MLBD / Lab 2 / Feature extraction

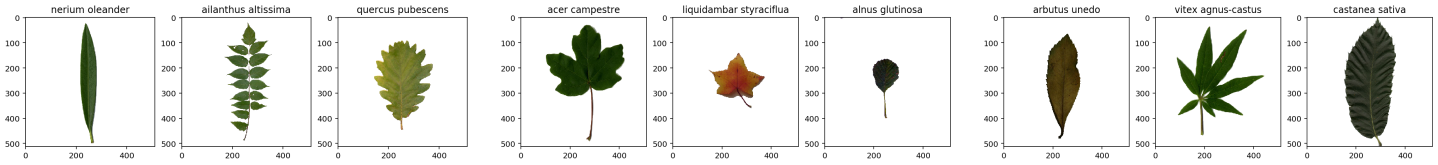
Authors: Romain Claret & Edward Ransome

Introduction

Plants classification is a very complex and time-consuming process. The usual way to classify the species biodiversity is by their leaves characteristics. However, even within the same species, leaves are not the same; indeed, the plant maturity and the current season make alterations.

However, the plant shape is less altered and is the most relevant for plant classification; it is, for this reason, the most used trait to classify plants is by discriminating based on measurements.

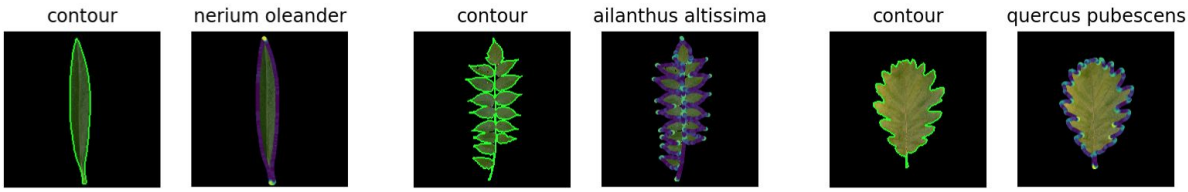
In this laboratory, we will be exploring machine learning approaches to identify leaves automatically, and hopefully, do better than humans, and much faster.

