KNN algorithm with DTW distance for signature classification of wine leaves

José Luis Seixas Junior Tomáš Horváth

Department of Data Science and Engineering
ELTE – Eötvös Loránd University, Faculty of Informatics
http://t-labs.elte.hu/
3in Research Group, Martonvásár, Hungary
{jlseixasjr,tomas.horvath}@inf.elte.hu

Abstract: The European Union has created a way of classifying wines to make life easier for consumers when choosing the product that most appeals to them, this classification may require control that is hampered by the distancing of production. The automation of control processes is a good way out, but this comes up against the difficulty of computational methods for image interpretation. Thus, this paper presents a form of abstraction of image data into a series of values that can be more easily understood by the computer. A Machine Learning algorithm is also applied to create a baseline for the classification of these entry images, obtaining up to 60% accuracy while classifying five classes of vine varieties.

1 Introduction

In 2009, the European Union unified the wine classification system in order to make it easier for consumers to understand the European wine quality. Protected Designation of Origin (PDO) and Protected Geographical Indication (PGI) are types of wines which have some influence from their place of origin and may require control of different aspects of production [1], PDOs being the highest influenced and having highest production restriction [2]. A third class is Wine which has no restrictions or influence.

In Hungary, 97% of vineyards areas are eligible for protected zones PDOs or PGIs [3], making it hard to keep track of all production and varieties available.

Automating the plant identification process reduces the need for specialist professionals and, thus, makes it possible to increase the recognition and production capacity in each area. The automation of the processes requires the application of image processing techniques and the interpretation of the data obtained after these processes.

The "Protocol for distinctness, uniformity and stability tests" for grapevine [4] made by the European Community Plant Variety Office (CPVO) indicates that shapes can be an important feature for determining varieties, but unfortunately, complex shapes are characteristics that are difficult to detect and compare in image computational processes.

Ratanamahatana and Keogh [5] suggest that some problems can be transformed into a pseudo time series and even mention leaf shape as one of them. In their case, the leaves belonged to six different species, four species of maple and two species of oak.

So, the goal of this paper is to create a process that could transform a leaf image into a series based on its shape and use Artificial Intelligence or Machine Learning algorithms to create a baseline of object signature classification in grape varieties identification task.

This paper is organized as follow: Section 3 presents steps and techniques used to transform a leaf image into a series, followed by Section 4 which describes how these series were used to build a classification model. Section 5 brings the results obtained and in Section 6 the conclusions which can be inferred by these results.

2 Related Works

As in Ratanamahatana and Keogh [5], most of the works in the literature classify different species of plants, but the variations found in the differentiation of species are much higher than varieties differentiation which belongs to the same species *Vitis vinifera*.

Remagnino et al. [6] present margin patterns as a structure that can contribute to plant identification and mentions that few works cite this characteristic in the identification automation.

Stubendek & Karacs [7] describe techniques for creating object description vectors which count pixels of a region of interest in different directions, i.e., counting region pixels in a column for index i from x axis. The Extended Projected Principal Shape Edge Distribution counts directionally pixels belonging to the edge which exist in the direction of counting in four angles (axes x, y, main and secondary diagonals) and finally, concatenates the vectors of each direction for description by a single vector, however, rotation variant.

Munisami et al. [8] uses the kNN algorithm to cluster thirty two different plant species leaves, creating a vector of morphological features such as aspect ratio, area by perimeter ratio, perimeter ratio by smaller window, distance maps. Testing the feature vectors of all images with all samples, obtaining up to 100% accuracy for some classes and 83.5% general accuracy. This indicates that the kNN method can be an important ally for classification after obtaining the descriptive vectors.

Patil and Bhagat [9] bring us a review where the same methodology is applied with different techniques for leaf shapes process and recognition. The difference among datasets shows that for different types of leaves they need different approaches for classification, and it reflects the importance of shapes as a decision criterion.

In Du [10], we can see several shape descriptors based on binary images. For image binarization, conversion to gray scale was used with conversion weights as channel Y ($Gray = 0.299R + 0.578G + 0.114B$), from YIQ model, which represents brightness in analog imaging systems. The descriptors shown deal with major differences, but they are used to separate species, and not varieties, which have greater differences among them.

