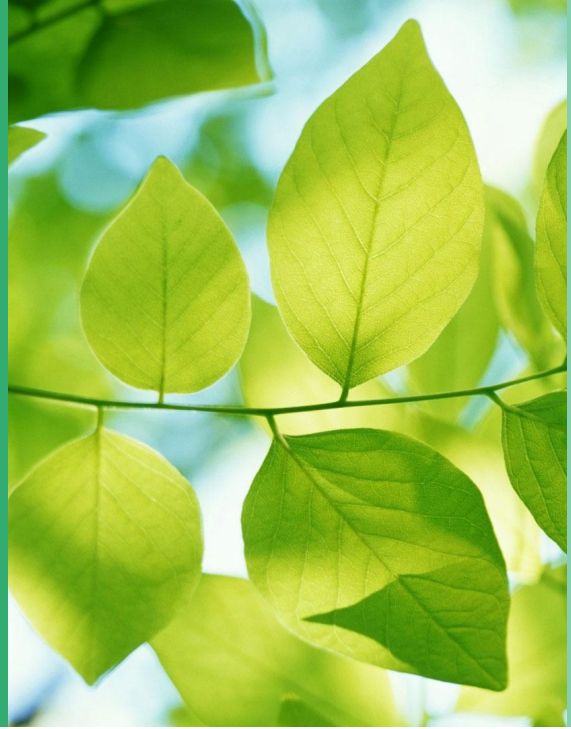
Leaves Classification

Project EduPARK

Computer Vision



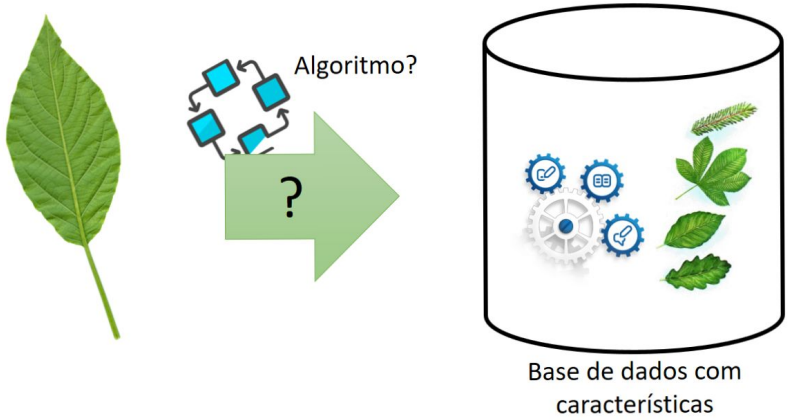
Rui Oliveira (68779) & Tomás Rodrigues (68129)
Teachers António Neves and Paulo Dias
Group 9

11 January 2017

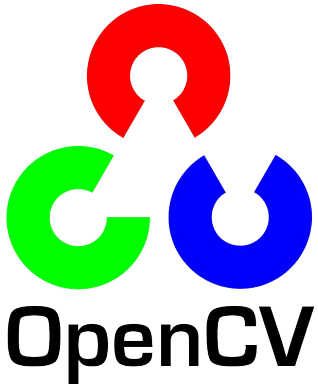
Contents

Main Goals

- Receiving a picture of a leaf predict its species name
- Test multiple features to compare performance and have a perception of what the best features to work with
- Train our own dataset with a classifier



