Plant Seedlings Classification

Dmitri Jakovlev, Julia Kamaletdinova, and Georgy Shevlyakov

Peter the Great Saint-Petersburg Polytechnic University, Department of Applied Mathematics, Russia

Abstract. The abstract should briefly summarize the contents of the paper in 150–250 words.

Keywords: First keyword · Second keyword · Another keyword.

1 Introduction

The demand for agricultural products is increasing day by day, as the population of the Earth is growing. Even though people are working on plant classification algorithms, approaches are still not as robust as desired. A significant part of work has still been done by people. The question arises of the efficiency with which human resources are used. We will use exhaustible natural resources wisely and increase harvests if we automatise quality assurance, which objectives are to detect and distinguish weeds among the variety of crop seedlings.

All this naturally leads to idea of automation of the classification process with help of machine learning algorithms. From recent experience, neural networks are well suited for image processing, but we have to pay for it with computational costs. On the other hand, we could use less costly algorithms, but they require finer tuning to achieve a comparable result.

The goal is to implement segmentation and classification of a specific type of data set for low time and computational complexity. In this paper we will research binary classifiers capabilities on the dataset [2] consisting of images of 12 species and containing the most common weed species in Danish agriculture.

2 Plant Seedlings Classification

2.1 Data

The dataset is a part of the database have been recorded at Aarhus University Flakkebjerg Research station in a collaboration between University of Southern Denmark and Aarhus University. Images are avaliable to researches at https://vision.eng.au.dk/plant-seedlings-dataset/. The specific of the dataset is that recorded plants are in different growth stages since detecting weed in it's early stage is the thing makes the task problematic.

The dataset contains 960 unique plant images of 12 species. The sizes of plant classes are not balanced among themselves - they range from 221 to 654

labeled samples for each class. Original images are cropped by plant boundaries, but their resolutions varyes from 50x50px to 2000x2000px. Also, images have a different background - some of them on the ground, other on the marked paper.

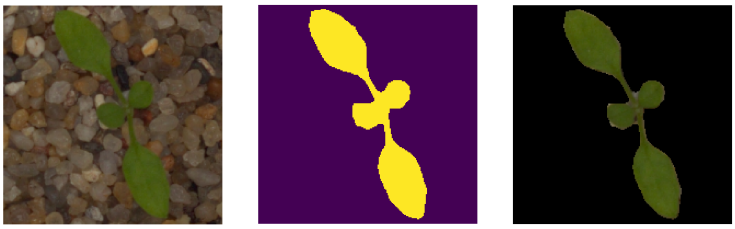
Fig. 1. Data overview

2.2 Data preprocessing

We have three stages of data preprocessing.

Firstly, we reduce resolution of all images to the same resolution 200x200px using bilinear interpolation. The main idea of bilinear interpolation that the new image pixel is the weighted sum of neighboring pixels of the original image. It helps to decrease computational complexity and build normalized features.

Secondly, the objects of our study are plants and they are painted green. Therefore, we can create a mask that filters the range of green channel and ignores the other pixels. For these purposes the HSV (Hue Saturation Value) color model is suitable representation. In the BGR format, the value of each component depends on the amount of light hitting the object. HSV allows us to distinguish between image color and brightness. Hue, saturation and value let us set the lower and upper borders of the shades of a certain color, in this case — green. Then we merely mark the pixels in the green range and get a color mask (Fig. 2(b)). Now we apply the operation of logical multiplication to the original image, assign the value of the background pixels to a black value and get a segmented plant.



(a) (b) (c)

Fig. 2. (a) Source image; (b) Mask; (c) Segmented image

Thirdly,

2.3 Feature selection?

Features of the images define their content. We recognise the information images provide us with taking into account a great number of features. Then we answer what do we see exactly. The same process can be projected on image classification task: image features let the classifier propose the output decision. Another advantage of the approach is that it reduces feature space for a machine learning algorithm. We often need only a part of the information image is carrying, hence we don't need to process and interpret all the pixels, what can lead to extra computational expences.

Selecting features is a complicated and convoluted research area itself, the assertion if supported by the variety of feature types and the need of presenting essential properties on the equal basis with the previous assertion.

As dicucussed before, we need to define the set of features describing the dataset in the best way. Supposed features must meet the following criterion:

- The feature space should be low-dimensional
- The features should not be correlated or be correlated as less as possible
- Selected features should represent the content of an image as fully as possible

We are going to group selected features and define them.

2.4 Color features

Overviewing the dataset we can notice that all the plant species are mostly green. Additionally, images were recorded under specific conditions. We will use RGB color model, which stands for red, green and blue colors, and calculate features described below.

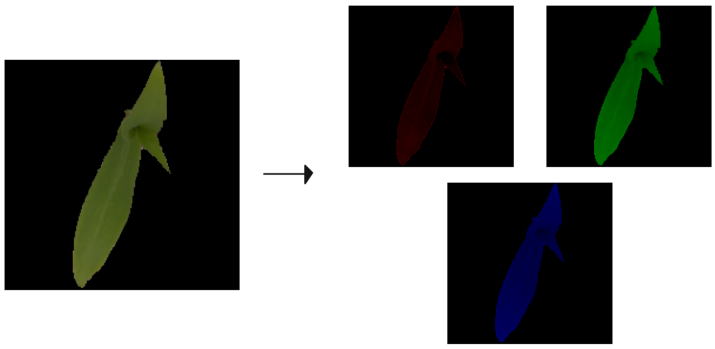


Fig. 3. RGB transformation

Let $\{x^{(k)}\}_{i=1}^{N}$, where $k = 1, 2, 3$ — an index of a channel in RGB color space respectively, N — total number of the image pixels, $x_i^{(k)}$ — i-th pixel of the k-th channel. We will compute sample mean and standard deviation for each channel:

$$\overline{x^{(k)}} = \frac{1}{N} \sum_{i=1}^{N} x_i^{(k)} \tag{1}$$

$$s^{(k)} = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x_i^{(k)} - \overline{x^{(k)}})^2} \tag{2}$$

2.5 Shape features

Total perimeter

In this feature we will count the sum of perimeters of all the areas bounded by contours:

$$P = \sum_{i=1}^{K} p_i, \tag{3}$$

where p_i — i-th perimeter, K — number detected bounding contours

Total area

It includes all the areas bounded by contours:

Maximal contour area

Number of bounding contours

Segmentation output may result in image divided in separate parts of the plant. We decided to use it and compute the number of bounding contours. We will use the boundary tracing algorithm for the boundary extraction. The

designated algorithm [3] was implemented in OpenCV [1] library for the Python programming language.

Rectangularity

Circularity

2.6 Classification

3 Results

4 Discussion

5 Conclusion

6 Acknowledgments?

References

1. Bradski, G.: The opencv library. Dr. Dobb's Journal of Software Tools (2000)
2. Giselsson, T.M., Jørgensen, R.N., Jensen, P.K., Dyrmann, M., Midtiby, H.S.: A public image database for benchmark of plant seedling classification algorithms (2017)
3. Suzuki, S., Abe, K.: Topological structural analysis of digitized binary images by border following. Computer Vision, Graphics, and Image Processing **30**(1), 32–46 (1985). https://doi.org/10.1016/0734-189X(85)90016-7, https://doi.org/10.1016/0734-189X(85)90016-7