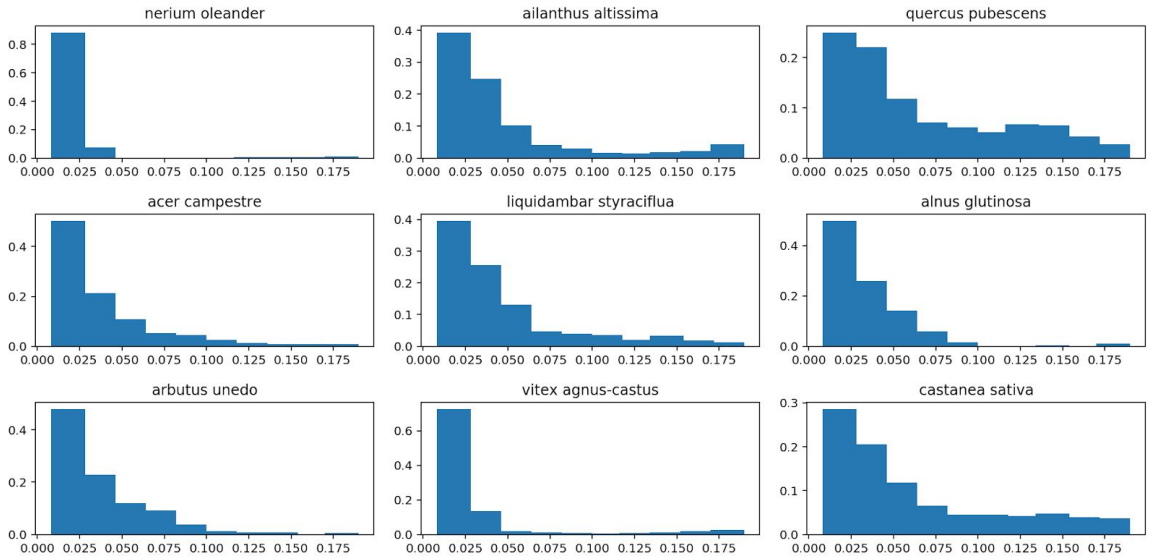


Implemented features and their usefulness

curvature_hist

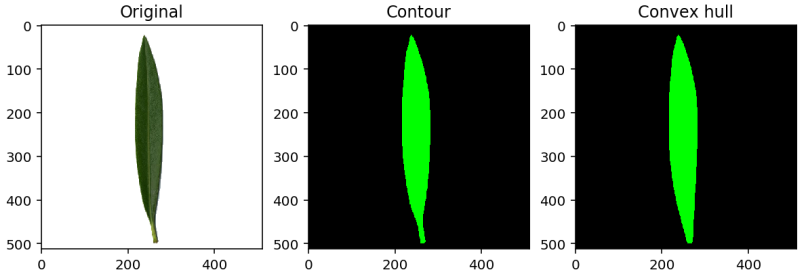
We are using the curvature as a measure of how fast a tangent line turns into a curve and describe the “pointiness” of the leaf. The usefulness comes from the fact that every species have a different leaf pointiness. Indeed, as we can compare below, the Nerium Oleander is similar to a needle, which is represented in the histogram as a single spike. For the Ailanthus altissima, the histogram is showing a significant peak similar to the Nerium Oleander as it can also be seen as a needle; however, the histogram is also showing that the leaf is also composed of groups of smaller pointiness. Finally, the Quercus Pubescens is described as a leaf formed a wide range of pointiness; however, the significant shape is less in a needle form than the two other leaves. After some tweaking, we found out that 10 is the right amount of bins for the histogram and that 0.18 is good for vmax.



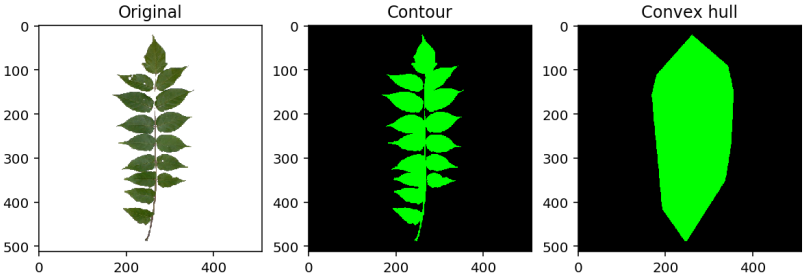


hull_concave_ratio

This feature differentiates leaves based on how concave they are. A perfectly round leaf with no concave elements will have a convex hull pretty much identical to its contour. A leaf with lots of stems branching out and leaving empty space between them will have long contour, but the hull will englobe it all. By checking the ratio between the hull area and the contour area, we get an idea of how concave a leaf is. Here is an example of a convex leaf:



This leaf has a contour pretty much equal to its convex hull. However, a concave leaf shows a more significant difference:

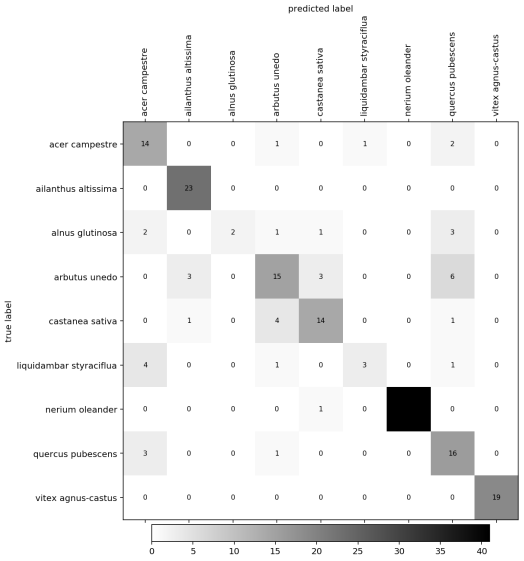


We can see that the contour is very intricate and quite different from the hull.

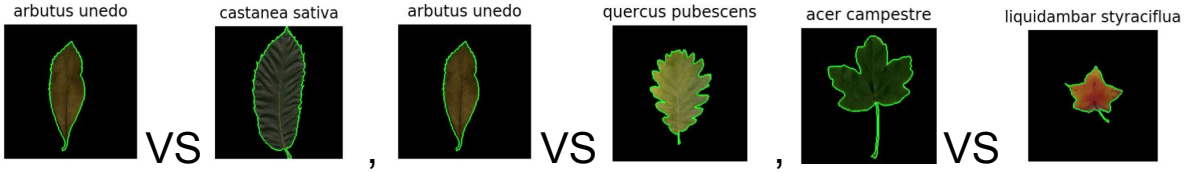
kNN Results

After some hours of tweaking, we came out with a fair solution, for a suitable algorithm with a weighted precision and recall averages of 0.81 and 0.79 respectively. The main diagonal on the confusion matrix give us good feedback on the classification success.

	precision	recall	f1-score	support
acer campestre	0.61	0.78	0.68	18
ailanthus altissima	0.85	1.00	0.92	23
alnus glutinosa	1.00	0.22	0.36	9
arbutus unedo	0.65	0.56	0.60	27
castanea sativa	0.74	0.70	0.72	20
liquidambar styraciflua	0.75	0.33	0.46	9
nerium oleander	1.00	0.98	0.99	42
quercus pubescens	0.55	0.80	0.65	20
vitex agnus-castus	1.00	1.00	1.00	19
micro avg	0.79	0.79	0.79	187
macro avg	0.79	0.71	0.71	187
weighted avg	0.81	0.79	0.78	187



Based on the confusion matrix, the following leaves are confused (>=4), as we can see, it makes sense for a human eye.



