to further improve accuracy.

Lagerwell [6] proposes a combination of morphological feature analysis and Hu's invariant moments for the classification of plant leaves using the Euclidean distance measure. The author compares classification methodologies based on PCA, Shape and a combination of the 2. The pre-processing, for both PCA and Shape methods, involve cropping and resizing. 38 morphological features are extracted for the shape method, while eigen pictures are extracted for the PCA method. Distance Euclidean measure is used for classification in the Shape method and Distance Cosine measure is used for classification in the PCA method. The PCA method is more computationally efficient, however it requires more memory and is not as accurate (85.4 percent) as the Shape method. The Shape method is not as computationally efficient as the PCA method and has to have it's feature vectors normalized before classification, but it uses less memory and has a higher accuracy of 91.9 percent. The committee of the 2 methods give an accuracy of 89.2 percent but it is slower and uses more memory.

3 Methodology

3.1 Leaf Image Acquisition

The leaf snap dataset [9] was used for training, validating and testing the proposed leaf classifier system. This data set comprises of all 185 tree species from the Northeastern United States. 10 species were chosen, namely: abies concolor ,abies nordmanniana ,acer campestre ,acer ginnala ,acer negundo ,acer palmatum ,acer pensylvanicum ,carya tomentosa ,carya ovata and pinus bungeana. 25 images were used per species, resulting in a total of 250 images. This dataset was separated according to the ratio 60:20:20 for training, validating and testing of the system.

3.2 Leaf Image Preprocessing and Enhancement

The aim of pre-processing leaf images is: to convert the colour image to gray scale, ensure all images are of the same size, to remove background objects in the image and to prevent uneven lighting from affecting segmentation.

Image Resizing: The images from the different species of leaves from the Leafsnap Dataset [9] were of various sizes. To ensure all images were processed in a standardized manner, every leaf image used was resized to a length and width of 300 by 300. This was performed using Python's OpenCV *resize* function. The images had to be resized into squares because a rectangular resized image can distort or stretch the image. A square resized image is shown in Figure 1 and a rectangular resized image is shown in Figure 2.

Fig. 1. Resized 300 by 300 image of abies concolor leaf

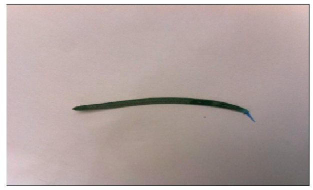

Fig. 2. Resized 500 by 300 image of abies concolor leaf

Converting to Grey Scale: The resized images were then converted into grey scale using Python's OpenCV *cvtColor* function. This was a necessary step as the next preprocessing step (mean filtering) and the method for segmentation (thresholding), used in the investigation, only works with grey scale images. Figure 3 shows the converted grey scale image of the resized image (as seen in Figure 1 above).

Fig. 3. Grey scale image of abies concolor leaf

Mean Filtering: Mean filtering, using Python's OpenCV *blur* method, with a 7 by 7 kernal, was then applied to the grey scale images in order to remove background noise to allow only the leaf in the image to be highlighted. This also prevented issues with segmentation later on due to uneven light distribution. Median filtering, using Python's OpenCV *medianBlur* method, with an 7 by 7 kernal, was also considered, however when used, it was observed that the median filter truncates narrow parts of the leaves, which can affect the accurate extraction of a leaf's features. Figure 4 shows the mean filtered image of Figure 3 above, and Figure 5 shows the median filtered image of Figure 3 above.

Fig. 4. 7 by 7 kernal Mean filtered image of abies concolor leaf

Fig. 5. 7 by 7 kernal Median filtered image of abies concolor leaf

3.3 Leaf Segmentation

The images had to be segmented in order to identify the region of interest (in this case the leaf). Obtaining the correct region of interest was discovered to be extremely vital when it came to extraction of features. The main goal was to find a way to fully separate the leaf from the background.

Inverse Binary Thesholding: The pre-processed images were converted to binary images, where the leaf was white, separated from the black background. This was done by using Python's OpenCV *threshold* method, to implement inverse binary thresholding. It was found that instead of choosing a set threshold value for all images, segmentation occurred best when using a different threshold value for each image. This threshold value T was calculated as follows:

$$T = \frac{MeanIntensity(image)}{2}$$

Segmentation by edge detection using the Laplacian filter, was also attempted. This however, this did not fully separate the leaf from the background and highlighted unnecessary details in the images. The thresholded image of Figure 4 above, is shown in Figure 6 and the Laplician filtered image of Figure 4 above, is shown in Figure 7 below.

