24th European Conference on Artificial Intelligence - ECAI 2020
Santiago de Compostela, Spain

Automatic Extraction of Leaf Venation Complex Networks

Maciej Rapacz[1] and **Radosław Łazarz**[2]

Abstract. Venation network analysis stands as a promising direction of scientific research, providing new insights into the origins and influence of plant phenotypic traits. However, its applicability is limited by a lack of tools that would facilitate relevant data acquisition. Dicotyledons form complex reticulate networks, often elusive for regular scanning equipment, hindering the attempts to capture details of their anastomoses arrangement. Currently available professional solutions operate on high-resolution noise-free images obtained in a complex process of chemical clearing, sample staining, and computationally expensive digitizing. This work introduces a novel technique capable of detecting leaf vasculature on pixel level and extracting a graph representation of its structure while operating on lower resolution scans of unprocessed specimens. The proposed transformation pipeline is designed as an array of steps — featuring automatic leaf segmentation, machine learning-based vein recognition, a sequence of custom spatial and morphological filters, segment radii retrieval and, finally, a graph compression and denoising algorithm. Each of those stages was separately evaluated using a range of metrics, including a new one aimed at assessing the uniformity of the reconstructed network. Obtained results confirmed that the method performs well in terms of both qualitative and quantitative analysis, given the characteristic imperfections in the examined images.

1 Introduction

The leaves of flowering plants tend to exhibit complex, intricate venation structures [30]. Their most characteristic feature is the existence of numerous anastomoses [24] that distinguish them from trivial tree-like topologies present in more primitive organisms [43]. These elaborate networks evolved to serve a diverse set of functions, such as reinforcing mechanical stability of the lamina, transporting water and nutrients, or discouraging potential herbivores [33].

As a result, they became a compelling research topic for a wide range of fields. In the case of plant physiology, vein arrangements heavily influence given leaf hydraulic capacities [25, 34] and its maximum photosynthetic rate [6]. In the case of paleoecology, investigating them gives crucial insights into past climate changes [39] and how they affected macroevolutionary trends [5]. Topological traits of the vascular networks can be also used as a robust basis for species taxonomy [42], enabling automatised differentiation and classification [25]. Finally, developmental biology analyses venation graphs to find facts that would advance our general understanding of tissue growth and formation [36].

Additionally, given that planar reticular networks are ubiquitous and occur in various places other than vasculatures of dicotyledons and monocotyledons [8], new discoveries in this field could potentially benefit other areas of research such as studies of the blood vessels in the retina, liver or brain, anastomosing foraging networks built by slime molds and fungi, lowland river networks, human-made road systems, force chain networks in granular materials and other scientific fields, some of which may not yet exist [32].

Nevertheless, the study of leaf venation systems is often hindered by difficulties in data acquisition and retrieval; costly, slow, and destructive sample processing methods being the chief issue [2]. In consequence, there were multiple attempts at designing computer-assisted techniques that would aid such endeavours.

1.1 Existing attempts

Some of those techniques, while also operating on input in the form of leaf images, strive to solve problems not directly related to the focus of this research (e.g. leaf texture classification [9]). Others deal only with certain parts of the challenge, such as edge detection [31] or vein identification [14].

There are, however, works aimed directly at distilling and investigating full venation graphs. LEAF GUI [29] was an attempt to deliver software capable of automatic vascular network detection and analysis. The programme used two different thresholding methods (global and local) to generate a binary image — an approach that improved segmentation precision for unevenly illuminated images. However, it offered only a set of built-in statistics, without providing the ability to extract the graph for further examinations.

Another notable result was obtained during research on nested graph cycles and the possibility of utilising them as a unique phenotypic trait [32]. The authors used a dataset of chemically cleared and stained leaves, scanned at high resolution (6400 dpi). The images were processed using a range of techniques, including Otsu thresholding, Teh-Chin dominant point, and Delaunay triangulation. While the results confirmed the method overall robustness, the restrictive input quality requirement limited the possible range of its applications.

Other noteworthy achievements include procedures employing X-ray imaging [2], scanning of pre-skeletonised leaf samples [3], or aspiring to construct a general-purpose solution capable of understanding a broader array of biological structures (e.g. those present in wings of certain insects) [11]. Similar problems were also encountered in case of related domains, i.e road [40] or cardiovascular networks [28]. Nonetheless, developing an accessible and low-cost extraction technique remains a valid challenge.

[1] email: mrapacz@protonmail.com
[2] AGH University of Science and Technology, Poland, email: lazarz@agh.edu.pl

2 Data acquisition

In order to conduct the further research, it was necessary to acquire a sufficiently diverse dataset containing images of unprocessed angiosperm laminae. Due to their dense and reticular venation networks, locally occurring dicotyledons species were chosen as the plants of interest. The acquisition procedure itself consisted of two stages: samples gathering and their subsequent digitising.

Collecting of the leaf samples took place during July and August of 2018. It was conducted in multiple locations in the Małopolskie region of southern Poland: Dobra, Kraków, Polanka Wielka, Skawina, Zabierzów, and Libertów. Obtained material belonged to the following species: *Ulmus minor Mill.*, *Tilia L.*, *Fagus sylvatica L.*, *Aesculus hippocastanum L.*, *Acer platanoides L.*, *Sambucus L.*, *Populus L.*, *Juglans regia L.*, *Carpinus betulus L.*, *Quercus L.*, *Morus alba L.*, *Alnus Mill.*, *Corylus L.*, and *Vitis L.*. Once collected, the leaves were stored under pressure for a day prior to scanning in order to make them flattened and expose the venation networks better. They were not processed physically in any other way, i.e. no chemical cleaning or staining was involved.

Digitising was performed using an *Epson Perfection V300* scanner with 600 dots per inch optical resolution — a general-purpose non-professional apparatus. The gear choice was motivated by the desire to develop an extraction technique whose applicability would not be restricted by expensive equipment requirements. Scans were captured in a blacked-out environment with the device lid kept open, as those conditions provided the best results quality. Other approaches (such as scanning with a white background or with an additional light source above the sample) were assessed as inferior and rejected at the initial stages of the research.

