Plant species identification using persistent homology

Hemming Ma

Abstract. We investigate whether the venous structure of a plant's leaf can be used to identify its species. In particular, we train a K-NN model using the persistent entropy, a scalar associated to persistence diagrams, of the venous structure of leaves from four species of plants as features. The final model was trained on the persistent entropy of 160 plant leaves and achieved a maximal average accuracy of 0.813, answering the question in the affirmative in our limited study.

Contents

1. Introduction and problem statement

It is a truth universally acknowledged that plants play an indispensable part for life on earth. Plant biodiversity is in decline, and an increase of conservation and protection efforts are needed in order to curb this trend [Pim+14]. Being able to identify plant species is crucial in these processes among others [Cop+12]. Traditional plant species identification requires extensive taxonomic knowledge and there is a shortage of people with such expertise (the so called *taxonomic impediment*) [Eng+21], whence it has become important to make this process doable even for non-experts.

There are many features of a plant that a computer can use to identify its species such as the contour of its leaves or its flowers [WM17]. In our paper, we use the venous structure (venation) of the plants to try identify its species together with a topological data analysis approach, seeking an answer to the question if *can we use the venation of a plant's leaves to identify its species?*

We build a ML workflow, where we extract venation from images of plant leaves, quantify them using persistent homology and finally use this to train a K-NN classification algorithm. All code can be found here.

The author would also like to thank Dr. [redacted] for the idea to study the persistent homology of plant leaves.

Date: April 2023.

2. The dataset and pre-processing

The raw images of plant leaves are taken from [Cho+19] which is a database of leaf images from 12 different plant species classified as either healthy or diseased. The dataset for our study consisted of 40 healthy leaf images of four plant species: mango, pongamia pinnata, basil and lemon leaves (see figure 1) making the total number of images 160. The average size of these images is 1.47 MB with a minimal size of 1.22 MB which means that they are too large to handle directly. For this reason we pre-process the images with the goal of reducing their file size.

We observe that the images contain a lot of irrelevant information, with many having more than half the image be the background. To remove the background, we used OpenCV [Bra00] to first threshold the greens in each image so the contour of the leaves could be found and subsequently its bounding box. The leaves were then cropped to their bounding boxes. Additionally, we down-sampled the images by 50% and increased the contrast. See figure 2 for the pre-processing of a pongamia pinnata leaf `0007_0026.jpg`.

This pre-processing scheme reduced the average file size by 92% to an average of 0.12 MB and lowest to 0.034 MB.

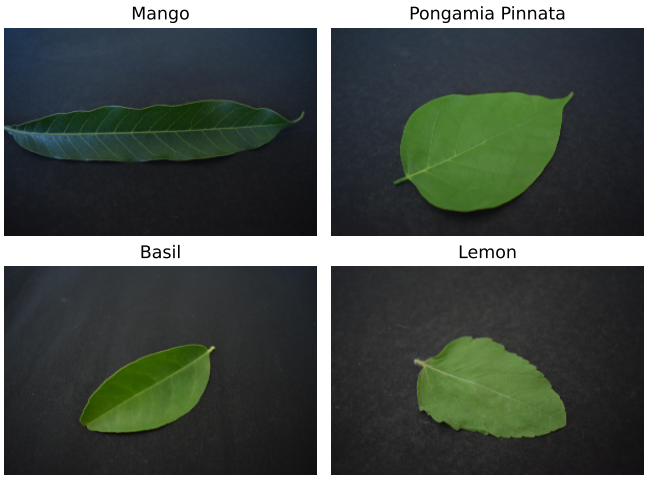

Figure 1. Representatives of each species in the dataset.

3. Venation extraction

Two options were explored to extract the venation from the processed images. Initially, a ridge detection algorithm [Wik23] was applied followed by a threshold but this produced unsatisfactory results. Instead, we used ilastik [Ber+19] which is an interactive tool for image classification and segmentation. In ilastik, we trained a model to separate the venous structure of a leaf from the lamina (the space between the veins) and background using two labels. From the ilastik segmentation (see 3, ilastik), it was easy to produce a binary image. The binary image produced contains a lot of redundant topological information as the extracted venation have thickness (see figure 3, Binary), to remove this we skeletonized it using scikit-image [Wal+14].