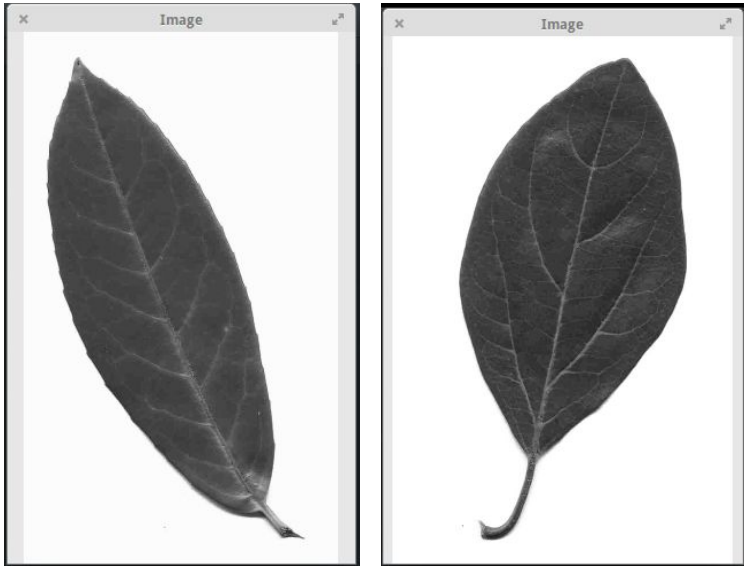
Technologies



Segmentation and processing (I)

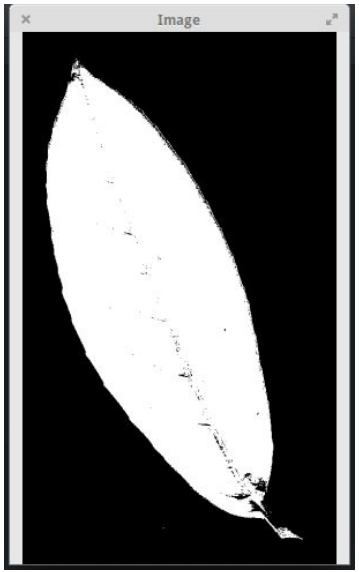
- Necessary procedure to **minimize noise**
- **Objective**: to obtain leaf contour
- **Example input**: two leaves

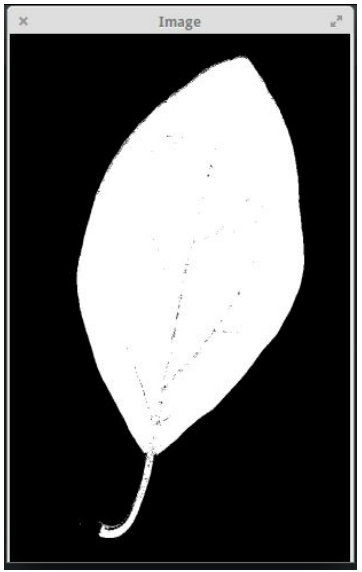
Segmentation and processing (II)



- Necessary procedure to **minimize noise**
- **Objective**: to obtain leaf contour
- **Example input**: two leaves
- Convert to **grayscale**

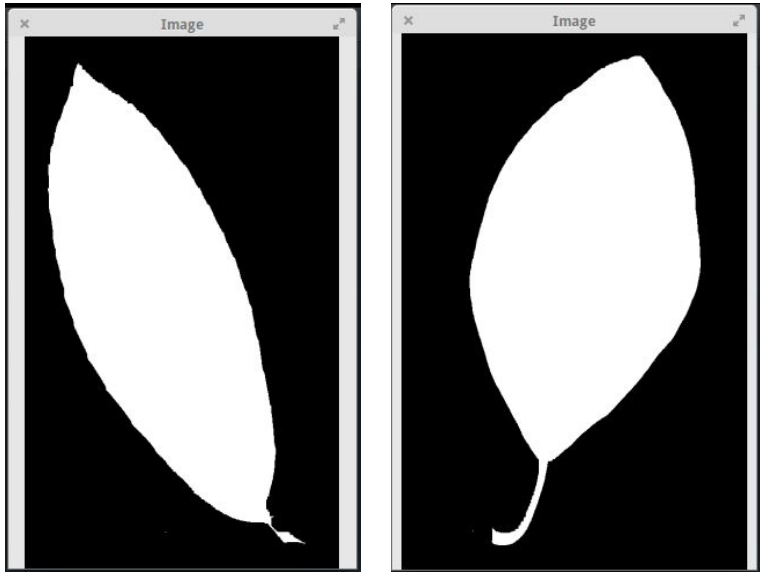
Segmentation and processing (III)