Output images were initially 7019 pixels wide and 5096 pixels high, which amounted to roughly 65MB files. However, it was possible to optimise their size by cropping out the void space around the region of interest, reducing the effective area up to 9 times. No further modifications were applied to the photographs before forwarding them to the actual graph extraction pipeline. The final dataset consisted of 376 pre-cropped scans. Their example representative can be seen in Figure 2a.

3 Extraction pipeline

The proposed network extraction method can be expressed as a pipeline of operations, the output of one acting as the input of the subsequent step. The whole process is divided into two major stages: one dealing with the problem of vein detection and one responsible for the venation graph reconstruction.

3.1 Vessel detection

The aim of the first pipeline stage was to transform the initial image into a binary form that would enable a straightforward distinction between the veins and the non-vein areas. In the ideal case pixels containing vessels should be coloured white, leaving all the other ones black. Its function can be interpreted as a digital (and therefore more accessible) equivalent of the previously mentioned chemical staining and etching process. A simple supervised learning solution in form of a feed-forward artificial neural network was employed in order to achieve this objective.

3.1.1 *Leaf area detection*

Training a supervised model requires a set of examples that had been labelled beforehand by an external expert. In case of vasculature recognition such set is comprised of appropriately marked pixels: established as either empty or depicting a vein. An additional auxiliary classifier was used to facilitate the process of obtaining those markings. Its purpose was to identify the non-rectangular area occupied by the leaf lamina and, as a consequence, spare the need to label and analyse the irrelevant background regions during the steps to follow.

That task turned out to be a relatively uncomplicated one, as a simple shallow neural network has proven to be a sufficiently capable tool for dealing with it. The utilised architecture consisted of only two layers: input (75 units) and output (2 units). The input contained a pixel to be classified and its neighbourhood up to the radius r_l of 2 pixels (in the sense of L_∞ metric), each pixel represented by 3 values corresponding to the RGB channels. The reasoning behind the chosen r_l is detailed in Section 4.1.1.

Normalized Exponential Function (Softmax) [13] was used as the activation function on the output layer in order to make the output vector a probability distribution. The weights of the network were optimised using the *Stochastic Gradient Descent* [4] technique with learning rate parameter $\eta = 0.01$ and a *Categorical Cross-Entropy* [17] loss function.

The classifier yielded by the aforementioned configuration successfully selected inhabited regions of the scanned images, while ignoring noise introduced by dust specks and minuscule tissue remains that tend to soil the device glass when working with organic matter. Figure 2b demonstrates the example results of its work.

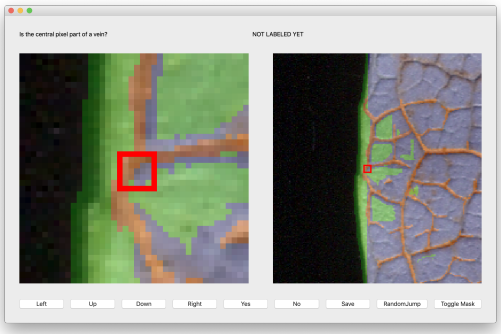


Figure 1. GUI application created to aid the manual labelling of the training samples

3.1.2 *Labelling procedure*

Pixel-level precision of the vein predictions is necessary to reconstruct the finest of the network vessels. Thus, the assembled set of training observations should strive to maintain the same order of fidelity. Nevertheless, collecting pixel-wise markings on a large scale is a tedious and time-consuming endeavour. To aid that effort and make it as effective as possible, a complementary GUI tool was implemented and employed during the labelling step. A sample screenshot of the application in the middle of the process can be seen in Figure 1.

There were three key design goals for this software: to provide the user with the necessary context to correctly label the given pixel, to minimise the number of actions required to do so, and to provide

a satisfactory perceivable latency (lower than 10 ms) [7]. The first one was achieved by the double-window interface (offering both a zoomed-in view and a broader picture), the second by offering customisable keyboard shortcuts. However, real-time rendering of the category mask, random jumping to an unmarked area, and persistence of the provided labels caused performance issues.

Several optimisations were deployed to deal with those problems and speed up the labelling experience. The current label mask for a whole image was precomputed and loaded from the hard drive during application start, with relevant cropping being applied to the presented view as suitable. Similarly, the list of possible jump locations was predetermined and stored in memory for future uses. Finally, saving of the gathered information became manually triggered to minimise the number of costly write operations. Together, they allowed a quick and reliable collection of user-provided labels.

3.1.3 Vessel detection

The actual detection of the vascular segments was performed by a slightly deeper neural network trained on the previously marked samples. It should be noted that it was sufficient to use a training set around three orders of magnitude smaller than the area of a single leaf, containing less than 0.15 % of its pixels (the validation and test sets were obtained in the same way, but using patches from samples unknown to the classifier). Additionally, an already trained model is usually reusable for other specimen belonging to the same species, lowering the need for manual labelling even further.

The utilised neural network encompassed three hidden layers, apart from the input and output one. The input layer once again consisted of 75 units (RGB channels of a square 5×5 pixel window). The hidden layers utilised the *Rectified Linear Unit (ReLU)* [16] activation function and comprised 35, 35, and 34 units, respectively. Lastly, the output layer contained the standard 2 *Softmax* [13] activated units. The *SGD* [4] optimiser and the *Categorical Cross-Entropy* loss [17] were used once more to obtain the final network weights.

