Flower detection using features

Computer Vision, UPC

Xavier Martín Ballesteros and Adrià Cabeza Sant'Anna
UNIVERSITAT POLITÈCNICA DE CATALUNYA

June 8, 2019

Contents

1 Introduction **3**

2 Descriptors **3**

2.1 Compactness 3
2.2 Color Histogram 3
2.3 Number of petals 4
2.4 Relative position of the centroid 5
2.5 Hoggs form: Orientation 6
2.6 Fourier descriptor: Shape 6

3 Classifier **6**

4 Data augmentation **6**

1 Introduction

The aim of this assignment is to classify 12 different types of flowers using feature extraction.

We have first implemented several ways to extract features that we believed could be key in order to classify correctly the flowers, we tested them, and using different classifiers (decisions trees, SVM or random forest) have tested the accuracy, specificity or sensitivity.

2 Descriptors

In the following sections we will introduce different descriptors we have used to find those valuable features.

2.1 Compactness

2.2 Color Histogram

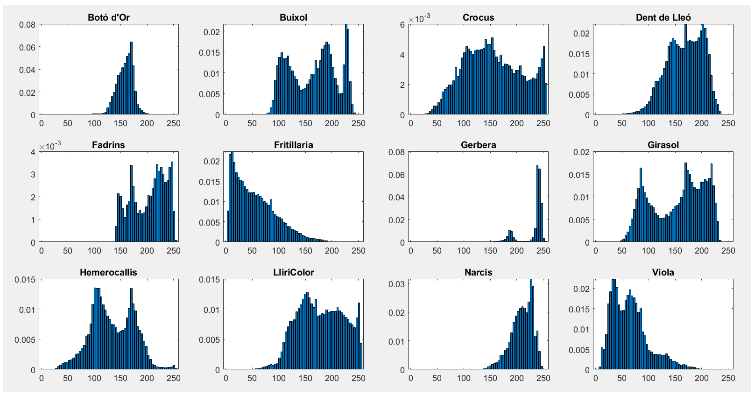


Figure 1: Prova1.

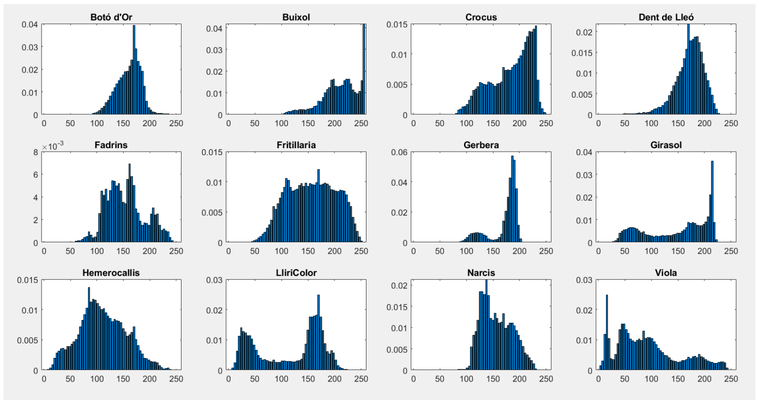


Figure 2: Prova2.

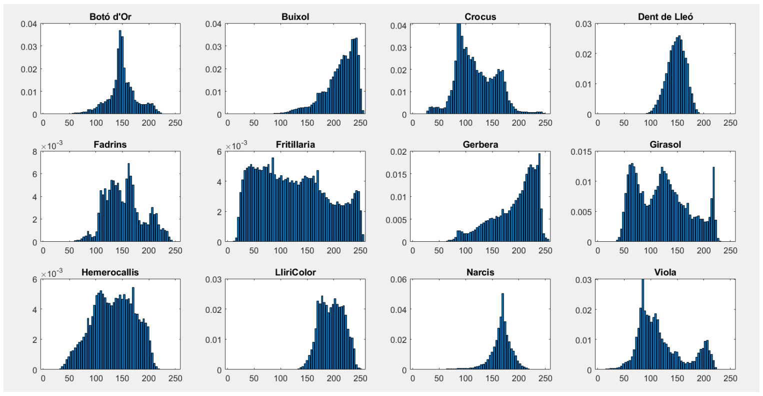


Figure 3: Prova3.