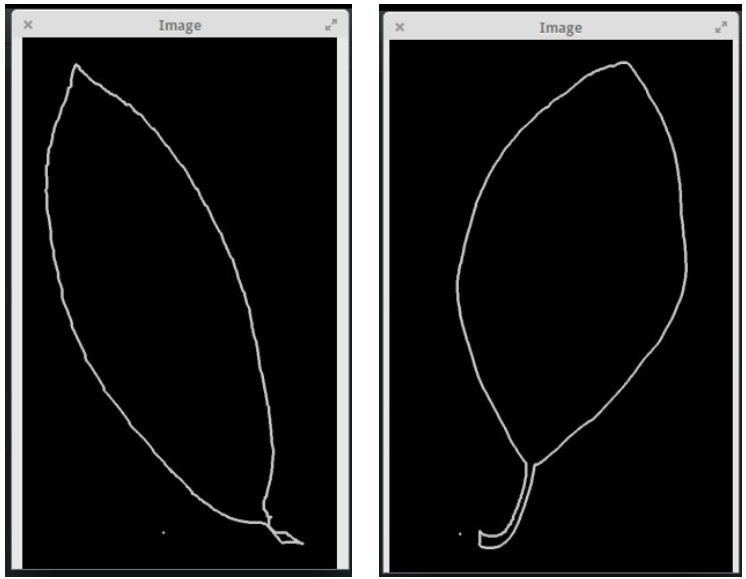
- Apply **threshold**
- **Inverted binary**
- **With** some noise…

Segmentation and processing (VI)



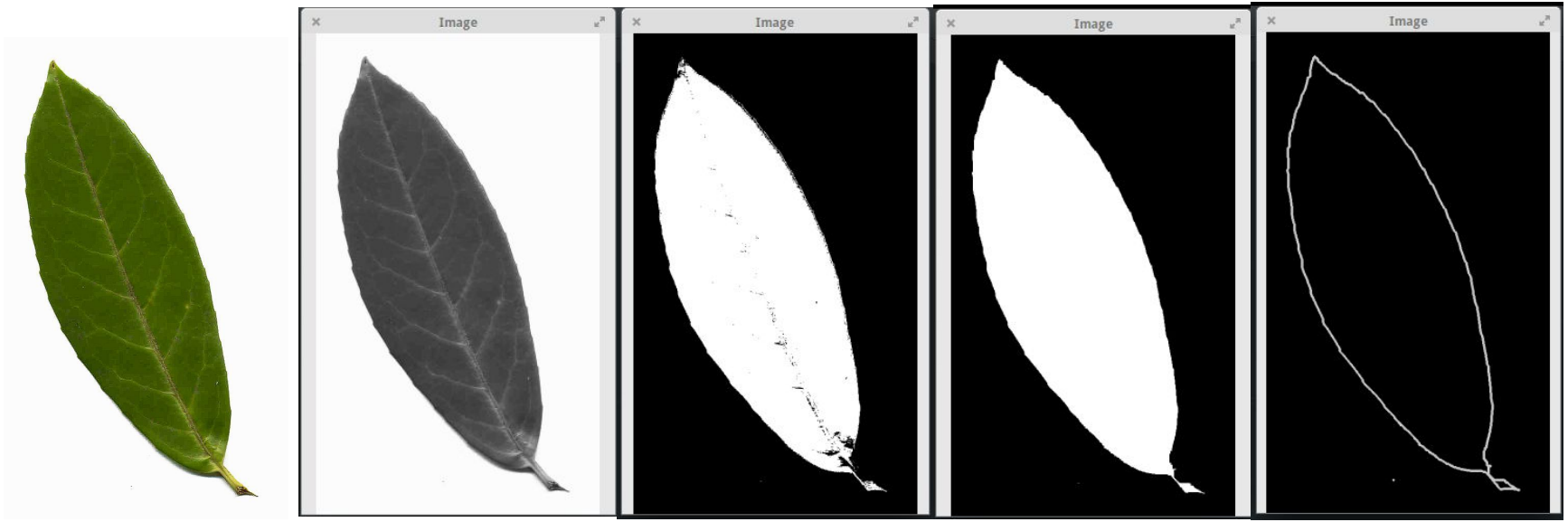
- Apply **morphological transformations**
- **Closing**
- Structuring element: **kernel**
- Allows to **remove noise**

