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 [3] 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

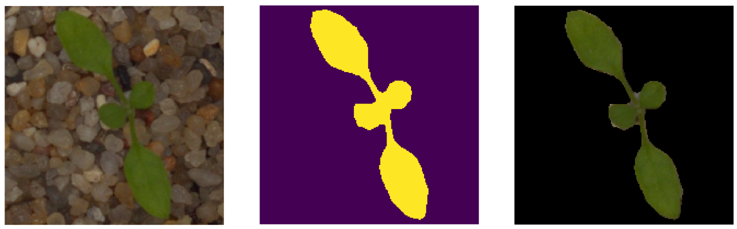
2.2 Data preprocessing

Resolution reducing

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.

Segmentation

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

Noise removal

Segmentation does not always work well (Fig. 2(c)). Small areas of the background may fall into the range of green values, which causes distortion of the binary mask (Fig. 3(a)).

(a) Before morphological closure

(b) After morphological closure

Fig. 3. Segmentation improvement example

Such drawbacks can be eliminated with the morphological operations — nonlinear transformations associated with the shape and structure of image. The morphology is used to study the interaction of an image with a specific structural element — the kernel. The kernel iterates over the entire image and compares with the neighborhood of the pixels, after which morphological operations are applied [2].

To improve segmentation, we use the operation of morphological closure — a combination of dilatation (Fig. 4) and erosion (Fig. 5) operations.

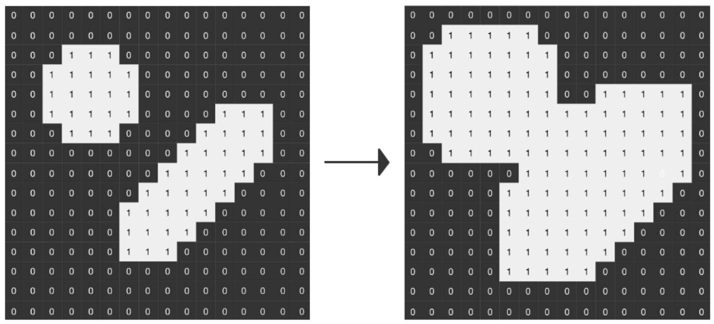


Fig. 4. 3x3 square core dilation

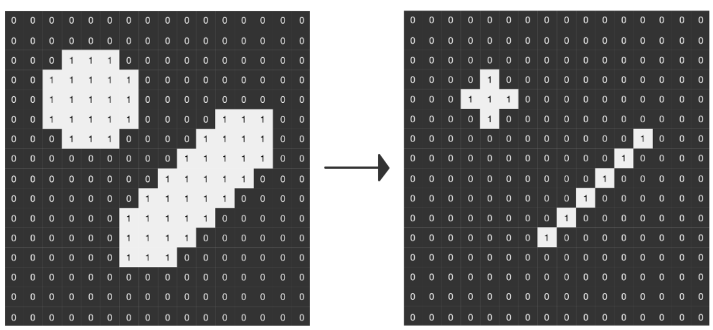
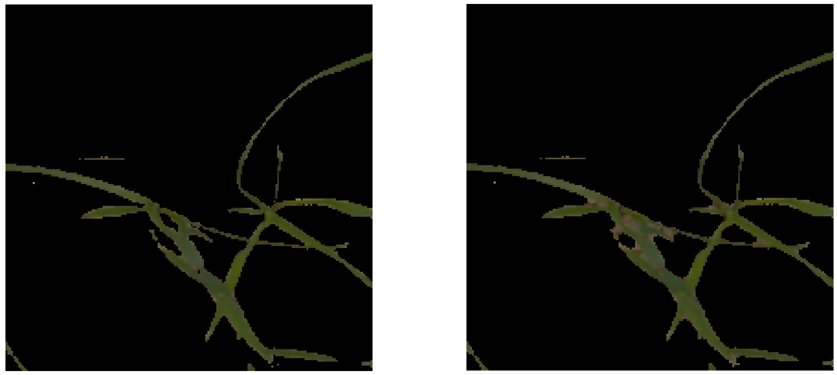