Fig. 6. Inverse Binary Thresholded image of abies concolor leaf

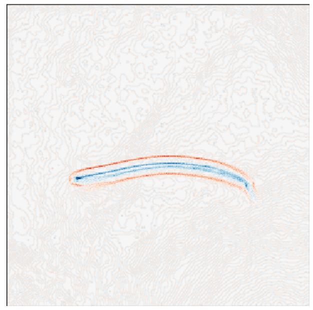

Fig. 7. Edge detection by Laplacian Filter on abies concolor leaf image

Isolating the Region of Interest: From the binary images, Python's skimage library method, *label*, was used to actually isolate the region of interest (leaf). This method allowed for the labelling of regions of interest for feature extraction using skimages's method *regionpropstable*. Figure 8 below shows the labelled image of Figure 6 above.

Fig. 8. Labelled Region of Interest in image of abies concolor leaf

3.4 Leaf Feature Extraction

From the literature studied, and explicitly mentioned in [5], region/shape based features give the most accurate classification, so the aim is to extract the necessary region/shape features. As mentioned above, Python's skimage library method ,*regionpropstable* , was used to extract 18 region/shape based features from the image's Region of Interest (ROI). The features extracted are as follows:

- Area: number of pixels in the ROI

- Perimeter: length of the outside boundary of the ROI

- Perimeter-Area Ratio: the ratio of the perimeter to the area of the ROI.

$$PerimeterAreaRatio = \frac{Perimeter}{Area}$$

- Circularity: a circularity value of 1 indicates a perfect circle, a value approaching 0 indicates an elongated shape.

$$Circularity = 4\pi\ (\frac{Area}{Permeter^2})$$

- Convex Hull Area: number of pixels in the convex hull of the ROI. The convex hull of a set S (which contains the intensity values of the image) is the smallest convex polygon for which each point in S is either on the boundary of P or it's interior.

- Solidarity: the ratio between the area of the leaf and the area of the convex hull of the leaf.

$$Solidarity = \frac{Area}{ConvexArea}$$

- Major Axis Length: the longest length from 1 end of the leaf to the other end, on the horizontal plane.

- Minor Axis Length: the longest length from 1 end of the leaf to the other end, on the vertical plane.

- Euler Number: Number of connected components subtracted by the number of holes. If C is the number of connected components and H is the number of holes, then

$$EulerNumber = C - H$$

- Maximum Feret's Diameter: The longest distance between points around the ROI's convex hull contour.

- Inetia Tensor Eigenvalue: Measure of the covariance of the image intensity along the image ROI axis. It serves as a measure of the elongation of a leaf.

- Hu's 7 invariant moments: the Hu set of image moments is a set of seven moments that remains invariant under changes in translation, scale and rotation. It was imperative to include these features as multiple images, for a single species, in the leaf snap dataset were translated, scaled and rotated in different ways. The features are calculated as follows:

$$I_1 = n_{20} + n_{02}$$

$$I_2 = (n_{20} - n_{02})^2 + (2n_{11})^2$$

$$I_3 = (n_{30} - 3n_{12})^2 + (3n_{21} - n_{03})^2$$

$$I_4 = (n_{30} + n_{12})^2 + (n_{21} + n_{03})^2$$

$$I_5 = (n_{30} - 3n_{12})(n_{30} + n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2] +$$

$$(3n_{21} - n_{03})(n_{21} + n_{03})[3(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2]$$

$$I_6 = (n_{20} - n_{02})[(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2] + 4n_{11}(n_{30} + n_{12})(n_{21} + n_{03})$$

$$I_7 = (3n_{21} - n_{03})(n_{30} + n_{12})[(n_{30} + n_{12})^2 - 3(n_{21} + n_{03})^2] -$$

$$(n_{30} - 3n_{12})(n_{21} + n_{03})[3(n_{30} + n_{12})^2 - (n_{21} + n_{03})^2]$$

3.5 Leaf Classification

The feature vectors extracted, along with their labels, were given to a model to train, in order to learn how to classify the leaves of the 20 different species. A classification report based on how the training set performed on the validation set was displayed after training to measure the accuracy of training. The classification report was obtained using Python's sklean method *classification_report*.