Aakif and Khan [11], on the other hand, uses a different intensity calculation that specifically fits leaves. It gives greater importance to green channel, since it is known that leaves tend to have high values on green, it was again used to differentiate species, background has less control than other articles, but with a unique color.

Diago et al. [12] also brings a study on leaves area estimation, but uses leaves as bush not individually and does not classify the type of grape, so it does not depend on smaller leaves details.

Convolutional Neural Networks (CNNs) have been used in all kinds of problems related to images and in viticulture it is no different, in addition to working in this sector, Ji and Wu [13] also cite several works that use these CNNs frameworks aimed at plant diseases identification. In our work, CNNs are avoids because these networks demand a much larger amount of images, computational power and time. Transfer learning, which is a proposed solution to such problems, is also not as efficient as models generally need to be tuned, as even mentioned by Ji and Wu.

3 Image to Series Transformation

Image acquisition process was done a long time before the beginning of the work, without the presence of an image specialist, so the process was very poorly controlled and there was no requirement for any type of special equipment. This caused the first impacting factor, as several of the images could not be used in this research due to excessive noise.

This happened due to the correct time for the plants to have leaves or be harvested, as this acquisition cannot be made at any time of the year by the climatic state that causes effects on production. Thus, the noisy images could not simply be reacquired at any time that was requested.

This lack of control was also designed to enable testing different approaches and after detecting the best possible process, later a new acquisition step can be done taking into account the relevant factors of this work when the appropriate time comes.

The number of images that could be really useful for work was also reduced, and this number was used to maintain the class balance. Thus, the number of images was thirty two per class, eight being used to create the classification and the rest for tests.

Two steps must be performed with the images, border identification, which is used in object description and reference point detection, which is used to measure the distances to the borders.

Figures 1 and 2 show the steps used in the two procedures, the figure representing the steps to find the outline of the object, while the figure shows the steps necessary to find the desired reference point in this work. Figures 1a and 2a are the same, since both algorithm has the same starting point.

Right after image reading, a resizing operation is carried out, there is no need to use such a high resolution as available in cell phone cameras nowadays, in addition, some algorithms that have size dependence may differ with different resolutions. Thus, resizing helps not only in execution speed, but also in standardizing values for all algorithms.

A test is done in image orientation, testing if the highest value is the width or height, to detect if the image is in portrait or landscape orientation. The highest value between them is used as a basis for reduction, where this value is now only 10% of its total value. The original images are 4000×3000 pixels, the reduction maintains the aspect ratio, thus, the bigger side is decreased to 400 and the smaller one consequently undergoes the necessary change to maintain the ratio.

At this point, two algorithms are applied, one that will find the reference point and the other that will find the border, so a copy is made of this resized image.

3.1 Contour Finder

To find the borders and avoid as much noise as possible, some preprocessing is applied. First, a Gaussian blurring filter is applied and then this result is subtracted from the original image.

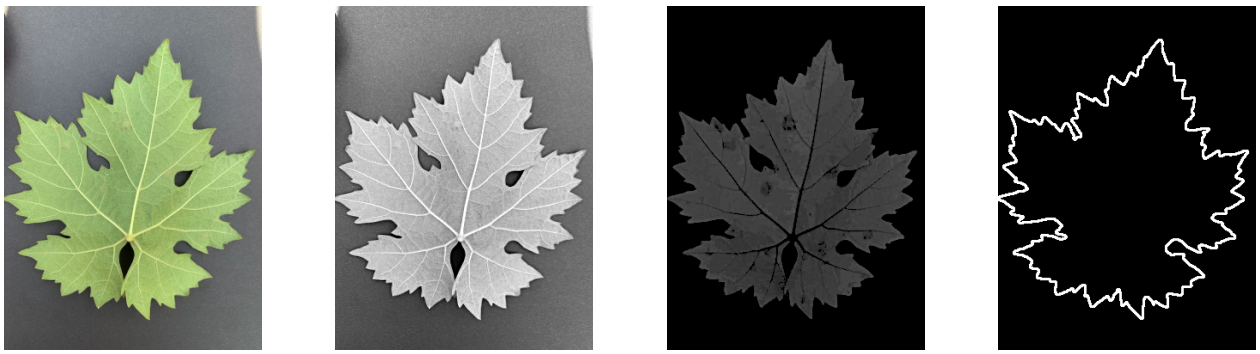
Due to the characteristic of the blur filter, plain areas do not have much alteration since plain regions have similar pixels and consequently the spread of similar information remains similar.