Fig. 5. 3x3 square core erosion

Apply the close operation to image (Fig. 6(a)) by selecting an elliptical core of 6x6px size and delete the remaining objects with an area of less than 160px. The plant in image (Fig. 6(b)) has no cavities, and the background is cleared of non-plant elements. But morphological closure does not always have a clear effect on images. Consider the result of working on an image of the class Loose silky-bent. It is noticeable (Fig. 6) that the cavities corresponding to the background were restored, which does not correspond to the desired result. We restrict ourselves to the removal of objects whose contours limit a small area.



(a) Before morphological closure (b) After morphological closure

Fig. 6. Segmentation degradation example

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.

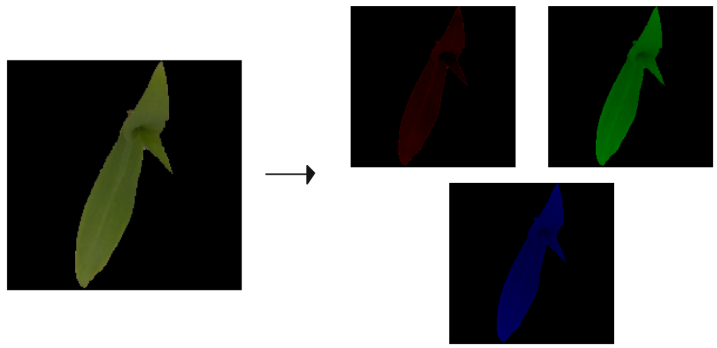


Fig. 7. RGB transformation

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

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 [5] was implemented in OpenCV [1] library for the Python programming language. The studies did not take into account contours bounding areas below a certain threshold, which was empirically chosen.

Let K be the number of detected bounding contours above the threshold in the further contour-related characteristics.

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

Total area

It includes all the areas bounded by contours:

$$S = \sum_{i=1}^{K} s_i, \tag{4}$$

where s_i — i-th area

Maximal contour area

We will be analysing the contours bounded maximal areas:

$$S_m = \max s_i,\ i = 1, \ldots, K, \tag{5}$$

where s_i — i-th area

Rectangularity

One of the methods to estimate rectangularity is to plot minimum bounding rectangle...

Circularity

Another title of this shape factor is the isoperimetric quotient and it shows how much area pe perimeter is bounding:

$$f_{circ} = \frac{4\pi A}{P^2}, \tag{6}$$

where P — total perimeter, A — total area of all detected elements of a plant

Consider the correlation matrix of the described features:

Based on the data in the figure 8, we conclude that the most linearly dependent features are the total area and the largest area, but this is not true for all classes due to the predominance of plants, limited by the onlyone contour, so the consideration of the maximum area feature is not excluded.