Segmentation and processing (V)



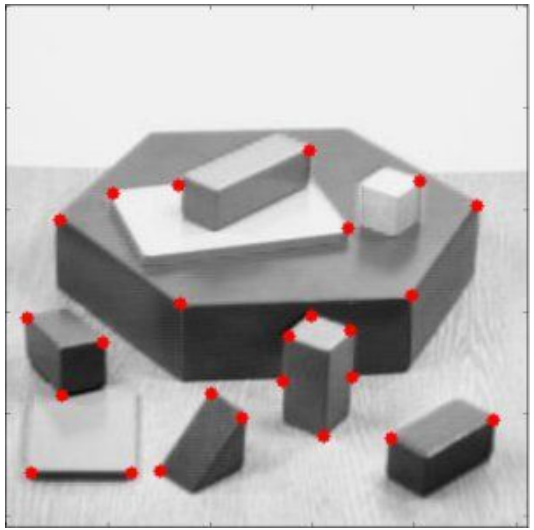
- Find and draw contours
- **findContours()**
- **drawContours()**

Segmentation and processing (VI)



Extracted Feature
Number of Corners

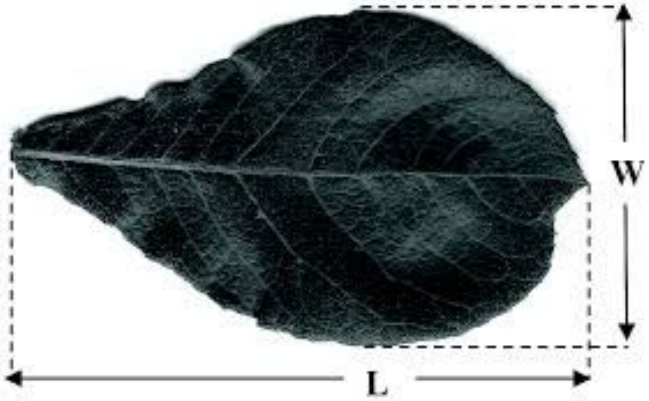
- cv2.goodFeaturesToTrack()
- Used variation of the **Harris Corner Detection** algorithm
- Determines **strong corners** on an image.



Extracted Feature
Ratio of length and Width

- Only width and height are **not useful**!

$$AspectRatio = \frac{Width}{Height}$$



Extracted Feature
Ratio of Perimeter and area

- Only areas and perimeters are not useful!

$$rpa = \frac{perimetro}{area}$$

rpa: razão entre perímetro e área

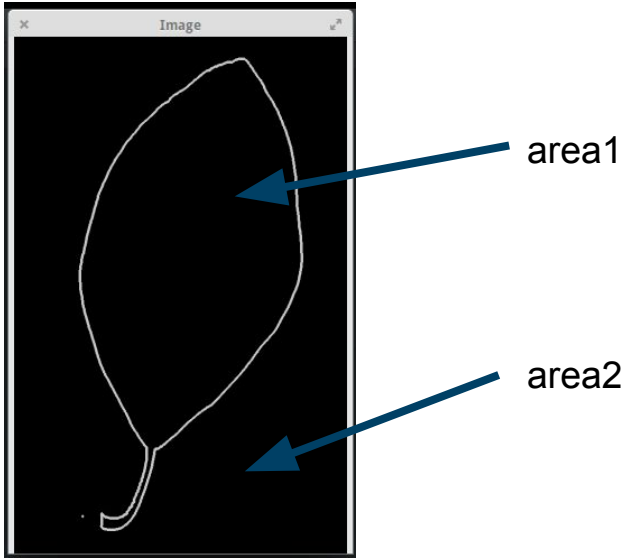
Extracted Feature

Ratio of area

- Ratio between **area of leaf** and the area of **outside the leaf's** contourn;