This rather simple and straightforward model architecture was chosen over more sophisticated approaches for a number of reasons. First of all, the discussed task was extremely local in its nature — the objects to be detected were, by definition, always placed in the exact centre of the input image. Also, apart from the largest ones, the vessels were usually a few pixels wide and positioned close to one another. Those factors, together with the limited volume of the training data, caused contemporary approaches (such as Deep Convolutional Neural Networks [27], which promote spatially local input filters and stack layers of them to construct a hierarchy of increasingly global and abstract features) to be an unfavourable alternative. Finally, the subsequent stage is, to a certain level, noise-tolerant due to the nature of the information aggregating operators (such as median). Near-perfect classifier accuracy was therefore desirable, but not critical — and the described model provided an acceptable performance (its more comprehensive evaluation can be found in Section 4.1.2). Sample outcome of this pipeline step is depicted in Figure 2c.

3.2 Venation graph reconstruction

A greyscale image depicting the probability of finding a vein in a given pixel was a starting point for the second processing stage, aimed at recreating the whole venation structure in a graph form. That stage began with several image-preprocessing steps, then performed network extraction, simplification, and finally, segment width recognition.

3.2.1 Network preprocessing

During this phase, a sequence of transformations listed below was applied to the input image to expose the network topology and facilitate the subsequent graph reconstruction.

Binarisation First, the classifier predictions had to be binarised. It was done using a standard single threshold technique [35], assigning 1 to the pixels with prediction values $I_{x,y} \in [0, 1]$ higher than binarisation threshold t_b, and 0 otherwise.

$$\text{binarise}_{t_b}(I_{x,y}) = \begin{cases} 1 & \text{if } I_{x,y} \geqslant t_b\,, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

A threshold value of $t_b = 0.5$ was chosen for the context of this work; justification of that decision is included in Section 4.1.2. Figure 2d presents a typical image state after this step.

Quasi-median filter The role of the next step was to both eliminate outlier noise and regenerate missing parts of the veins disconnected due to not satisfying the t_b threshold. To do so, a quasi-median filtering scheme was designed and applied to the image. Normally, the median filter sets the value of a central pixel to the dominant class of a certain neighbourhood. However, veins are often relatively thin and, thus, tend to not constitute the majority in a given area. As a consequence, a threshold parameter t_q was introduced so as to adjust the filter to this particular phenomenon.

$$\text{quasi-median}_{t_q,r_q}(I_{x,y}) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n}\sum_{j=1}^{n} N_{i,j}(I_{x,y}) \geqslant t_q n^2\,, \\ 0 & \text{otherwise;} \end{cases} \tag{2}$$

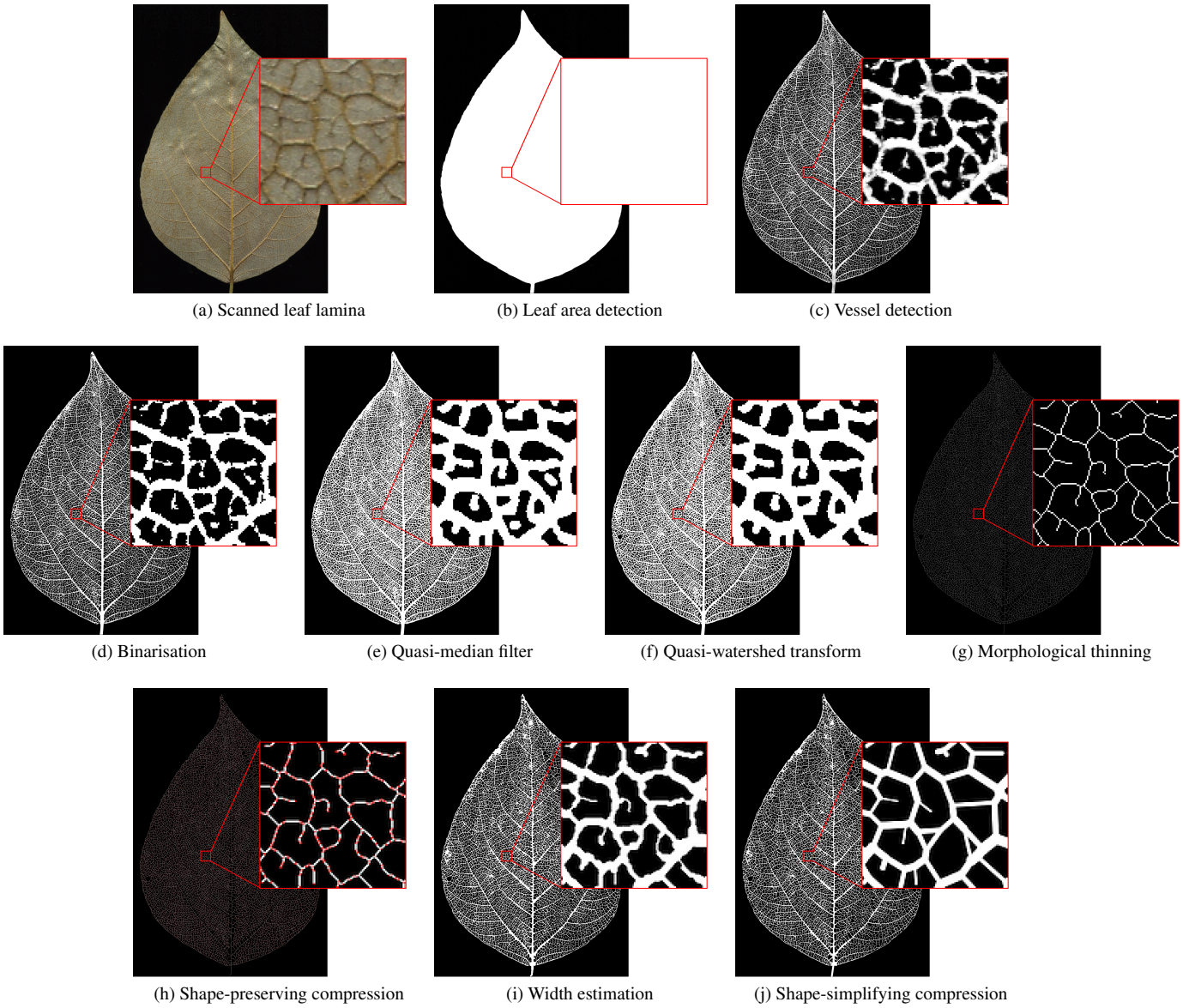
where $N_{n\times n}(I_{x,y}) \mid n = 2r_q + 1$ is the matrix containing neighbourhood of a given pixel $I_{x,y}$ up to the radius r_q.