At the edges, this causes a difference, where, also by definition, the edges have drastic variations and therefore interactions with neighboring pixels will change the previous values. Thus, when subtracting values close to plains and more distant to the edges results in an edge sharpening filter, as can be seen in Figure 1b.

After edge sharpening, as known, leaves are greenish in color, a mathematical channel operation is performed using the green channel with double value and subtracting the red channel value.

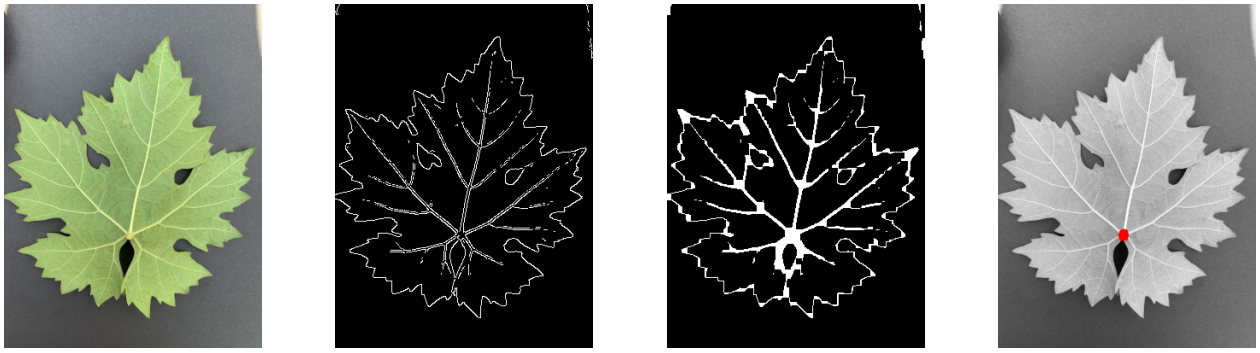
In the used images, the blue channel has low values for almost the entire image, thus, this channel was not very useful in the operation and no benefit was seen when used, thus, discarded.

Background area has closer values for red and green channels. However, within leaf areas there are distant val-



(a) Image example. (b) Sharpened Green Channel (c) Subtraction operation. (d) Detected contour

Figure 1: Contour finder procedure.



(a) Image example. (b) Canny edge detection (c) Closing operation (d) Point detected.

Figure 2: Point finder procedure

ues, with great intensities of green and very small (almost zero) for red, so the operation produces values close to zero in the background and values close to the green channel on the leaf area.

After this operation, the object is much more visible and the background practically tends to black, which greatly increases the ability of edge detection algorithm, so much that even image binarization was not necessary, in Figure 1c there is example of the result of this mathematical operation and how the object is able to stand out from the background, which at this moment, becomes black.

The contour detection algorithm used was the algorithm from OpenCV library, all edge points were stored, no approximation was made and only the most external object was used. The result of the contour detection can be seen in Figure 1d.

3.2 Reference Point

The copy of resized image is used in this process, so none of the processes mentioned in the contour detection, described in Subsection 3.1, will impact at this point.

Tak and Hwang [14] use a procedure that allows the margin and its patterns to be taken into account, using the distances to a reference point. Their work uses the center point of the object as a reference, so they calculate the distance from all points on the margin to the center of the object, as it is rotation and scale invariant.

These characteristics are important and are maintained in this work, however, by definition, the center of the image is the average distance over the margin values, which will flatten the different distances. This approximation of values tends to make it more difficult to solve the problem by the models.

So different steps were taken to find the reference point. For this task, it is assumed that the petiole region is a region with a lot of information.

The petiole is the stalk which bind the leaf to the steam, its region is the part of the leaf which contains the cut and it is full of thick ribs, many edges and a small circle may appear in some cases. Thus, since this region has a junction of many parts, it also contains many edges, therefore, the first step in this process is to run an edge detection filter.