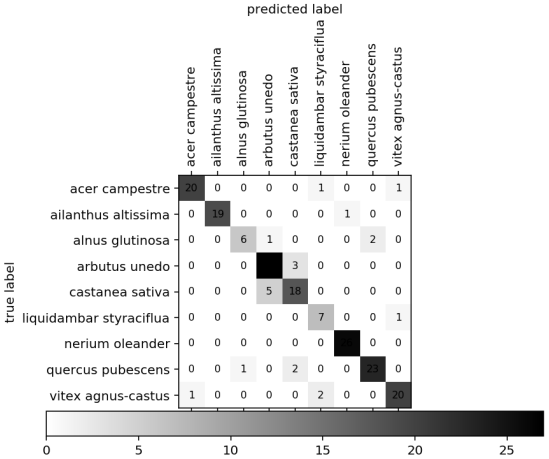
KNN Parameter choice

For our kNN, we decided to use all our extracted features (eccentricity, curvature_histogram, and hull_concave_ratio) combined with some brute force on the k neighbors, usually resulting with k=6 or k=7 (for this result).

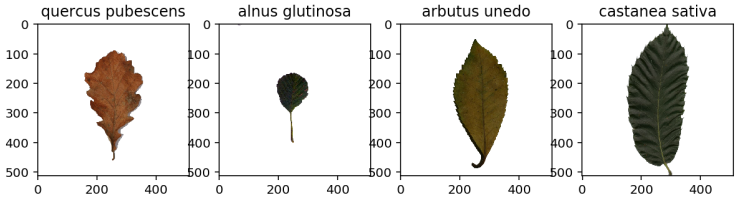
MLP Results

The MLP performed very well on average, with weighted precision and recall averages of 0.89 and 0.89 respectively. The confusion matrix gives us a clear view of the correct classifications, the main diagonal.

	precision	recall	f1-score	support
acer campestre	0.95	0.91	0.93	22
ailanthus altissima	1.00	0.95	0.97	20
alnus glutinosa	0.86	0.67	0.75	9
arbutus unedo	0.82	0.90	0.86	30
castanea sativa	0.78	0.78	0.78	23
liquidambar styraciflua	0.70	0.88	0.78	8
nerium oleander	0.96	1.00	0.98	26
quercus pubescens	0.92	0.88	0.90	26
vitex agnus-castus	0.91	0.87	0.89	23
micro avg	0.89	0.89	0.89	187
macro avg	0.88	0.87	0.87	187
weighted avg	0.89	0.89	0.89	187



The plants with the incorrect classifications were 'Quercus Pubescens' and 'Alnus Glutinosa'. For the 'Alnus Glutinosa', the small number of samples in the test set means that any error will have a big impact on the class accuracy. For 'Quercus Pubescens', we can observe which leaves are commonly falsely classified as each other:



All these leaves seem to have a similar edge, meaning the curvature histogram will be close. Since the edges are similar, the hull ratio will be similar as well. To better classify these samples, we could have used a feature to extract color from the samples since that varies a lot between them.

MLP Parameter choice

While most parameters of our Multi-Layer Perceptron did not give significantly different results (alpha, activation function...), we had to make a choice regarding the size of the hidden layers. To do this, we decided to test all combination of values for the hidden layers, first a single hidden layer with 1 to 10 nodes, then test two hidden layers with 1 to 10 nodes each. We simply set up a loop and calculated results with each setup, then sorted them by accuracy on the training set with a 65/35 training/test split on our data. We found the highest performing combination to be two hidden layers with 6 nodes each (2,6).
While this method would not be suited for very large data sets due to the "brute force" approach, we found it ideally suited for our data since all 100 combinations can be calculated in under a minute. For much larger datasets, an estimation based on experience would have to be performed, and the results might not be as accurate.

Analysis of the results

We found out that a split of 65/35 on our data is providing a good enough spread of all plant species. Indeed, the data for each species in the dataset are not equal in amount, and while testing, some could not show up and be evaluated.
Concerning the algorithms used; while in theory, a MLP is more elegant than a KNN, they both perform similarly on our data and both have value. A KNN is more intuitive and allows us to understand why a certain sample was classified a certain way (we can visualize a sample in a neighborhood of similar samples) whereas a MLP remains quite mysterious in its workings. Since both methods give similar and quite good results, we can assume that our features were useful in determining differences and similarities in the data. The classes that were difficult to distinguish were the same for both methods: Quercus Pubescens' and 'Alnus Glutinosa' had the most incorrect classifications.