The lower the threshold, the more often positive values are assigned, increasing the recall of the method, yet risking to compromise its precision by producing new false positives. The final values of $t_q = 0.2$ and $r_q = 2$ were chosen as a middle ground between connecting erroneously separated veins and introducing extra noise into the pipeline. Effects of the filter application are shown in Figure 2e.

Quasi-watershed transform Leaf vascular networks form a reticulate structure that keeps the organ nurtured [21]. The minuscule free-ending veinlets have the crucial role in this process, but to make the delivery possible they have to be connected with the root of the network. Hence, the reconstructed structure should form a connected graph.

However, noise and errors accumulated during the preceding steps might have introduced additional, unreachable components. In order to obtain a valid outcome, all those isolated sub-graphs had to be removed. This problem was solved using a custom stochastic filter inspired by the *Watershed Transform* [20].

The utilised implementation began by generating w_n random seed points (catchment basins) on the image. Each of them acted as a metaphorical fountain, immersing the neighbouring non-zero pixels in water. Water flow was executed by running *Breadth-First Search* simultaneously from each of the sources. A *Disjoint Set* [15] was employed to keep track of discovered components, performing a union operation whenever two of them overlap. In the end, only pixels belonging to the largest connected component are allowed to keep the



(a) Scanned leaf lamina (b) Leaf area detection (c) Vessel detection

(d) Binarisation (e) Quasi-median filter (f) Quasi-watershed transform (g) Morphological thinning

(h) Shape-preserving compression (i) Width estimation (j) Shape-simplifying compression

Figure 2. Step-by-step overview of the proposed network extraction pipeline

positive binary value. In case of the discussed images $w_n = 100$ has proven to be a sufficient number of basins; a sample result is depicted in Figure 2f.

Morphological thinning Some of the methods applied in related works use semi-automated techniques of junction and centre line detection [26]. However, they are not feasible for venation networks processing because of the problem scale — a single leaf scan contains at least several thousand junctions. Instead, to separate singular veins from the others and mark their centrelines automatically, the *Morphological Thinning* [23] filter was employed.

Every thinning method strives to satisfy the following four goals: to preserve the currently present connectivity, to leave the already existing curves unchanged, to create medial curves that lie along the midpoints of elongated objects and are as thin as possible, and finally, to do it in a lowest achievable number of operations. The utilised variant analyses the image iteratively, removing pixels that match a set of criteria, which themselves change slightly during odd and even iterations. Those criteria were designed to reduce densely populated areas, but simultaneously avoid breaking eight-connected components and 2×2 squares [18].

The output of the characterised filter is a skeletonised version of the venation image, depicting centrelines of the individual veins and their intersections. Example of such outcome is portrayed in Figure 2g.

3.2.2 *Graph reconstruction*

The binary skeleton provided by the precedent pipeline steps can be utilised as a basis for a naive recreation of the vessel graph. The

method works as follows: each non-zero pixel on the resulting image constitutes a graph node and there is a graph edge between such node and all the other nodes in its Moore neighbourhood [41] of range $r_m = 1$. The result is a raw network that faithfully reflects the input data but has several shortcomings.

The unnecessary high number of nodes and edges (up to 3×10^5) inflates the memory footprint of such structure and would significantly affect the running time of most graph analysis algorithms. Also, the fact that all edges are exactly 1 pixel long hinders the efforts to obtain meaningful features characterising them (such as the diameter of a given segment). Two graph compression algorithms (shape-preserving lossless and shape-simplifying lossy one) together with width estimation scheme were introduced to deal with the aforementioned problems.

Shape-preserving compression Due to the nature of the previously applied thinning transform the only nodes with a degree greater than 2 are those lying near the junctions. Therefore, it is potentially possible to replace a sequence of connected degree 2 nodes with a single edge equivalent. Nonetheless, such operation would destroy information about the given edge curvature and hamper future investigations of its parameters. Instead, the employed graph compression procedure aimed to reduce only those edges that form a single straight line.

The proposed algorithm works in an infinite loop. During each iteration, it lists all the non-intersection nodes. It then visits each one of them and, if the straight-line criterion is met, removes it along with its edges and creates a new edge connecting its neighbours. After such iteration, the number of such replacements is checked. If no changes were made in the current run, the algorithm stops.

Algorithm 1 Compression of the straight edges

$(V, E) \leftarrow G$
$s \leftarrow$ true
while s **do**
  $s \leftarrow$ false
  $D \leftarrow \{v \in V \mid \text{Degree}(v) = 2\}$
  for $d \in D$ **do**
    if $\text{Degree}(d) = 2$ **then**
      $\{m, n\} \leftarrow \{v \in V \mid (v, d) \in E\}$
      if $\text{Direction}(n, d) = \text{Direction}(d, m)$ **then**
        $V \leftarrow V \setminus \{d\}$
        $E \leftarrow E \setminus \{e \in E \mid d \in e\} \cup \{(m, n)\}$
        $s \leftarrow$ true

Since the initial edge placement was based on the Moore neighbourhood, nodes that belonged to separate branches but were lying too close to the intersection (i.e. 1 pixel away) became erroneously connected. Such faulty edges were nonsensical from the domain perspective but cluttered most of the junction points. An additional graph clean-up procedure was introduced in order to discard them.

It was derived from the following three observations: incorrect edges were always exactly 1 pixel long; they formed triangle-shaped (and rarely square-shaped) cliques; if the intersections were close to each other those cliques overlapped. Thus, the aim of the clean-up algorithm was to remove every clique satisfying the aforementioned conditions by replacing its components with a single node. The coordinates of that new node were equal to a median of the ones belonging to the nodes it supplanted.