$$ra = \frac{area1}{area2}$$

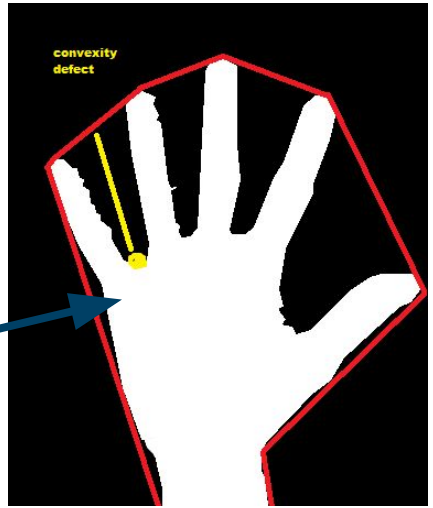
ra: razão entre áreas



Extracted Feature
Solidity

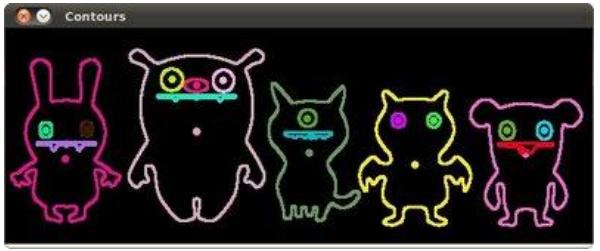
- Ratio between the **contour area** and the **convex area** formed by the union of the points forming a polygon

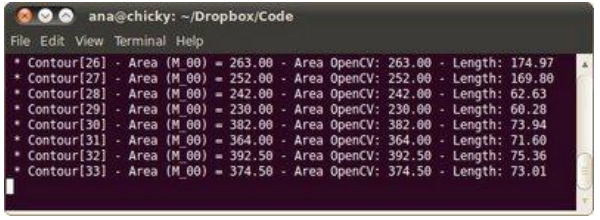
$$Solidity = \frac{Contour\,Area}{convex\,Hull\,Area}$$



Extracted Feature
Moments

- Help us to calculate some **characteristics** of the image such as, **mass center** or **orientation** of the image.
- **cv2.moments(image)**





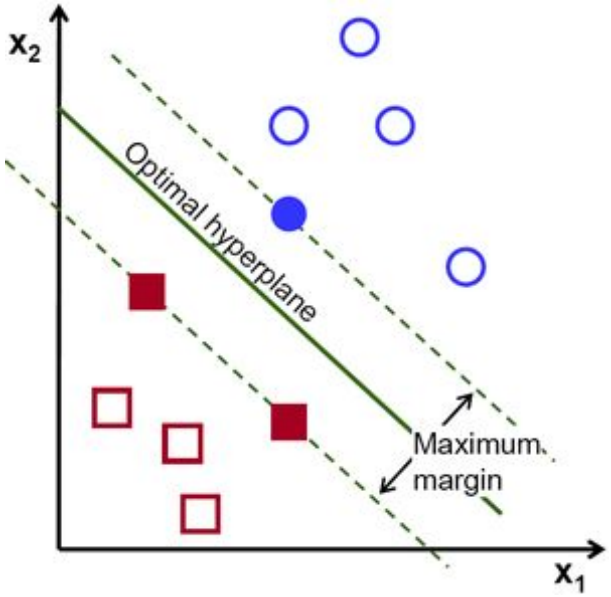
Extracted Feature
Summary

- Number of Corners
- Ratio of length and height
- Ratio of Perimeter and area
- Ratio of area
- Solidity
- Moments

