TreeID: An Image Recognition System for Plant Species Identification

Manu Chopra
Department Of Computer Science
Stanford University
mchopra@stanford.edu

1. ABSTRACT

Plant identification is the building brick of plant research and development, and is very important for environmental protection and exploration. Usually, the leaves can be easily obtained from a plant and have sufficient visible characteristics for differentiating between their respective plant species. Plant identification is a huge problem that has escaped into neglect for years. Without visual recognition tools, users currently have to manually navigate through a dichotomous key.

Sample Questions From A Dichotomous Identification Key
Are the leaflets at least twice as long as they are wide?
Are the leaves palmately veined or pinnately veined?
Are the leaflets sessile?
Do the leaves show translucent dots when held against light at a certain angle?
Are the leaves fat or thin?
Are the leaves curved a lot or just a little?

Identifying a single species usually involves answering vague questions, such as, "Are the leaves flat or thin?" or "Are the leaflets at least twice as long as they are wide?". The questions usually involve obscure technical terms like "Are the leaflets sessile?" or require some sort of mathematical intuition like "Are the leaflets wider than 0.75 inches?"[1]. On average, to identify a single plant species, You have to answer at least a dozen such questions. This process can be difficult and frustrating for amateurs, and experts.

It is then easy to see how an accurate image recognition system can dramatically transform the world of plant species identification. More over, these questions are fundamentally simpler for computer algorithms to solve. It's much easier for an image recognition system to find out if a leaflet is at least twice as long as it's wide or if it's wider than 0.75 inches, than it's for a human to perform the same tasks.

Stanford has an awe inspiring collection of trees from across the globe. From trees older than five centuries to one of the only clones of the famous Newtons apple tree in the world, Every tree on campus has a story to tell. We are lucky to have trees from almost every country in the world. There are more than 60,000 trees and 800 different plant species on the Stanford campus. Unfortunately, without any simple method of identifying trees, There is very little interest among students to explore trees on campus.

In this paper, I have attempted to show how we can use computer vision to make the problem of plant species identification simpler and more intuitive. I hope that such a system which reignite the lost interest in the field of plant species identification.

2. INTRODUCTION

TreeID is an automatic leaf image recognition system for plant species identification. The aim of the project is to classify all of the 60,000 trees on campus. The dataset was obtained from several different sources including self-conducted tree walks . The TreeID model contains the following steps:

2.1 Classification

To make the recognition process easier, we require the user to take the image of the leaf against a light, non-textured background. If an image features a single leaf against a white, non-textured background, We classify it as a valid image. In the figure below, the image on the left is rejected. This decreases the computational load on our system as images that fail this classification are not processed further. This process is occasionally used by consumer-focused image recognition systems [2]. The classification is achieved by applying gist features on an image and feeding the obtained features into a Support Vector Machine.

Figure 2: The image on the left is not accepted while the image on the right is accepted.

2.2 Segmentation

Reliable leaf segmentation is critical in order to obtain an accurate leaf shape description. After capturing and validating the leaf image, the leaf is separated from its background by using color based segmentation. In [4], authors propose Hybrid Image Segmentation Algorithm for leaf recognition and extracting features such as leaf veins, texture and curvature. However, features such as the color of the leaf or venation patterns are not be the most suitable features they are either too highly variable across different specimens of the same plant species or are usually undetectable due to poor image quality. Previous researchers have had the liberty of using these features as they can assume that the system is given a high definition image from where these features can be extracted. TreeID is designed to be a consumer facing application and most users don't have access to high definition cameras and will rely mostly on their mobile phone cameras.These images hence may contain varying amounts of blur, noise, illumination patterns, shadows among other possible defects.

2.3 Training and Testing the CNN

I used a simple three-layer ConvNet with the architecture as (conv-relu-pool) x 2 -affine-softmax which gave me surprisingly accurate results. [See Conclusion] I also attempted to use a Growing Convolutional Neural Network, in which the network grows up itself until it solves the target problem thereby, achieving the best trade-off between classification accuracy and computation cost.

3. RELATED WORK

The problem of plant species identification has attracted a lot of interest from prior researchers. Leafsnap, a series of electronic field guides was developed by researchers from Columbia University, the University of Maryland, and the Smithsonian Institution.[2] Leafsnap achieves a high rate of accuracy using a simple KNN based classifier which is trained on high quality images taken in a lab. The Leafsnap database covers all 185 tree species from the Northeastern United States including 23,147 Lab images of pressed leaves. Trees.stanford.edu is a SULAIR and Stanford Historical Society digital initiative to map every single tree on campus and relies heavily on Ron Bracewell's legendary but slightly outdated book, "Trees of Stanford and Environs." [8] Stanford's Bio29N class [Party With The Trees] maintains an iNaturalist project map of tree observations around campus and has over 130 reliable observations of plant species on campus.