To prevent many details from being highlighted in the middle of the leaf, the Canny algorithm was used with relatively high values, 200 and 250 for minimum and maximum detection parameters, respectively, the result is as shown in Figure 2b.

At the petiole region, which is the region sought, many edges are detected, following the edge detection, a closing operation with a rectangular shape and size 7×7 is

executed. Close lines will therefore be connected, small holes filled and as mentioned, this region has many lines, it becomes a very highlighted region, i.e., it becomes a big blob or structure, as seen in Figure 2c.

At this point, the resizing previously mentioned gains huge importance, since with no image size adjusting, kernels in this type of algorithm could vary greatly in images obtained from different cameras.

After that, several erosion operations is carried out, with a cross structural element size 3×3. Any noise or small region is eliminated. The erosion operation takes place until a new execution would clear the image, i.e., the operation is performed until if performed once more, everything would be eliminated.

Therefore, the resultant is the smallest possible portion of the largest portion found after the closing operation. With these operations, the remaining point is only the one which belonged to the largest region, which is the petiole region due to its characteristics. The (x, y) coordination of this point is stored. In Figure 2d, the point found was painted over the image for a better visualization of precise position found after the subsequent erosion were made.

3.3 Signature Creation

With the object contour points and a reference point, it is possible to create a object signature. The signature is a vector with distances of each contour point in relation to the reference point.

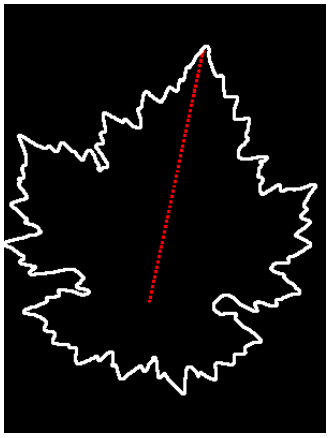


Figure 3: Distance.

The dashed red line in Figure 3 is an example of distance to be put in the series. Distances from the reference point to all points belonging to contour are calculated and put together following the order according to the return of the operation by OpenCV.

As the contour points are placed according to your neighborhood, there are no jumps and the output is the description of the shape of the object in vector format. Last but not least, the vector is normalized, i.e., all values were divided by its highest value.

However, as the algorithm starts with the upper left point on the contour, if the object rotates, the starting point is shifted, as can be seen in Figure 4. Because of this, two datasets were created, in the first, the process is as described above. In the second, the vector was rolled so that its highest value was the first index of the vector. Therefore, all vectors in this case start with one, since they have been normalized.

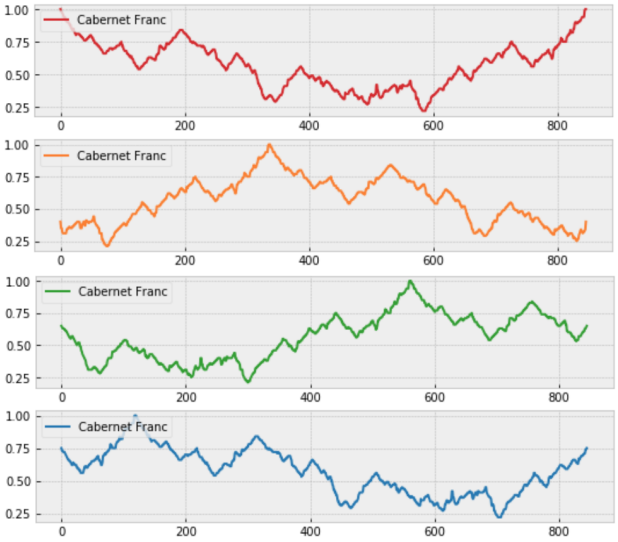


Figure 4: Series from different angles.

This was done because the same leaf would produce quite different values and, consequently, larger distances between values that should be very close.

For example, in Figure 4, all series belong to the same leaf, the one previously used as example, and rotated 90°, 180°and 270°, respectively. But measuring distances between its original and after applying rotation, the results are approximately 211, 234 and 143, respectively. With the vector rotation technique, all of them are represented as the first example, as this has its greatest value in the first position, and the distance between them goes to zero.