4. Computing persistent homology and extracting features

4.1. **Persistent homology.** To compute the persistent homology of the extracted venation we use giotto-tda [Tau+20], a Python topological data analysis library.

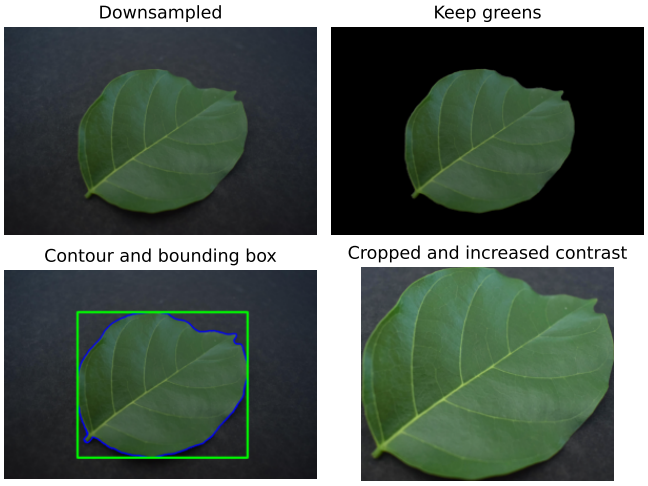

FIGURE 2. The pre-processing procedure as described in section 2 applied to a pongamia pinnata leaf `0007_0026.jpg`.

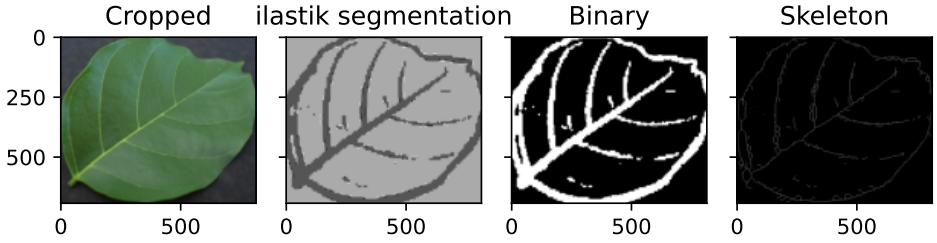

FIGURE 3. Venation extraction process applied to pongamia pinnata leaf `0007_0026.jpg`.

In order for giotto-tda to calculate the Vietoris–Rips filtration and the number of zero and one dimensional simplices we need to convert the venation skeletons into point cloud data. To do this, we take each coordinate point (x, y) in the skeleton and append it to a list of tuples (x, y), see figure 4.

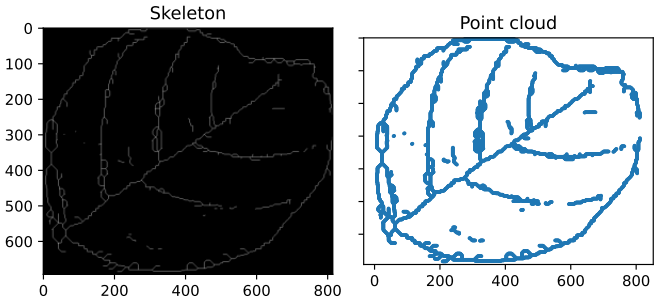

FIGURE 4. Point cloud representation of the skeleton on the left from figure 3.

This produces point cloud representations of the skeletons consisting of on the order of 10^5 points, which is too many to calculate the persistent homology in a

reasonable time (did not finish within 20 minutes). Our solution is to randomly (uniform) select a proportion $k \in [0, 1]$ of all points. By visual inspection in figure 5 we see that most features seem to be preserved under this process, we will investigate the dependence on k in section 4.3.