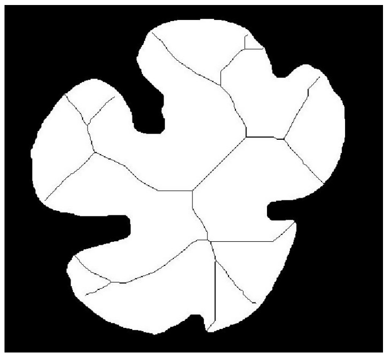
2.3 Number of petals

This descriptor was one of the most difficult to think of. It can be done in several ways so we first had to choose in which way we wanted to attack the problem. In our case we decided to skiz the flower.

To do so we used *bwmorph(image, 'skel', inf)* from Matlab. This creates the skeleton iterating as many times as necessary in order to not see any change between iterations (until it converges). Then we observed that sometimes the skiz did not touch the contour of the flower so we could not really count the petals; to solve it we applied an small erosion. The value of the disk structure was chosen based on several trials.

In this example using *Boto d'Or* we get as a result 5 petals:

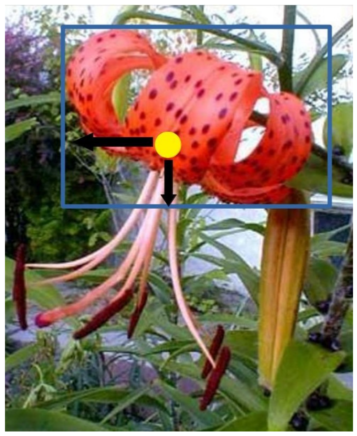
(a) Original

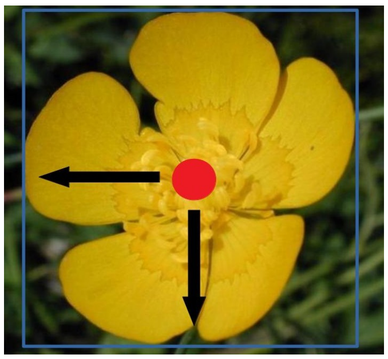

(b) Skez of the image

2.4 Relative position of the centroid

Just evaluating the position of the centroid would have been useless, it is not a feature of the flower per se. It can vary depending on the angle of the photo or the distance. However, if we calculate the relative position we can get really powerful information.

This two flowers, for example, should have really different values for this descriptor and it could be key in order to differenciate them in our classifier:


(a) Centroid and bounding box of a *Hemerocallis*


(b) Centroid and bounding box of a *Girasol*

The way our descriptor is implemented is the following: we get the first and last pixels of X and Y axis in order to create the **bounding box**, then using regions props we get the **centroid of the flower** and finally we calculate the **relative position** of the centroid over the bounding box.

In this example using *TBD* we get as a result X Y:

Figure 6: Centroid and bounding box output of a *TBD*

2.5 Hoggs form: Orientation

Available at the *Matlab Computer Vision Toolbox*

2.6 Fourier descriptor: Shape

3 Classifier

4 Data augmentation

In order **to strenghten our descriptors** we have also used the following public repository: *Albumentation: fast image augmentation library and easy to use wrapper around other libraries*(see AFEGIR NUM in References for more), which proporcionates facilities to augment the dataset including several transformations like: flipping, blurrring, RGB Shifting, Random contrast, Random brightness, etc...

To decide which transformations we wanted to apply, we looked first to our descriptors and the importance of each feature. For example, we discarted the Channel Shuffle because we believe that the color is very important for our implementation. Then, based on trial and error, we picked some of them which gave us an overall better result. Finally we are using:

- polla
- polla
- polla

References

[1] A. Buslaev (2018). Based on *Albumentations: fast and flexible image augmentations* [online]. Paper available at:https://arxiv.org/abs/1809.06839. Code available at:https://github.com/albu/albumentations [Accessed 3 June. 2019].