Classifier

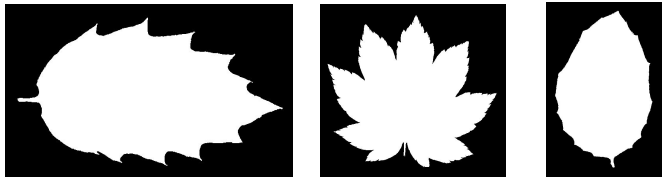
Support vector machine (SVM)

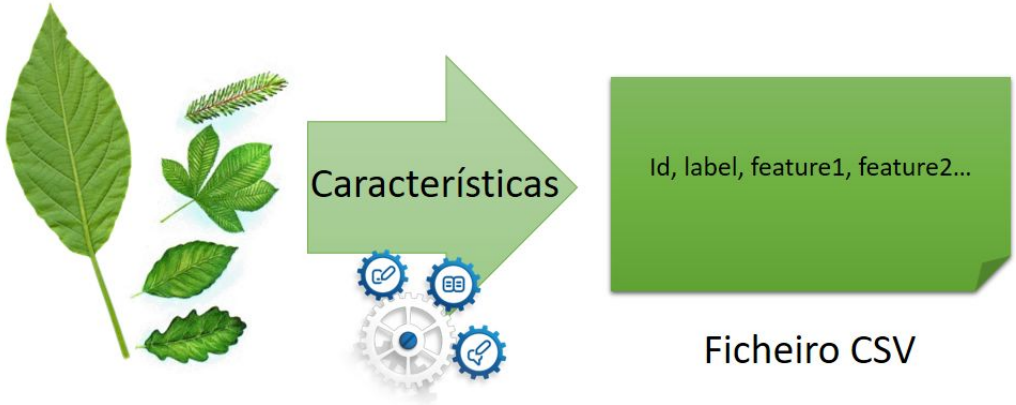
- Supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis.



Training Data (I)

- ≈ **160** leaf images (binary images)
- **10** species
- Output: **CSV file** with the extracted features





Training Data (II)

- `python training_Leaves.py` or `python training_Leaves.py nm`

  Extracts all the characteristics **except the moments**

- `python training_Leaves.py all`

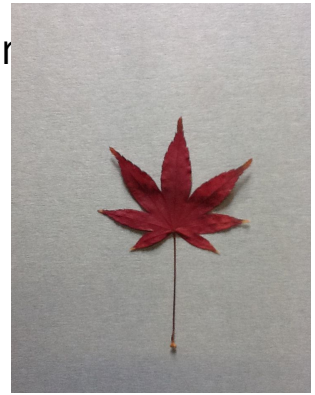
  Extracts **all** the **features**

id	label	number_corners	length_width_ratio	perimeter_area_ratio	ratio_of_areas	solidity	mu02	mu03
1	Acer_platanoides	22	1.60103626943	0.403494207766	55.1340138108	0.523424099838	53324969550.7	6.64652797279e+12
2	Acer_platanoides	53	1.63001240182	0.411419491143	86.1167319742	0.447832445344	611007314359.0	3.1631336848e+14
3	Acer_platanoides	27	1.28092783505	0.409558004209	54.570756469	0.522850288505	47910792572.0	1.00351232489e+12
4	Acer_platanoides	29	1.61773255814	0.402451920492	54.8989299733	0.496183423517	45808535982.0	6.13111977024e+12
5	Acer_platanoides	33	1.63769690386	0.405251007552	72.3952341824	0.556717157664	208447567256.0	2.56630831266e+13

Testing Data (i)

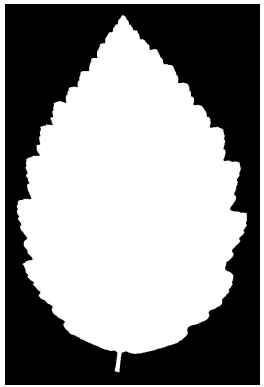
- `python test_Leaves.py`
  All the features **except the moments** are used in the **classification**

- `python test_Leaves.py all`
  All the characteristics are used in the **classification**



Testing Data (II)