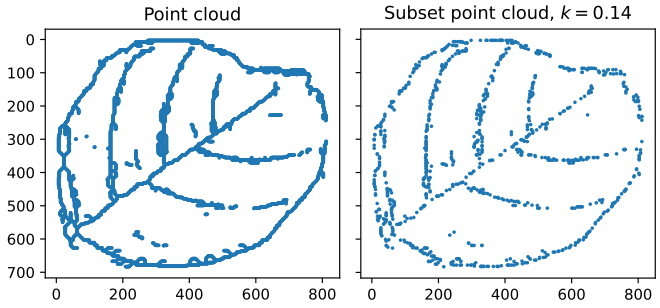

FIGURE 5. Randomly selected subset of the point cloud representation from figure 4 consisting of $k = 0.14$ of all points. The subset contains 1000 points.

With the subset in hand, gitto-tda can calculate the persistent homology. The persitence diagram for the pongamia pinnata leaf is shown in figure 6.

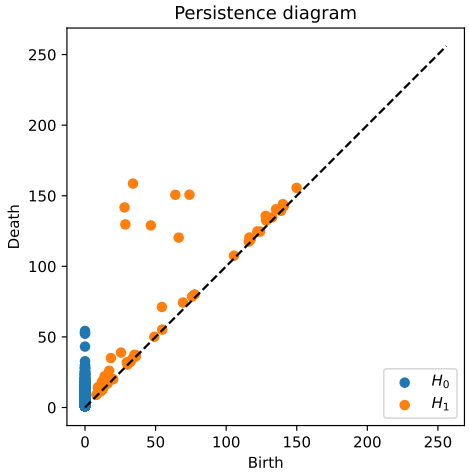

FIGURE 6. Persistence diagram of a pongamia pinnata leaf `0007_0026.jpg`. H_0 represents the degree 0 persistent homology and H_1 the degree 1 persistent homology.

4.2. **Persistent entropy.** To extract features from the persistence diagram we calculate the *persistent entropy* for each homological dimension.

Definition 4.1. [Tau+20, glossary] If $D = \{(b_i, d_i)\}_{i \in I}$ is a persistence diagram where each $d_i < \infty$, then its *persistent entropy* is

$$S(D) = \sum_{i \in I} p_i \log\left(\frac{1}{p_i}\right)$$

where

$$p_i = \frac{d_i - b_i}{L_D} \quad \text{and} \quad L_D = \sum_{i \in I} (d_i - b_i).$$

[AGR17] has shown that this quantity is a stable way of comparing persistence diagrams, meaning that is robust to small changes and that it is useful for discerning between different diagrams.

For instance, we turn the persistence diagram in figure 6 to the feature vector

$$(9.50317359 \quad 4.0271918)$$

where the first entry is the persistent entropy of the degree zero 0 persistence diagram and the second the persistent entropy of the degree 1 persistence diagram.

4.3. **Dependence on subset size.** We study how the final feature vectors depend on the size of subset k chosen, by computing the feature vector for increasing values of k for a pongamia pinnata leaf, see figure 7. Unfortunately, none of the values of k tested make the final feature vector independent of k, i.e. there is no k_0 so that for all $k \geq k_0$ we have that the feature vector remains the same. This means that we should choose a k as large as possible to get a result that most resembles the one we would obtain for the full point cloud.

The choice of k is also constrained by the computation time, as already for $k \geq 0.2$ computing the persistent homology of 160 skeletons becomes a lengthy process. Hence, we for this study take $k = 0.2$. With a more powerful computer a larger k may be chosen and we conjecture that this will improve the accuracy of the final model.

The feature vector computed for all 160 leaves are shown in figure 8.

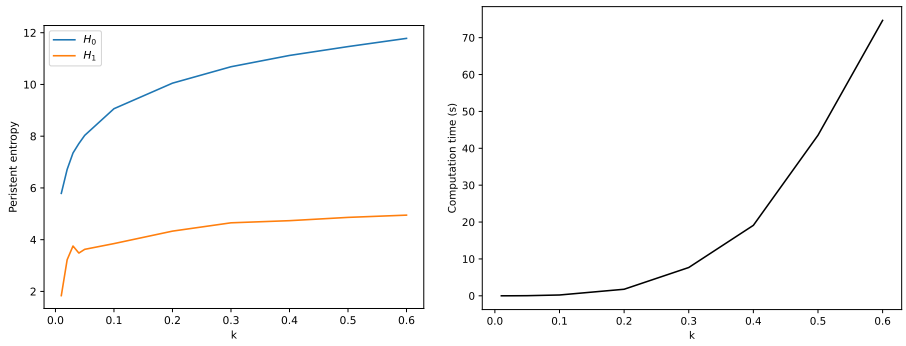

FIGURE 7. Persistent entropy values (left) and computation time (right) as a function of subset proportion k.