Zhong-Qiu Zhao et al. [6] attempted to use Growing convolutional neural networks for plant species identification to exceptional results.Niko Sunderhauf et al. [10] used convolutional neural networks for fine grained plant classification. Most of these projects (except Leafsnap) are not meant to be used by a consumer directly and hence avoid several key user facing optimizations and use methods which don't necessarily work well for TreeID. Jyotismita Chaki et al.[3] use shape based features for plant leaf recognition. The previously mentioned Leafsnap project uses color based segmentation, a technique I employ in TreeID to get shape based features [3].

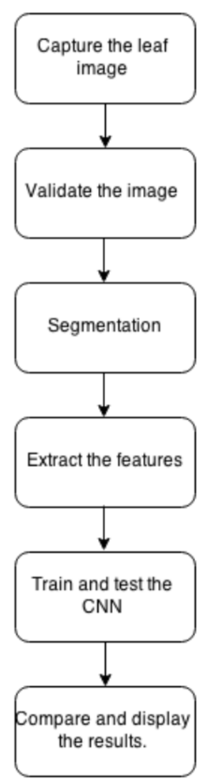


Figure 3: A flowchart of the TreeID model.

4. DATABASE

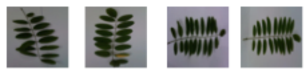
Plant Species	Sample Pictures from the database
Gingko Biloba	
Robinia Pseudo-acacia	
Quercus Agrifolia (Coast Live Oak)	
Acer Campestre	
Chamaecyparis Pisifera	

Figure 4: Sample pictures from TreeID database

One of the most challenging parts of this project was to collect the relevant data. I founded existing databases of leaf images to be helpful but lacking in several prospects. Notably albeit expectedly, there is no database of Stanford specific leaf images. Instead, I created my own data set by conducting tree walks to map common trees on campus. These images contain varying amounts of blur, noise, illumination patterns, shadows among other things to mimic images taken by users in the real life and were all taken from my phone. I also relied on existing databases like the previously mentioned LeafSnap database which covers all 185 tree species from the Northeastern United States.

For the scale of the project for this class, I collected images of the 50 most common plant species on campus. The list of the most common trees on campus was also obtained from Ron Bracewells book,"Trees of Stanford and Environs" [8]. Combined, these species account for at least 80% of trees on campus. Notably, Quercus Agrifolia or the Coast Live Oak alone accounts for more than 60% of trees on campus.

For every single plant species, I collected around 40-50 square images before data augmentation. All the collected images were shot against a light, non-textured background to enhance segmentation results and were reshaped to 200 by 200 pixels before undergoing segmentation and feature extraction. **The collected database is the largest collection of Stanford specific tree leaves that I am aware of.**

5. APPROACH

(a) (b) (c)

Figure 5: A leaf image (a) in original RGB, (b) converted to SV space, and (c) segmented using EM

TreeID uses leaf shape as the sole recognition cue. As explained earlier, Other features such as the color of the leaf, venation pattern or leaf texture are not suitable for various reasons they are either too highly variable across different leaves of the same species, undetectable due to poor image quality or highly seasonal.

Reliable leaf segmentation is thus critical in order to obtain an accurate leaf shape description. Color-based segmentation has been used successfully in previous leaf recognition applications and has several advantages compared to other approaches. Leaves vary greatly in shape. Some species of leaves are compound (consisting of small leaflets) while others are found grouped into clusters. This gives rise to complex segmentation boundaries that are difficult to handle for edge-detection algorithms. A color-based approach works much better by not making any assumptions about color distributions or leaf shape.

We start by converting the image from RGB space to HSV space. In [2], the authors showed that both the saturation and value of the HSV space are consistently useful to distinguish leaf pixels from the background. Hue is not found to be as useful because the background often has a greenish tinge due to reflections from the leaf or surrounding foliage.

We segment images by estimating foreground and background color distributions and using these to independently classify each pixel. This initial segmentation is solved using Expectation-Maximization based on a Gaussian model.This initial segmentation, solved using Expectation-Maximization, is then processed to remove false positive regions.

5.1 Segmentation Using Expectation Maximization

Neeraj Kumar et al. run initial segmentation using EM and achieve highly accurate segmentation results.