The unchanged vector was still maintained, since some small variations in the vector would be interpreted by the distance calculation algorithm, so comparisons would be made on the necessity for the rotations.

Even though rotated vector adjusts variations in angulation, it also approximates values, as it is possible to see when realizing all vectors start with the maximum value, one. So, the experiments were done with both types of vectors to check if the gain of the process is greater than the damage caused by the approximation of the values.

4 Object Classification

After the process of creating object signatures, they need to be classified among the available groups. Our research included five different types of vine leaves:

- Cabernet Franc
- Blaufränkisch (Kékfrankos)
- Muscat Blanc à Petits Grains (Sárgamuskotály)
- Pinot gris (Szürkebarát)

- Gewürztraminer (Tramini)

So, the task is a multiclass series classification from object signature generated by vine leaves contours.

4.1 k-Nearest Neighbors

The classification was made using the k-Nearest Neighbors (kNN) algorithm. The kNN is a distance-based sorting algorithm, where the object to be classified receives the class of most of the objects closest to it. The kNN algorithm was chosen because it does not need many examples, since there not so many images.

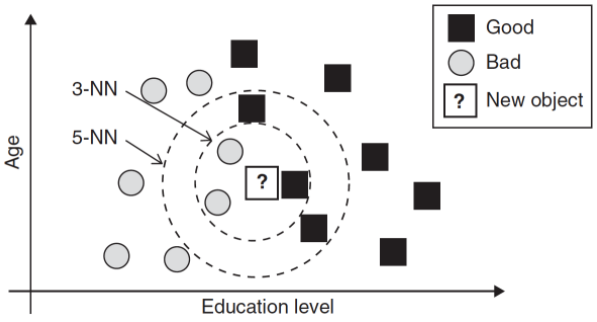


Figure 5: kNN example [15].

Figure 5 shows an example of kNN operation where the new object class is assigned as the same as the most numerous class of its neighbors. The value of k determines how many objects are taken into account when assigning the new object's class, so if k equals 1, the object to be classified receives the same class as the closest object, which may be interesting because the object would be classified according to the value most similar to it, but it makes the model more susceptible to noise.

kNN can be used with any distance measurement, it is commonly found with Euclidean distance as it is a distance calculation easy to understand and with good accuracy. However, Euclidean distance needs vectors of equal sizes for its calculation, in addition, small flaws in the detection of contour, or any image noise, such as shadows or reflections, close to the edges could cause variations in distance positions of the calculated vector.

Thus, in order to avoid problems in the calculation of distances due to small inaccuracies and still eliminating the problem of the vectors sizes, the distance calculation method used to measure the series distances was using the Dynamic Time Warping algorithm.

4.2 Dynamic Time Warping

Dynamic Time Warping (DTW) is a technique for finding the alignment of two time series even if one of them has been deformed by shortening or stretching the time axis, the algorithm finds the minimum distance of the series deformation path [5].

In Cope and Remagnino [16], they show this technique can be used to compare leaves using their margins, however, in their case, plants belong to different species that make the distances and differences in the series more accentuated.

To make the algorithm more efficient, its fastest form was used, named fastDTW introduced by Salvador and Chan [17]. In this approach the warp path is found first in a very low level resolution of the series, then projected and refined into higher resolution levels until full resolution, decreasing the DTW's complexity from $O(N^2)$ to $O(N)$.

5 Results

The kNN algorithm was executed with k values from 1 to 7, the best result in all executions was with value 3, as can be seen in Figure 6. Where the dashed blue line represents the accuracy of non-rolled vectors and the red full line shows the accuracy of rolled vectors using different values for k in the kNN algorithm.

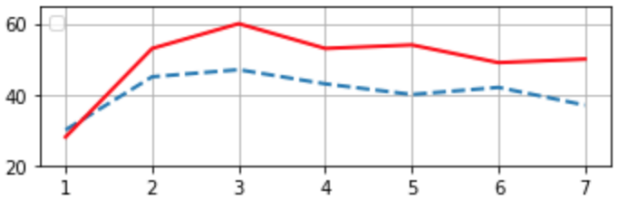


Figure 6: Accuracy varying k.