5. K-NN MODEL AND RESULTS

We use a K-nearest neighbours (K-NN) model implemented in scikit-learn [Ped+11] as a classifier. K-NN is a simple algorithm that labels a new data point p to the label that the majority of its K nearest neighbours has in the training data. Letting n denote the the carnality of the training data, we see that computing p's distance to every point in the training data can be done in $\mathcal{O}(n)$ time and finding its nearest neighbours in $\mathcal{O}(n \log n)$ time. Overall, this makes K-NN an $\mathcal{O}(n)$ algorithm.

To fit a K-NN model to our feature vectors, we need to first choose a K. To do this, we use repeated 5-fold cross validation and pick the K value that yields the

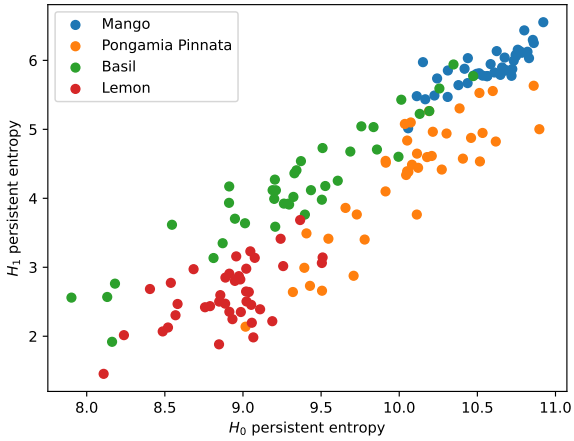

FIGURE 8. Visualisation of all feature vectors, each point corresponds to the 0th and 1th degree persistent entropy of a single image of leaf.

maximum average accuracy. The results of this study is shown in figure 9 and we conclude that taking $K = 4$ yields the highest average accuracy of 0.813.

The final decision boundary of the model is displayed in figure 10.

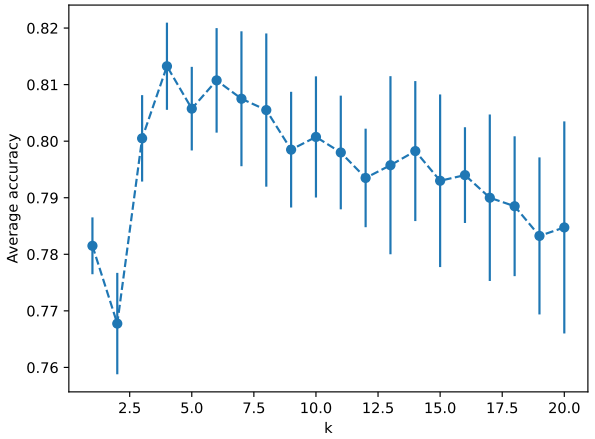

FIGURE 9. Average repeated 5-fold cross validation, 25 repetitions. Maximum accuracy of 0.813 at $K = 4$.

6. CONCLUSIONS

The achieved accuracy of 0.813 is much greater than that of the baseline accuracy $40/160 = 1/4$ coming from randomly guessing. We thus conclude that that persistent homology can be be used to identify a plant's species in our limited study.