TreeID uses the same model. The probability distribution of a pixel x, represented by its saturation and value, is modeled as the sum of two Gaussians.

$$p(x|\theta) = \sum_{k=1}^{2} \frac{1}{2} p(x|\mu_k, \sum)$$

where each $p(x|\mu_k, \sum)$ is a Gaussian with a mean μ_k and a common covariance of $\sum$. Each Gaussian is assigned an equal weight of $\frac{1}{2}$.The set of models is represented by

$$\theta = \left\{ \mu_1, \mu_2, \sum \right\}$$

We initialize each of the two Gaussians near the center of their respective distributions, so that they converge to the corresponding clusters when provided with a new image.

The covariance matrix ($\sum$) is set to a value near the expected final values so that it can quickly converge. The fact that the covariance matrix is shared between the two Gaussians brings a significant speed advantage. In a standard two class case, if we denote the label of pixel x as $z \in \{1, 2\}$, then

$$p(z = 1|x) = \frac{1}{1+e^{\beta_0 + \beta^T x}}$$

After computing the value of $p(z = 1|x)$, we can quickly calculate the value of $p(z = 2|x)$ as

$$p(z = 2|x) = 1 - p(z = 1|x)$$

thus taking a linear logistic form.The segmentation is done via EM, by alternating between estimating probabilities of each pixel using the current parameters, and updating the parameters using current pixel probabilities. [2]

A very common type of false positive region can appear at the outer border of the image. Most users will place the leaf on a white sheet of paper when taking the picture. It is common to find that some parts of the image border lay outside the piece of paper, giving rise to falsely detected regions. [See the 2nd image in 2nd row in Figure 6] To remove these false positive regions, We first compute connected components on the segmented image. Any connected component that has a large boundary on the image border (relative to its area) is then excluded, thus eliminating any false positive regions.

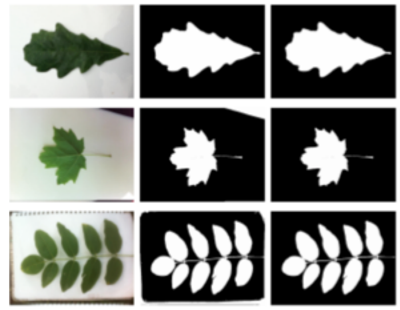


Figure 6 : A leaf image (a) in original RGB, (b) segmented using EM (c) removing false positive regions

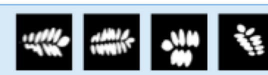
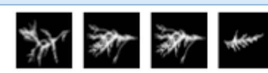
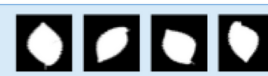
Plant Species	Sample Segmented Pictures from the database
Gingko Biloba	
Morus Alba	
Robinia Pseudo acacia	
Chamaecyparis Pisifera	
Ulmus Americana	

Figure 7: Sample segmented images from TreeID database

6. EXPERIMENT

We obtain segmented images for all the 50 classes. All the images are 200 by 200 pixels and have been weeded out for false positive zones. All images are zero-meaned and transposed so that the channels come first. These images are then feeded into a three-layer ConvNet with an architecture of (conv-relu-pool) x 2 -affine-softmax. Every class has 40 - 50 segmented images of a unique plant species prior to data augmentation. Two separate validation sets are also created.

The first validation set has 10 images each for every plant species (or class) thereby containing 500 (50 x 10) validation set images. The second validation set has a different number of images for every plant species depending on their popularity on campus. What this means is that 60 percent of images in the second validation set belong to a single plant species, namely

Quercus Agrifolia or the Coast Live Oak. Each plant species is represented by its popularity on campus in the second validation set. While the first validation set accuracy is a better representation of our model's accuracy, the second validation set accuracy is more useful in the real world.

	1st Validation Set	2nd Validation Set
Accuracy	0.375	**0.750**

Table 1: Results On First and Second Validation Set

Running the model on the first validation set gives us a decent accuracy of 37.5% compared to the random accuracy for 50 classes of 2%. Running the same model on the first validation set with non-segmented images gives a comparatively poor accuracy of 10%. I believe this is largely because of blur, shadows and other background defects in the original image. This seems to confirm that our decision to segment all of the input images was a good one.

	Non-Segmented	Segmented
Accuracy	0.10	0.375

Table 2: Results of Segmented Non Segmented Images on First Validation Set

What is interesting is just how well the model performs on the second validation set achieving around 75% accuracy. I believe this is largely because of the unique shape of a Coast Live Oak leaf compared to other leaves in the database. The leaves are dark green, oval, often convex in shape, 27 cm long and 14 cm broad; the leaf margin is spiny-toothed , with sharp fibers that extend from the lateral leaf veins. The only leaf with a shape similar to the Coast Live Oak leaf is the leaf of the Canyon Live Oak (Quercus chrysolepis). The two species are considered to be sympatric as they usually exist in the same geographic area and the two may be hard to distinguish because their spinose leaves are superficially similar. This is a potential problem but fortunately, Canyon Live Oak is not a very common tree at Stanford and mostly occurs at the Jasper Ridge area. The Canyon Live Oak also doesn't appear on the list of the 50 most common trees on campus. To be sure, I conducted a separate experiment by inserting 10 images of a Canyon Live Oak leaf into the second validation set. The experiment still achieved around 70% accuracy thereby confirming that that our TreeID model should work well even with leaves of sympatric species.