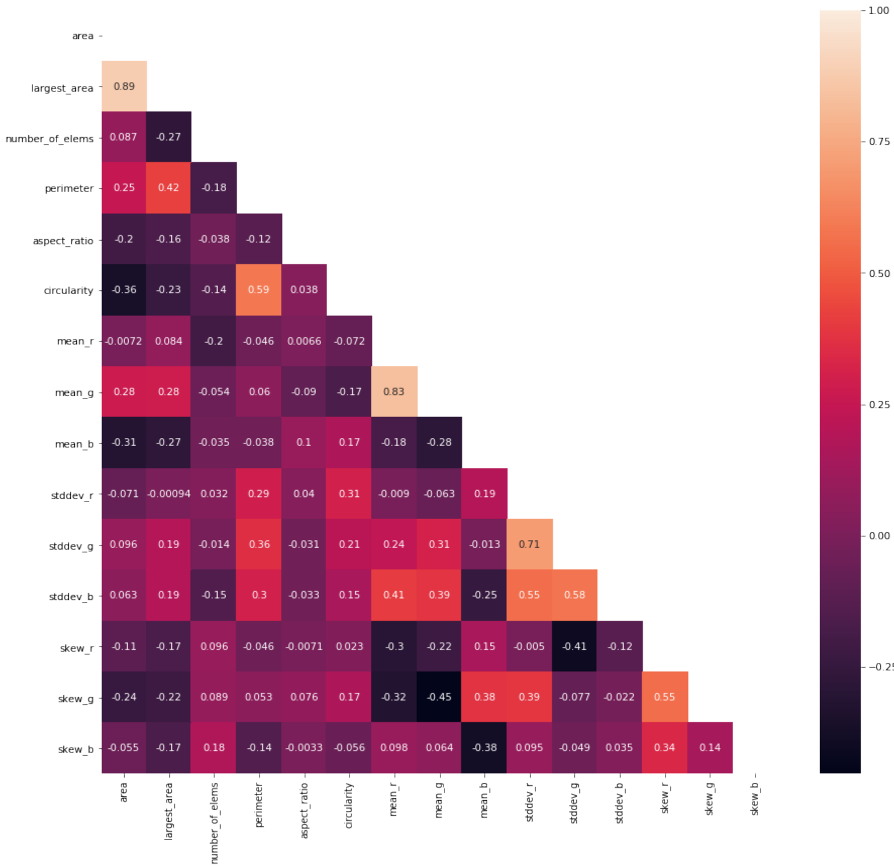


Fig. 8. Feature correlation matrix

2.6 Classification

The main method for this task is Support Vector Machine (SVM) — a binary classification algorithm based on building a separating hyperplane. The algorithm is implemented in Scikit-learn ([4]) library for the Python programming language.

We will use the Radial Basis Function (RBF) as the kernel function for SVM. The choice was made based on the following advantages:

- RBF usage allows to build a hyperplane when the data is not linearly separable
- Only one parameter that affects the resulting solution needs to be tuned
- Values of the RBF function are in $(0, 1]$, which does not include zero or infinity

The SVM algorithm is sensitive to non-normalized data, especially when using the RBF kernel, which is the Euclidian distance itself. In the case when the feature values are in different intervals, a slight difference in one of them can lead to going out of range in second feature values. The solution is to map all the values into one segment, in this task we will choose $[0, 1]$.

3 Results

Results of classification are evaluated on micro-averaged F-score. Given positive and negative rates for each class, the resulting score is computed the following way:

$$Precision_{micro} = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FP_k}, \quad Recall_{micro} = \frac{\sum_{k \in C} TP_k}{\sum_{k \in C} TP_k + FN_k} \tag{7}$$

$$F_{micro} = \frac{2 * Precision_{micro} * Recall_{micro}}{Precision_{micro} + Recall_{micro}}, \tag{8}$$

where C — set of the plants classes

The choice of such a metric is justified by the fact that classes are not balanced in terms of data volume, and in this case, the influence of classes decreases due to averaging by classification characteristics, not by F-scores.

The classificator output is shown on 9

4 Discussion

5 Conclusion

6 Acknowledgments?

References

1. Bradski, G.: The opencv library. Dr. Dobb's Journal of Software Tools (2000)
2. Friel, J.J.: Practical Guide to Image Analysis, vol. 1. ASM International (2000)
3. 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)
4. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in python. J. Mach. Learn. Res. **12** (2011), http://dl.acm.org/citation.cfm?id=1953048.2078195
5. 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

	precision	recall	f-score
Sugar beet	0.91	0.98	0.94
Fat Hen	0.91	0.94	0.92
Scentless Mayweed	0.86	0.87	0.87
Charlock	0.89	0.94	0.96
Small-flowered Cranesbill	0.95	0.97	0.96
Maize	0.92	0.90	0.91
Shepherds Purse	0.81	0.73	0.77
Common wheat	0.85	0.85	0.85
Common Chickweed	0.92	0.92	0.92
Cleavers	0.87	0.82	0.84
Loose Silky-bent	0.88	0.82	0.85
Black-grass	0.72	0.70	0.71
micro F-score	0.89		

Fig. 9. Metric by class ant total metric