The effect of those two sub-steps was a graph still completely consistent with the input image, but considerably smaller and free of faulty edges. Sample visualisation of such network can be seen in Figure 2h.

Algorithm 2 Removal of the intersection cliques

$(V, E) \leftarrow G$
$G' \leftarrow (V, \{e \in E \mid \text{Length}(e) = 1\})$
$C \leftarrow \text{ConnectedComponents}(G')$
for $c \in C$ **do**
  $Q \leftarrow \{q \in \text{Cliques}(c) \mid |q| > 2\}$
  for $q \in Q$ **do**
    $E \leftarrow E \setminus \{(m, n) \in E \mid m \in q \wedge n \in q\}$
    for $k \in q$ **do**
      for $l \in \{v \in V \mid (k, v) \in E\}$ **do**
        $E \leftarrow E \setminus \{(k, l)\} \cup \{(\text{Centre}(q), l)\}$
    for $u \in Q$ **do**
      if $q \cap u \neq \varnothing$ **then**
        $E \leftarrow E \cup \{(\text{Centre}(q), \text{Centre}(u))\}$
$V \leftarrow \{v \in V \mid \text{Degree}(v) > 0\}$

Vessel width estimation The graph obtained during the previous steps captured only the topological features of the given leaf vasculature. However, it did not include information about the diameter of individual vessels, crucial when analysing network flow capacity. This step aimed to retrieve that information by applying a simple inflate-and-stop technique.

The algorithm iterates over all edges once and tries to expand each of them until a stopping criterion is met. Since the processing of a single edge does not affect the other ones, the operation can be parallelised without further changes. The inflation begins with a set containing all pixels belonging to a given edge. Then it adds to the set all pixels in $r_m = 1$ Moore neighbourhood of the ones already taken. Next, it checks whether the stop condition is satisfied. If not, the loop is repeated and the current set becomes inflated once more.

The stopping criterion checks whether the percentage of pixels depicting veins in a pre-thinned image among all the inflation set pixels (left side of the inequality) is lower than a certain threshold (right side of the inequality). Formally, it can be expressed as

$$\frac{v}{(l + 2i)(2i + 1)} < 1 - \frac{l(1 - t_w)}{l + 2i}, \tag{3}$$

where i is the inflating iteration number, v is the number of vessel pixels, l is the original length of the edge and t_w is the sensitivity parameter governing the algorithm tolerance to missing vessel fragments.

In case of this work, a value of $t_w = 0.95$ was chosen, allowing only a small fraction of pixels in the expanded set to be empty. The provided formula is adaptive in its nature, i.e. with each iteration the condition becomes increasingly harsh, preventing noise-induced uncontrolled expansion. The estimated width is equal to the number of feasible iteration. Results of such assessment are visualised in Figure 2i.

Shape-simplifying compression The straight-line requirement of the first compression algorithm was a demanding one, limiting the possibility of edge aggregation to relatively short segments. Width estimation described in the previous step was, in consequence, done in a very local manner and yielded a noisy outcome.

To deal with that problem, and to further reduce the output graph size, another compression pass was performed, but this time omitting

the same-angle principle and only requiring the removed node degree to be equal to 2 in order to be discarded. The width w of the final aggregated edge was equal to the weighted sum of the intermediate ones:

$$w = \frac{\sum_i l_i w_i}{\sum_i l_i} \tag{4}$$

where w_i is the width of the i^{th} edge it replaced and l_i is its length. The outcome of that second compression algorithms (and the final result of the extraction pipeline) is presented in Figure 2j.

4 Tuning and evaluation

The following acronyms are used in table headers throughout this section: $\mathbf{F_1}$ (F-score), **P** (precision), **R** (recall), **TP** (true positives), **FP** (false positives), **FN** (false negatives), and **TN** (true negatives). Detecting a vein in a given pixel is treated as a positive result, the opposite outcome as a negative one.

4.1 Evaluation of the vessel detection stage

The initial stage consisted of typical machine learning tasks. Their evaluation was therefore executed by employing standard measures and tools, such as accuracy or cross-validation.

4.1.1 Leaf area detection tuning

The first parameter that needed to be tuned was the r_l radius determining the input of the leaf area detector. Even the smallest possible value $r_l = 1$ offered acceptable performance ($99.417 \pm 0.035\%$ accuracy, estimated using 10-fold cross-validation on a dataset containing almost 0.5×10^6 samples). However, closer inspection of the misclassified samples revealed that they were mostly dust specks, few pixels in size and far away from the region of interest. Increasing the radius to $r_l = 2$ (resulting in 5×5 pixel input window) solved that problem without overcomplicating the classifying model.

4.1.2 Vessel detection tuning

The vessel detecting classifier was tuned using the train-validation-test approach. Three separate datasets were obtained by labelling small fragments of leaf lamina, each dataset containing samples from a different area. Their sizes and class compositions are presented in Table 1. The training dataset was intentionally created as almost balanced, others reproduce more accurately the actual class distribution.

Table 1. Contents of vessel detector tuning datasets.

Dataset	Samples	Veins	Non-veins
Train	6437	3165 (49.17%)	3272 (50.83%)
Validate	3299	1018 (30.86%)	2281 (69.14%)
Test	3298	1083 (32.84%)	2215 (67.16%)

Over 10^3 various neural network configurations were evaluated in order to select the optimal one. All their parameters were picked randomly from the pre-set ranges unless specified otherwise. The random search of parameter space was utilised instead of grid search due to its statistically higher efficiency and lower required computation time [1].