Random Forest Classifier: Python's sklearn method *RandomForestClassifier* was used to implement a Random Forest classifier with 50 trees and a maximum tree depth of 20. The maximum tree depth was restricted to prevent the trees from over-fitting. Once trained on the training data, it produced 96.91 percent training accuracy when tested on the validation set. A k-NN neighbour classifier was also implemented using Python's sklearn method *KNeighborsClassifier*, but when tested on the validation set it only produced a 95.88 percent training accuracy. The Random Forest classifier was also selected due to the fact that, unlike k-NN, it is does not need the feature vectors to be normalized prior to training or testing. Figure 9 shows the training report for the RF classifier and Figure 10 shows the training report for the k-NN classifier.

– Precision is the ability of a classifier not to label a negative result as a positive one. If t_p is the number of true positives and f_p is the number if false positives, then:

$$Precision = \frac{t_p}{t_p + f_p}$$

	precision	recall	f1-score
abies_concolor	1.00	1.00	1.00
abies_nordmanniana	1.00	1.00	1.00
acer_campestre	1.00	1.00	1.00
acer_ginnala	1.00	1.00	1.00
acer_negundo	1.00	1.00	1.00
acer_palmatum	0.88	1.00	0.93
acer_pensylvanicum	0.90	1.00	0.95
carya_ovata	1.00	0.96	0.98
carya_tomentosa	0.92	1.00	0.96
pinus_bungeana	1.00	0.78	0.88
accuracy			0.97
macro avg	0.97	0.97	0.97
weighted avg	0.97	0.97	0.97

Training Accuracy: 96.91%

Fig. 9. Random Forest classifier training report on validation set

	precision	recall	f1-score
abies_concolor	1.00	1.00	1.00
abies_nordmanniana	1.00	0.67	0.80
acer_campestre	1.00	1.00	1.00
acer_ginnala	1.00	1.00	1.00
acer_negundo	1.00	1.00	1.00
acer_palmatum	0.88	1.00	0.93
acer_pensylvanicum	0.90	1.00	0.95
carya_ovata	1.00	0.96	0.98
carya_tomentosa	0.92	1.00	0.96
pinus_bungeana	0.86	1.00	0.92
accuracy			0.96
macro avg	0.95	0.96	0.95
weighted avg	0.96	0.96	0.96

Training Accuracy: 95.88%

Fig. 10. k Nearest Neighbor classifier training report on validation set

– Recall is the ability of the classifier to find all positive samples. If t_p is the number of true positives and f_n is the number if false negatives, then:

$$Recall = \frac{t_p}{t_p + f_n}$$

– F1 score is the mean of precision and accuracy:

$$F1score = \frac{(Precision + Recall)}{2}$$

4 Results and Discussions

As illustrated in Figure 9 above, the proposed classifier performs accurately on each species in the validation set of seen data. However, this is not the true representation of the accuracy of the classifier. Unknown leaf images samples in the test set must be tested using the classifier to truly determine how the effectiveness of this classifier in the real world.

The classification results reflected that 40 out of 50 test leaf images were classified correctly, resulting in an overall testing accuracy of 80 percent. Every species had 5 test leaf images each. Table 1 below shows the confusion matrix for the test set. Note that species names were abbreviated as follows:
ab con = abies concolor
ab no = abies nordmanniana
ac cam = acer campestre
ac gin = acer ginnala
ac neg = acer negundo
ac pal = acer palmatum
ac pen = acer pensylvanicum
car ova = carya ovata
car tom = carya tomentosa
pin bun = pinus bungeana

The testing accuracy obtained (80 percent) is lower than the training accuracy (96.91 percent) as expected, however the testing accuracy is still fairly high, indicating that the proposed plant classifier was not over-fitted to the training data. Furthermore, the classifier was able to correctly identify all leaves in 4 out of the 10 species. It is also important to note that there was no species that had 0 correctly classified leaves - the least number of correctly identified leaves was 3.