Non-rolled series reached approximately 44% accuracy, while 60% for rolled vectors for all five classes. When analyzing classes individually, Kékfrankos was the most accurate with 80% accuracy, while Cabernet Franc got just over 43%.

These individual results were obtained with the value of k as 3, for being the best for all experiments. The same values of k were applied for individual classes measurements, but the results of the best and worst varieties were always the same, changing only the percentages.

As a comparison, random checks were performed, ten attempts were made with a maximum accuracy of 25% and an average of 20% for the five classes.

One of the reasons for the low accuracy of the kNN model was the noise caused in the detection of contours that united the grooves in the middle of the leaf with the bottom, as this included contour values that were not leaf margins, besides, they are values with more significant differences than those belonging to margin patterns.

In Figure 7 there is an example where the internal grooves of the image were detected as margins. The first problem is that the reference point, in this case, is not part of the interior of the region, which should not happen. The second problem is from contour detection which detects contours on the inside of the leaf, creating patterns that are not from margins, but will be part of the series.

Figure 7: Image noise.

The application of morphological filters was not considered, as these filters would also alter the shape of the outermost layer of the leaf and it would not be interesting in this case, because we are considering the shape as the criterion for classification. Thus, morphological filters would have the same disadvantage of using the center of the image as the reference point.

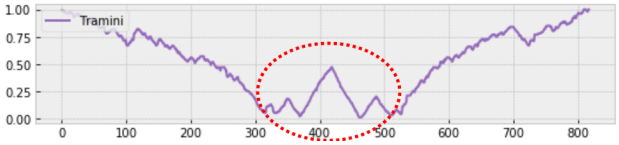


Figure 8: Noisy series.

In Figure 8 shows the variations caused by noise in the contour detection. Since those inside lines does not have any margin pattern, they appear straighter than outside lines and with higher changes from surrounding values.

In Cabernet Franc, there is another problem that may have caused problems in the series and may be responsible for the poor accuracy result of this type. The lack of control of the acquisition and the fact that the leaf is not flat allows some places on the edges to touch each other.

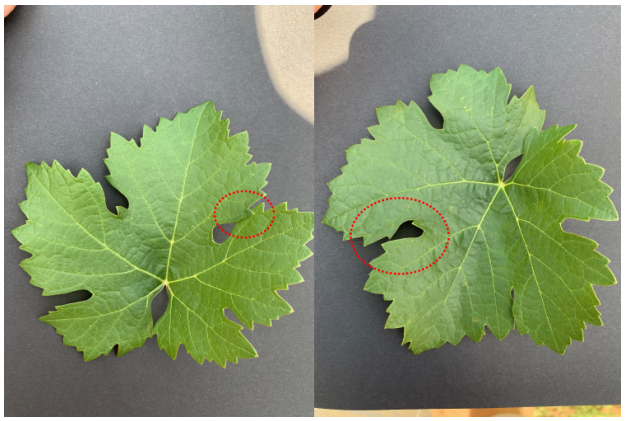


Figure 9: Touching margins.

In Figure 9, there are two examples of Cabernet Franc leaves where the margin touches itself. As highlighted by the red dashed ellipses, one of them touches itself and the other does not, so it is not possible to be sure this will always happen and it will be part of the series or that it would never happen.

Tramini was responsible for the use of few images, because in this variety we had thirthy two usable images, eight of which were used for kNN training and the rest to obtain the results. All other classes had more images, but were not used to maintain a balance between classes.

The lack of images for training was one of the reasons for choosing kNN for classification, as this algorithm does not require many examples to produce considerable results, but which can be improved if there are more examples for training and testing.

6 Conclusions

In this article, a process for transforming an image of a leaf into a series of values was presented. This series can be saved, analyzed or compared easier than a shape inside an image.

In addition, the kNN clustering model was used to try to classify the different types of leaves through these series. This first attempt was chosen with the intention of creating a baseline and studies of models or improvement of this model can help in a more correct recognition of leaves series.

It was also seen that the rolling process on the vectors helps the algorithm to compare the sequences more correctly, because this process makes the series rotation invariant, and applied with the normalization, also makes it scale invariant.