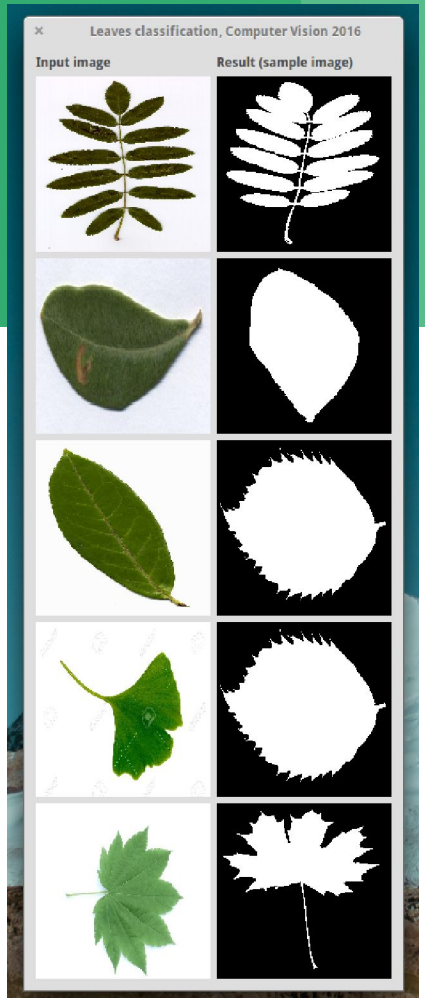
- Test image in folder `Test_Leaves`
- Extract the characteristics of testing images
- Send those and training data's characteristics to the SVM
- Predict the plant's specie



© baumportal.de

Results

- Interface with
  - input images (at left)
  - predicted specie image (at right)
- In terminal we have the name of the predicted specie and of the input image (e.g Ginkgo_biloba2.jpg)



Demo

Conclusions

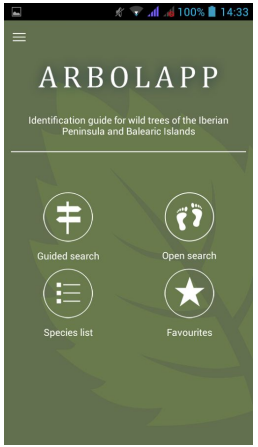
- Previsions are **not** always correct.
- SVM predict with **higher accuracy** testing the features **without moments.**
- Input images **used in training** are always predicted **correctly.**

Future Work

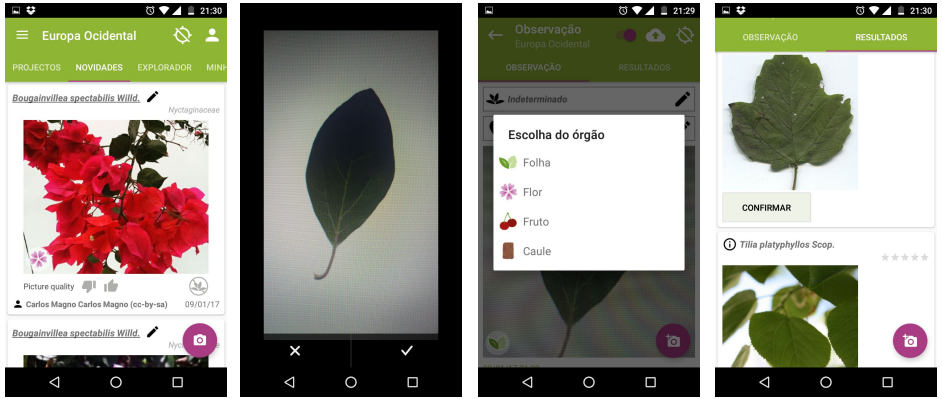
- Use **deep learning** and/or **neural networks** to increase accuracy.
- Increment/Improve the **vector of features.**
- **Increase** the training **data set**, maybe with colored images too.
- **Improve** the predictions for images in a more **realistic context**, maybe with leaf tracking first and then all the pré-processing

Mobile Application For Leaves Recognition

Pl@ntNet



treezilla
the monster map of trees



Questions?