	ab con	ab no	ac cam	ac gin	ac neg	ac pal	ac pen	car ova	car tom	pin bun
ab con	3	0	1	0	0	0	0	0	1	0
ab no	0	5	0	0	0	0	0	0	0	0
ac cam	0	0	5	0	0	0	0	0	0	0
ac gin	0	0	0	4	1	0	0	0	0	0
ac neg	0	0	0	1	3	0	1	0	0	0
ac pal	0	0	0	0	0	5	0	0	0	0
ac pen	0	0	1	0	0	0	4	0	0	0
car ova	0	0	1	0	1	0	0	3	0	0
car tom	0	0	0	0	1	1	0	0	3	0
pin bun	0	0	0	0	0	0	0	0	0	5

Table 1. Confusion matrix for Test set

The accuracy achieved by this system is lower than the accuracy achieved by similar leaf classifiers in the studied literature. An important example of this is the study in [4], which also uses Random Forests but has an average accuracy of 88.82 percent. This could be due to the fact that more images were used for training the RF model in [4] and more types of features were used for classification. Another imperative comparison can be made between this study and the research conducted in [5]. The study in [5] has a 91.5 percent accuracy with k-NN classification, using shape based features. From the emperical tests done (as shown above in *Leaf Classification*), it was seen that k-NN is considered a worse classification method as compared to RF for classification of leaves, yet [5] has proven that k-NN produces a higher accuracy. A possible reason for this discrepancy in accuracy could be due to the use of the frequency domain processing in [5] for pre-processing and segmentation, which may highlight regions better for feature extraction.

5 Conclusion

In this work, multiple methods for pre-processing, segmentation and classification of plant leaves were compared against each other in empirical experiments. The most suitable methods were selected in each case to maximise performance and accuracy of the final classifier. For pre-processing, mean filtering was selected over median filtering, for segmentation, thresholding was selected over edge detection (by use of Laplacian filter), for feature extraction, only region/shape features were extracted and for classification, Random Forests was selected over k-Nearest Neighbor classification. The overall accuracy of the system was 80 percent on the test set.

Future work can involve the use of convoluted neural networks for automatic segmentation and feature extraction. This could allow for obtaining more relevant regions of interest and for extraction of more relevant features in order to possibly improve accuracy.

References

[1] Z. Liu, Q. Zhang, P. Wang, Z. Li, H. Wang, "Automated Classification of stems and leaves of potted plants based on point cloud data," 2002.

[2] G. Suniara, "Plant Classifier based on Leaf Shape and Texture features," M.S. Thesis, Dept. Eng., Tharpar University, Patiala, India, 2017.

[3] N. M. Sunny, "A Review on Deep Learning for Plant Species Classification using Leaf Vein," *International Journal of Engineering Research & Technology (IJERT)*, vol. 9, Jun 2020.

[4] E. Elhariri, N. El-Bendary, A. E. Hassanien, "Plant Classification System based on Leaf Features," *DOI: 10.1109/ICCES.2014.7030971*, 2014.

[5] A. Bhardwaj, M. Kaur and A. Kumar, "Recognition of Plants by Leaf Image using Moment Invariant and Texture Analysis," *International Journal of Innovation and Applied Studies* [Online], vol. 3, pp. 237-248, Available: http://www.issr-journals.org/ijias/

[6] R. Lagerwall, "Plant Classifcation Using Leaf Recognition," B.S, School of Computer Science, University of KwaZulu-Natal, KwaZulu-Natal, South Africa, 2011.

[7] P.K. Mishra, et al., "A semi automatic plant identification based on digital leaf and flower images," in *Advances in Engineering, Science and Management (ICAESM)", 2012 International Conference* , 2012.

[8] P. Salve, et al., "Identification of the Plants Based on Leaf Shape Descriptors," in *Proceedings of the Second International Conference on Computer and Communication Technologies*, Springer, 2016.

[9] Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, David W. Jacobs, W. John Kress, Ida C. Lopez, João V. B. Soares, "Leafsnap: A Computer Vision System for Automatic Plant Species Identification," *Proceedings of the 12th European Conference on Computer Vision (ECCV)*, October 2012.