Two processes may be performed in future works, one of them on the preprocessing algorithm for a better detection of the contour, which can use segmentation of the area of interest, or the study of models that can work better with the series already created.

Acknowledgement

We would like to thank Telekom who has us as one of its technology partners on Telekom Innovation Laboratories and the Tempus Public Foundation for the financial support through the Stipendium Hungaricum Scholarship Programme.

The research has been supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.2-16-2017-00013, Thematic Fundamental Research Collaborations Grounding Innovation in Informatics and Infocommunications).

References

[1] Machiel J. Reinders, Marija Banovic, and Luis Guerrero. Chapter 1 - introduction. In Charis M. Galanakis, editor, *Innovations in Traditional Foods*, pages 1 – 26. Woodhead Publishing, 2019.

[2] M.J. Martelo-Vidal and M. Vázquez. 3 - advances in ultraviolet and visible light spectroscopy for food authenticity testing. In Gerard Downey, editor, *Advances in Food Authenticity Testing*, Woodhead Publishing Series in Food Science, Technology and Nutrition, pages 35 – 70. Woodhead Publishing, 2016.

[3] Agrosynergie EEIG. Evaluation of the cap measures applied to the wine sector. Agricultural policy 10.2762/79919, Directorate-General for Agriculture and Rural Development (European Commission), 2018.

[4] Protocol for distinctness, uniformity and stability tests. `cpvo.europa.eu/sites/default/files/documents/vitis_2.pdf`, 2009. Accessed: 2020-06-10.

[5] Chotirat Ann Ratanamahatana and Eamonn Keogh. Everything you know about dynamic time warping is wrong. In *Third Workshop on Mining Temporal and Sequential Data*. Citeseer, 2004.

[6] Paolo Remagnino, Simon Mayo, Paul Wilkin, James Cope, and Don Kirkup. *Computational Botany*. 01 2017.

[7] Attila Stubendek and Kristóf Karacs. Shape recognition based on projected edges and global statistical features. *Mathematical Problems in Engineering*, 2018:1–18, 04 2018.

[8] Trishen Munisami, Mahess Ramsurn, Somveer Kishnah, and Sameerchand Pudaruth. Plant leaf recognition using shape features and colour histogram with k-nearest neighbour classifiers. *Procedia Computer Science*, 58:740 – 747, 2015. Second International Symposium on Computer Vision and the Internet (VisionNet'15).

[9] Akshay Patil and Kanchan Bhagat. Plants identification by leaf shape recognition: A review. *International Journal of Engineering Trends and Technology*, 35:359–361, 05 2016.

[10] Ji-Xiang Du, Xiao-Feng Wang, and Guo-Jun Zhang. Leaf shape based plant species recognition. *Appl. Math. Comput.*, 185(2):883–893, February 2007.

[11] Aimen Aakif and Muhammad Khan. Automatic classification of plants based on their leaves. *Biosystems Engineering*, 139:66–75, 11 2015.

[12] Maria-Paz Diago, Christian Correa, Borja Millán, Pilar Barreiro, Constantino Valero, and Javier Tardaguila. Grapevine yield and leaf area estimation using supervised classification methodology on rgb images taken under field conditions. *Sensors*, 12(12):16988–17006, Dec 2012.

[13] Miaomiao Ji, Lei Zhang, and Qiufeng Wu. Automatic grape leaf diseases identification via unitedmodel based on multiple convolutional neural networks. *Information Processing in Agriculture*, 2019.

[14] Yoon-Sik Tak and Eenjun Hwang. A leaf image retrieval scheme based on partial dynamic time warping and two-level filtering. In *7th IEEE International Conference on Computer and Information Technology (CIT 2007)*, pages 633–638, 2007.

[15] J. Moreira, A. Carvalho, and T. Horvath. *A General Introduction to Data Analytics*. Wiley, 2018.

[16] James S. Cope and Paolo Remagnino. Classifying plant leaves from their margins using dynamic time warping. In Jacques Blanc-Talon, Wilfried Philips, Dan Popescu, Paul Scheunders, and Pavel Zemčík, editors, *Advanced Concepts for Intelligent Vision Systems*, pages 258–267, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

[17] Stan Salvador and Philip Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580, October 2007.