Each considered arrangement started with a fixed 5×5 pixel input layer, followed by a varying number of hidden fully connected layers (between 1 to 4, each of them containing from 10 to 60 units), and a 2 unit output layer. Also, three different optimisers were tested: *SGD* [4] (with and without the *Nesterov Momentum* [37] modification), *Adam* [22], and *Nadam* [12]. Learning rates were picked uniformly from a $]0, 1]$ range. *Adam* hyper-parameters were assigned in accordance with the original authors' suggestion ($\beta_1 = 0.9$, $\beta_2 = 0.999$, $\varepsilon = 10^{-8}$). Network training was stopped if the loss value did not improve by at least $\Delta = 0.005$ during the last $p = 10$ iterations. In the end, all candidate architectures were compared using their accuracy scores on the validation dataset.

The winning arrangement has been already characterised in Section 3.1.3. Its performance was additionally assessed using the previously left aside test dataset. Details of the obtained results are presented in Table 2. Elbow rule together with Receiver Operating Characteristic and Precision-Recall curves [19] were used to establish an appropriate discrimination threshold t_b for the binarisation step (Section 3.2.1).

Table 2. Results obtained with the chosen vein detection model.

Dataset	$\mathbf{F_1}$	P	R	TP	FP	FN	TN
Validate	0.946	0.944	0.948	0.674	0.017	0.016	0.293
Test	0.939	0.926	0.951	0.647	0.025	0.016	0.312

4.2 Graph reconstruction evaluation

Quantitative evaluation of the following steps (i.e. image filters application and network reconstruction) is conceptually harder, as there exists no indisputable ground truth to which one can compare the pipeline outcome. Consequently, the measures discussed in this section should be interpreted as guidelines to correct performance judgement, rather than the optimisation goals themselves.

4.2.1 Quasi-median filter threshold

The threshold parameter t_q of the quasi-median filter represents the trade-off between being able to rectify detection result and creating unintended false positives. Low values cause small segments disconnected by noise from a central component to stay this way. High ones, on the other hand, create new edges non-existent in the original image.

Table 3. Skeletonised image consistency scores depending on t_q.

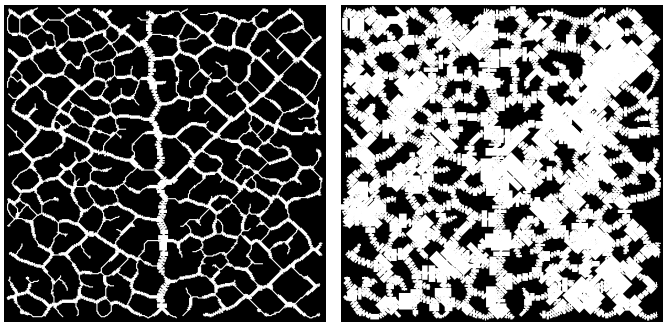
$\mathbf{t_q}$	$\mathbf{F_1}$	P	R	TP	FP	FN	TN
0.1	0.342	0.903	0.211	0.806	0.004	0.150	0.040
0.2	0.338	0.933	0.206	0.808	0.003	0.151	0.039
0.3	0.321	0.955	0.193	0.809	0.002	0.153	0.037
0.4	0.301	0.966	0.178	0.809	0.001	0.156	0.034
0.5	0.237	0.979	0.135	0.810	0.001	0.164	0.026
0.6	0.129	0.993	0.069	0.810	0.000	0.177	0.013

In order to estimate the desirable t_q a quasi-accuracy measure was proposed. To calculate it, the outcome of the binarisation step was treated as the correct class labels, while pixels from the thinned image served as the hypothetical classifier output. As a result, the obtained scores (presented in Table 3) depicted the consistency between the network skeleton and the original vein predictions. If the applied filtering prevented noise-caused detachments from the core network the score should improve. Thus, $t_q = 0.2$ was selected using the

elbow rule, as for the subsequent values the associated F_1 scores started to deteriorate.

4.2.2 Network uniformity measure

The process of designing the graph extraction pipeline highlighted one additional issue. During the method development phase, there was a need for a measure that would express the general quality of the output structure, i.e. lack of artefacts and anomalies. The first attempt at specifying such score mirrored the approach used in Section 4.2.1: the final graph was rendered and compared with the initial vessel prediction. However, the gathered F_1 values were often unrelated to the common-sense assessment of the result correctness.



(a) $\delta_u = 0.49 \times 10^6$, $F_1 = 0.58$ (b) $\delta_u = 14.67 \times 10^6$, $F_1 = 0.70$

Figure 3. Uniformity measure and its relation to the F_1 score

A novel measure, inspired by an observation of arterial branches blood flow [38], was introduced to deal with that problem. In case of nutrition flow networks, the relation between radii of adjacent vessel tends to satisfy the following equation:

$$R^\alpha = R_1^\alpha + R_2^\alpha + \cdots + R_n^\alpha \,, \tag{5}$$

where R is the radius of the main branch, $R_i \mid i \in \{1, \ldots, n\}$ are the radii of the thinner sub-branches and α is a parameter specific to a given vessel type. For the sake of this work, it was assumed that α is roughly constant and equal to 2.7 — a guess that should be verified in future studies.

The proposed score measures how much a given graph violates the law described above. In formal terms it could be defined as

$$\delta_u = \sum_{i=1}^{m} |R^\alpha - \sum_{j=1}^{n_i} R_{i,j}^\alpha| \,, \tag{6}$$

where α is the previously discussed coefficient, m is the number of intersections, n_i is the number of sub-branches in a given intersection, and $R_{i,j}$ are the appropriate radii. An example of its practical application is demonstrated in Figure 3, where (unlike the F_1 score) it favours the cleaner variant of width reconstruction.

5 Conclusions

The proposed pipeline is a robust tool, enabling efficient extraction of the vessel networks from scanned leaf images. Choices made during its design were driven by two principles: to maintain simplicity and to perform both qualitative and quantitative results evaluation when the correct decision is ambiguous. Full-size example of the final reconstructed venation graph can be seen in Figure 4. After the last compression step, it is comprised of 32500 nodes and 44297 edges (together with information about their diameter), confirming the method ability to capture low-level details of analysed structures.