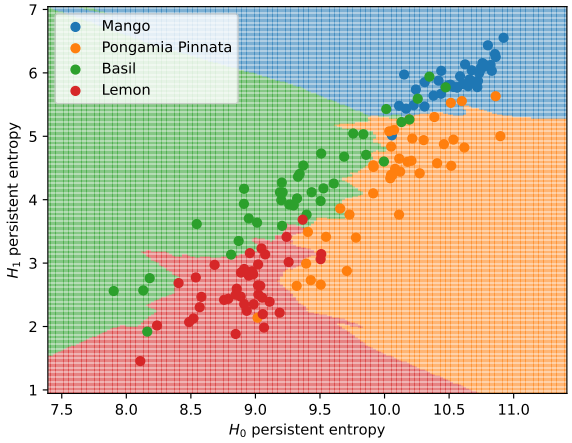

FIGURE 10. 4-NN decision boundary with average accuracy of 0.813.

7. REMARKS

The model is limited in two main ways, the first in the number of species used and secondly that the plant leaves are all taken on a black background as seen in figure 1. To work around the first is easy: [Cho+19] contains leaves of more species than used in our study and also leaves from more species. The second is hard, and we leave it as future work.

As discussed in section 5, K-NN is a $\mathcal{O}(n)$-time algorithm meaning that it may be slow when more leaves are studied. We suggest switching to a different model in this case, such as a random forest classifier.

Work can also be done to quantify the accuracy of the ilastik model used, which right now is validated only by visual inspection.

REFERENCES

[AGR17] Nieves Atienza, Roco Gonzalez-Diaz, and Matteo Rucco. "Persistent Entropy for Separating Topological Features from Noise in Vietoris-Rips Complexes". In: *CoRR* abs/1701.07857 (2017). arXiv: 1701.07857. URL: http://arxiv.org/abs/1701.07857.

[Ber+19] Stuart Berg et al. "ilastik: interactive machine learning for (bio)image analysis". In: *Nature Methods* (Sept. 2019). ISSN: 1548-7105. DOI: 10.1038/s41592-019-0582-9. URL: https://doi.org/10.1038/s41592-019-0582-9.

[Bra00] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000).

[Cho+19] Siddharth Singh Chouhan et al. *A Database of Leaf Images: Practice towards Plant Conservation with Plant Pathology*. Mendeley Data. 2019. DOI: 10.17632/hb74ynkjcn.4.

[Cop+12] James S. Cope et al. "Plant species identification using digital morphometrics: A review". In: *Expert Systems with Applications* 39.8 (2012), pp. 7562–7573. DOI: https://doi.org/10.1016/j.eswa.2012.01.073.

[Eng+21] Michael S Engel et al. "The taxonomic impediment: a shortage of taxonomists, not the lack of technical approaches". In: *Zoological Journal of the Linnean Society* 193.2 (Sept. 2021), pp. 381–387. ISSN: 0024-4082. DOI: 10.1093/zoolinnean/zlab072. eprint: https://academic.oup.com/zoolinnean/article-pdf/193/2/381/49555079/zlab072.pdf. URL: https://doi.org/10.1093/zoolinnean/zlab072.

[Ped+11] F. Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[Pim+14] S. L. Pimm et al. "The biodiversity of species and their rates of extinction, distribution, and protection". In: *Science* 344.6187 (2014), p. 1246752. DOI: 10.1126/science.1246752. eprint: https://www.science.org/doi/pdf/10.1126/science.1246752. URL: https://www.science.org/doi/abs/10.1126/science.1246752.

[Tau+20] Guillaume Tauzin et al. *giotto-tda: A Topological Data Analysis Toolkit for Machine Learning and Data Exploration*. 2020. arXiv: 2004.02551 [cs.LG].

[Wal+14] Stéfan van der Walt et al. "scikit-image: image processing in Python". In: *PeerJ* 2 (June 2014), e453. ISSN: 2167-8359. DOI: 10.7717/peerj.453. URL: https://doi.org/10.7717/peerj.453.

[Wik23] Wikipedia. *Ridge detection — Wikipedia, The Free Encyclopedia*. http://en.wikipedia.org/w/index.php?title=Ridge%20detection&oldid=1146952445. [Online; accessed 23-April-2023]. 2023.

[WM17] Jana Wäldchen and Patrick Mäder. "Plant Species Identification Using Computer Vision Techniques: A Systematic Literature Review". In: *Archives of Computational Methods in Engineering* 25 (2017), pp. 507–543.

Email address: hemmingm@kth.se