Curvature is a fundamental property of shape. Researchers in the past have used curvature to successfully differentiate between leaves. Taking inspiration, I computed the histograms of curvature values at each scale and concatenate these histograms together to form a Histograms of Curvature over Scale (HoCS) feature. A similar approach was used by researchers in [2]. Histograms have the benefit of being simple to represent, compact, and fast to compare using metrics such as L_1 or L_2. However, I received the same accuracy rates as without the curvature in the first place. This makes sense as most plant species in the data set don't have curved leaves. Even the convex shaped oak leaves have sufficiently distinctive shapes which makes the curvature feature pretty much useless. I do not believe that this would be true for all leaves on campus. Hence, A curvature feature may be useful in a large scale model but serves no greater purpose in the current model.

7. CONCLUSION

The results from TreeID have reaffirmed my belief in using Convolutional Neural Networks for plant species identification. The intersection of CNNs and Plant Biology is ripe for exploration and I believe that TreeID is a great start. The project clearly had several limitations to begin with. I spent most of my time building a relevant database and even then, I didn't have access to lab pressed leaf images that Leafsnap had. I certainly learnt a lot during the course of building TreeID. I plan to make the TreeID database available for free on line. I have also used TreeID in the wild and it performs remarkably well. The second validation set rewarded TreeID with 75% accuracy and the real world accuracy is certainly close to that.

Due to the lack of time, I couldn't completely implement a growing convolutional neural network which was proposed by researchers in [5]. Zhong-Qiu Zhao et al. achieved remarkable accuracy in plant species identification by using a growing CNN and I would like to finish my implementation of the same in the future. Post the completion of such a TreeID model, I plan to build a mobile application which allows any person on campus to identify a tree by clicking a picture of its leaf. Such an application will almost certainly use the phones GPS location and existing Stanford tree maps to provide more accurate results. I believe that such an application will make the process of plant species identification much simpler and intuitive.Leafsnap achieved such a model for North-Eastern United States and I want to achieve the same for Stanford and the rest of California. Every tree on campus has an amazing story to tell. Imagine a world where anyone anywhere irrespective of their prior knowledge can take a picture of the tree in front of them and know all about it. With TreeID, I hope to make this dream a reality.

8. ACKNOWLEDGEMENTS

Special thanks to Professor Devaki Bhaya from the Department of Plant Biology at the Carnegie Institution for Science for helping me in every way possible. Thanks also to Kelly McManus, the head TA for BIO 29N for inspiring me to work on this project. Finally thanks to Andrej Karpathy and Professor Li for building such an amazing class and without whom, I wouldn't have been introduced to this wonderful material.

9. REFERENCES

1. Matt Ritter, 2011, Californian's Guide to the Trees among Us. Published by Heyday. 192 pages. 2. Neeraj Kumar, Peter N. Belhumeur, Arijit Biswas, "Leafsnap: A Computer Vision System for Automatic Plant Species Identification,"
3. Plant Leaf Recognition using Shape based Features and Neural Network classifiers : Jyotismita Chaki, Ranjan Parekh
4. Hybrid Image Segmentation Algorithm for Leaf Recognition and Characterization : N.Valliammal, Dr.S.N.Geethalakshmi
5. Neural Network Application on Foliage Plant Identification : A. Kadir,L. E. Nugroho, A. Susanto ,P.I. Santosa
6. Plant Leaf Identification via A Growing Convolution Neural Network with Progressive Sample Learning : Zhong-Qiu Zhao, Bao-Jian Xie Yiu-ming Cheung,Xindong Wu
7. iNaturalist Bio29N: Party with Trees (Available online at http://www.inaturalist.org/projects/bio29n-party-with-trees)
8. Trees of Stanford and Environs: Ronald Newbold Bracewell
9. M-K Hu, Visual pattern recognition by moment invariants, Transactions on Information Theory, 1962, pp. 179-187.
10. Fine-Grained Plant Classification Using Convolutional Neural Networks for Feature Extraction : Niko Sunderhauf, Chris McCool, Ben Upcroft, and Tristan Perez