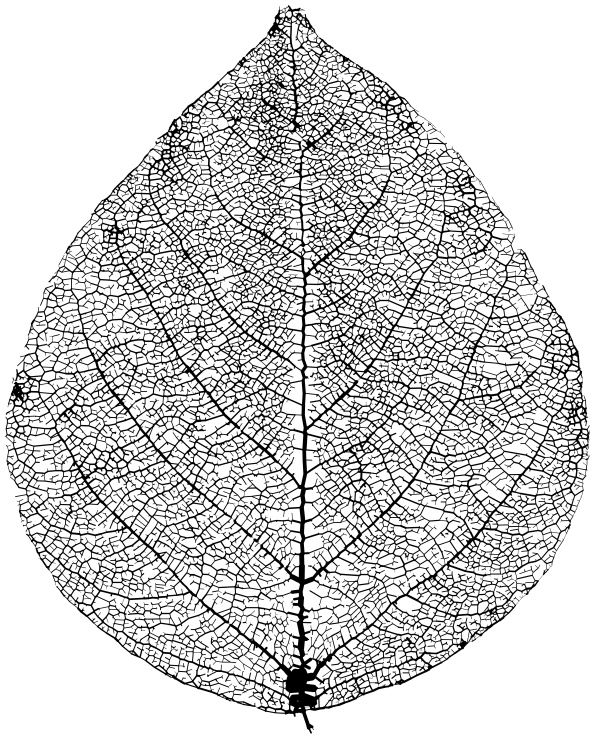


Figure 4. Example of an extracted leaf venation graph

Nonetheless, there are various areas left open for subsequent research. Vessel detection is a context-dependent activity (the probability that a given pixel contains a vein is higher if its neighbours contain it as well). Integrating that information into the utilised classifiers is a promising solution for increasing their accuracy and reliability. Alternative pipeline arrangements and configurations (using different perceptive windows and filter sequences) could be also explored and evaluated, potentially boosting the method performance. Furthermore, it might be fruitful to design and implement a middleway graph simplification algorithm: one that would keep nodes in key bending points of a given segment while still considerably reducing their number. Finally, the gathered networks themselves await an in-depth investigation with dedicated tools, such as invariant-based fingerprinting [10].

The presented outcomes were obtained using an Apple MacBook Pro 15 2019 Series personal computer (2.6 GHz Intel Core i7, 16 GB 2400 MHz DDR4, Intel UHD Graphics 630 1536 MB). The average pipeline execution time for a single input image was equal to 17.5 ± 6.6 min, with vessel detection (10.1 ± 3.7 min) and width estimation (2.9 ± 1.1 min) being the chief contributors (estimates are based on 12 repeated runs). The high variance of the reported values is a result of differences between individual leaves, particularly in size and venation structure. However, it should be noted that optimising the method calculations time was not the focus of this study and, as a consequence, utilised implementations were often unrefined (e.g. no

GPU support for neural network). If necessary, it would possible to parallelise the computation heavy steps in a map-reduce scheme: the vein segments (as well as the whole leaf scans) can be processed independently from one another.

ACKNOWLEDGEMENTS

The research presented in this paper was financed using the funds assigned by the Polish Ministry of Science and Higher Education to AGH University of Science and Technology.

REFERENCES

[1] James Bergstra and Yoshua Bengio, 'Random search for hyper-parameter optimization', *Journal of Machine Learning Research*, **13**(Feb), 281–305, (2012).
[2] Benjamin Blonder, Francesco De Carlo, Jared Moore, Mark Rivers, and Brian J Enquist, 'X-ray imaging of leaf venation networks', *New Phytologist*, **196**(4), 1274–1282, (2012).
[3] S Bohn, B Andreotti, S Douady, J Munzinger, and Y Couder, 'Constitutive property of the local organization of leaf venation networks', *Physical Review E*, **65**(6), 061914, (2002).
[4] Léon Bottou, 'Large-scale machine learning with stochastic gradient descent', in *Proceedings of COMPSTAT'2010*, 177–186, Springer, (2010).
[5] C Kevin Boyce, Tim J Brodribb, Taylor S Feild, and Maciej A Zwieniecki, 'Angiosperm leaf vein evolution was physiologically and environmentally transformative', *Proceedings of the Royal Society B: Biological Sciences*, **276**(1663), 1771–1776, (2009).
[6] Tim J Brodribb, Taylor S Feild, and Gregory J Jordan, 'Leaf maximum photosynthetic rate and venation are linked by hydraulics', *Plant physiology*, **144**(4), 1890–1898, (2007).
[7] Jake Brutlag. Speed matters for google web search, 2009.
[8] Kenneth M Cameron and William C Dickinson, 'Foliar architecture of vanilloid orchids: insights into the evolution of reticulate leaf venation in monocotyledons', *Botanical Journal of the Linnean Society*, **128**(1), 45–70, (1998).
[9] James S Cope, Paolo Remagnino, Sarah Barman, and Paul Wilkin, 'Plant texture classification using gabor co-occurrences', in *International Symposium on Visual Computing*, pp. 669–677. Springer, (2010).
[10] Wojciech Czech and Radosław Łazarz, 'A method of analysis and visualization of structured datasets based on centrality information', in *International Conference on Artificial Intelligence and Soft Computing*, pp. 429–441. Springer, (2016).
[11] Michael Dirnberger, Tim Kehl, and Adrian Neumann, 'Nefi: Network extraction from images', *Scientific reports*, **5**, 15669, (2015).
[12] Timothy Dozat, 'Incorporating nesterov momentum into adam', (2016).
[13] Rob A Dunne and Norm A Campbell, 'On the pairing of the softmax activation and cross-entropy penalty functions and the derivation of the softmax activation function', in *Proc. 8th Aust. Conf. on the Neural Networks, Melbourne*, volume 181, p. 185. Citeseer, (1997).
[14] Hong Fu and Zheni Chi, 'A two-stage approach for leaf vein extraction', in *International Conference on Neural Networks and Signal Processing, 2003. Proceedings of the 2003*, volume 1, pp. 208–211. IEEE, (2003).
[15] Harold N Gabow and Robert Endre Tarjan, 'A linear-time algorithm for a special case of disjoint set union', *Journal of computer and system sciences*, **30**(2), 209–221, (1985).
[16] Xavier Glorot, Antoine Bordes, and Yoshua Bengio, 'Deep sparse rectifier neural networks', in *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pp. 315–323, (2011).
[17] Ian Goodfellow, Yoshua Bengio, and Aaron Courville, *Deep learning*, MIT press, 2016.
[18] Zicheng Guo and Richard W Hall, 'Parallel thinning with two-subiteration algorithms', *Communications of the ACM*, **32**(3), 359–373, (1989).
[19] James A Hanley and Barbara J McNeil, 'The meaning and use of the area under a receiver operating characteristic (roc) curve.', *Radiology*, **143**(1), 29–36, (1982).
[20] Robert M Haralick and Linda G Shapiro, 'Image segmentation techniques', *Computer vision, graphics, and image processing*, **29**(1), 100–132, (1985).
[21] Vini Katyal, 'Leaf vein segmentation using odd gabor filters and morphological operations', *International Journal of Advanced Research in Computer Science*, **3**(3), (2012).
[22] Diederik P Kingma and Jimmy Ba, 'Adam: A method for stochastic optimization', *arXiv preprint arXiv:1412.6980*, (2014).
[23] Louisa Lam, Seong-Whan Lee, and Ching Y Suen, 'Thinning methodologies-a comprehensive survey', *IEEE Transactions on pattern analysis and machine intelligence*, **14**(9), 869–885, (1992).
[24] Norman MacDonald, 'Trees and networks in biological models', (1983).
[25] Athena D McKown, Hervé Cochard, and Lawren Sack, 'Decoding leaf hydraulics with a spatially explicit model: principles of venation architecture and implications for its evolution', *The American Naturalist*, **175**(4), 447–460, (2010).
[26] Zelang Miao, Bin Wang, Wenzhong Shi, and Hua Zhang, 'A semi-automatic method for road centerline extraction from vhr images', *IEEE Geoscience and Remote Sensing Letters*, **11**(11), 1856–1860, (2014).
[27] Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller, 'Methods for interpreting and understanding deep neural networks', *Digital Signal Processing*, **73**, 1–15, (2018).
[28] Javier A Montoya-Zegarra, Erica Russo, Peter Runge, Maria Jadhav, Ann-Helen Willrodt, Szymon Stoma, Simon F Nørrelykke, Michael Detmar, and Cornelia Halin, 'Autotube: a novel software for the automated morphometric analysis of vascular networks in tissues', *Angiogenesis*, **22**(2), 223–236, (2019).
[29] Charles A Price, 'Leaf gui: analyzing the geometry of veins and areoles using image segmentation algorithms', in *High-throughput phenotyping in plants*, 41–49, Springer, (2012).
[30] Charles A Price, Scott Wing, and Joshua S Weitz, 'Scaling and structure of dicotyledonous leaf venation networks', *Ecology letters*, **15**(2), 87–95, (2012).
[31] R Radha and S Jeyalakshmi, 'An effective algorithm for edges and veins detection in leaf images', in *2014 World Congress on Computing and Communication Technologies*, pp. 128–131. IEEE, (2014).
[32] Henrik Ronellenfitsch, Jana Lasser, Douglas C Daly, and Eleni Katifori, 'Topological phenotypes constitute a new dimension in the phenotypic space of leaf venation networks', *PLoS computational biology*, **11**(12), e1004680, (2015).
[33] Anita Roth-Nebelsick, Dieter Uhl, Volker Mosbrugger, and Hans Kerp, 'Evolution and function of leaf venation architecture: a review', *Annals of Botany*, **87**(5), 553–566, (2001).
[34] Lawren Sack and N Michele Holbrook, 'Leaf hydraulics', *Annu. Rev. Plant Biol.*, **57**, 361–381, (2006).
[35] Jaakko Sauvola and Matti Pietikäinen, 'Adaptive document image binarization', *Pattern recognition*, **33**(2), 225–236, (2000).
[36] Enrico Scarpella, Michalis Barkoulas, and Miltos Tsiantis, 'Control of leaf and vein development by auxin', *Cold Spring Harbor perspectives in biology*, **2**(1), a001511, (2010).
[37] Ilya Sutskever, James Martens, George Dahl, and Geoffrey Hinton, 'On the importance of initialization and momentum in deep learning', in *International conference on machine learning*, pp. 1139–1147, (2013).
[38] Norio Suwa, Takashi Niwa, Hitoshi Fukasawa, and Yasuhiko Sasaki, 'Estimation of intravascular blood pressure gradient by mathematical analysis of arterial casts', *The Tohoku journal of experimental medicine*, **79**(2), 168–198, (1963).
[39] D Uhl and V Mosbrugger, 'Leaf venation density as a climate and environmental proxy: a critical review and new data', *Palaeogeography, Palaeoclimatology, Palaeoecology*, **149**(1-4), 15–26, (1999).
[40] Weixing Wang, Nan Yang, Yi Zhang, Fengping Wang, Ting Cao, and Patrik Eklund, 'A review of road extraction from remote sensing images', *Journal of traffic and transportation engineering (english edition)*, **3**(3), 271–282, (2016).
[41] Eric W Weisstein, 'Moore neighborhood', *From MathWorld–A Wolfram Web Resource. http://mathworld. wolfram. com/MooreNeighborhood. html*, (2005).
[42] Bing-Qiang Xu, Nian-He Xia, Shao-Ping Wang, and Gang HAO, 'Leaf venation of osmanthus (oleaceae) from china and its taxonomic significance', *Guihaia*, **27**(5), 697–705, (2007).
[43] EL Zodrow and CJ Cleal, 'The epidermal structure of the carboniferous gymnosperm frond reticulopteris', *Palaeontology*, **36**(1), 65–79, (1993).