

# MLRF Lecture 02

J. Chazalon, LRDE/EPITA, 2019

# Agenda for lecture 2

1. Introduction
2. Global image descriptors
3. Clustering
4. Texture descriptors
5. (Character descriptors)
6. Local feature detectors

# Introduction

Lecture 02 part 01

Previously, in MLRF...

# Summary of last lecture

## Machine learning

- Machine learning = searching for the best model in a hypothesis space
- Inductive machine learning, optimization-based
- Inductive bias, bias/variance compromise
- Supervised, reinforcement, unsupervised learning
- Regression, classification, density estimation
- Model validation: test generalisation, separate/decorrelate test & training sets

## Template matching

- Sum of squared differences  $(T-I)^2$ , or correlation-based methods  $(T \times I)$
- Normalization needed for correlation-based methods
- Tolerates translation and small noise, but not rotation, intensity shift, ...

# Debriefing of practice session 1

## PS1 content

1. Jupyter tricks
2. NumPy reminders
3. Intro to image manipulations
4. Twin it! part 1: Template matching
5. (Bonus level: segmentation)

## Discussion

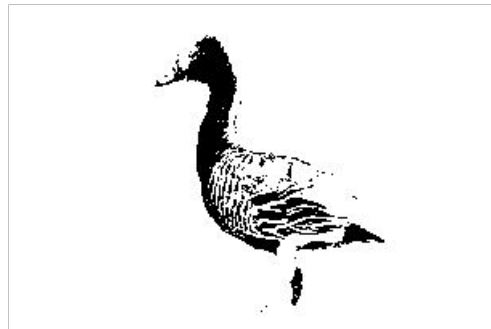
- Who completed part 1? 2? 3? 4? 5?
- Any remarks, comments, questions?
- Things to keep, change, remove?

# Practice session 1: Take home messages (1/2)

background-less duck



mask



*How annoying was it  
to manually adjust color thresholds  
to select the duck?*

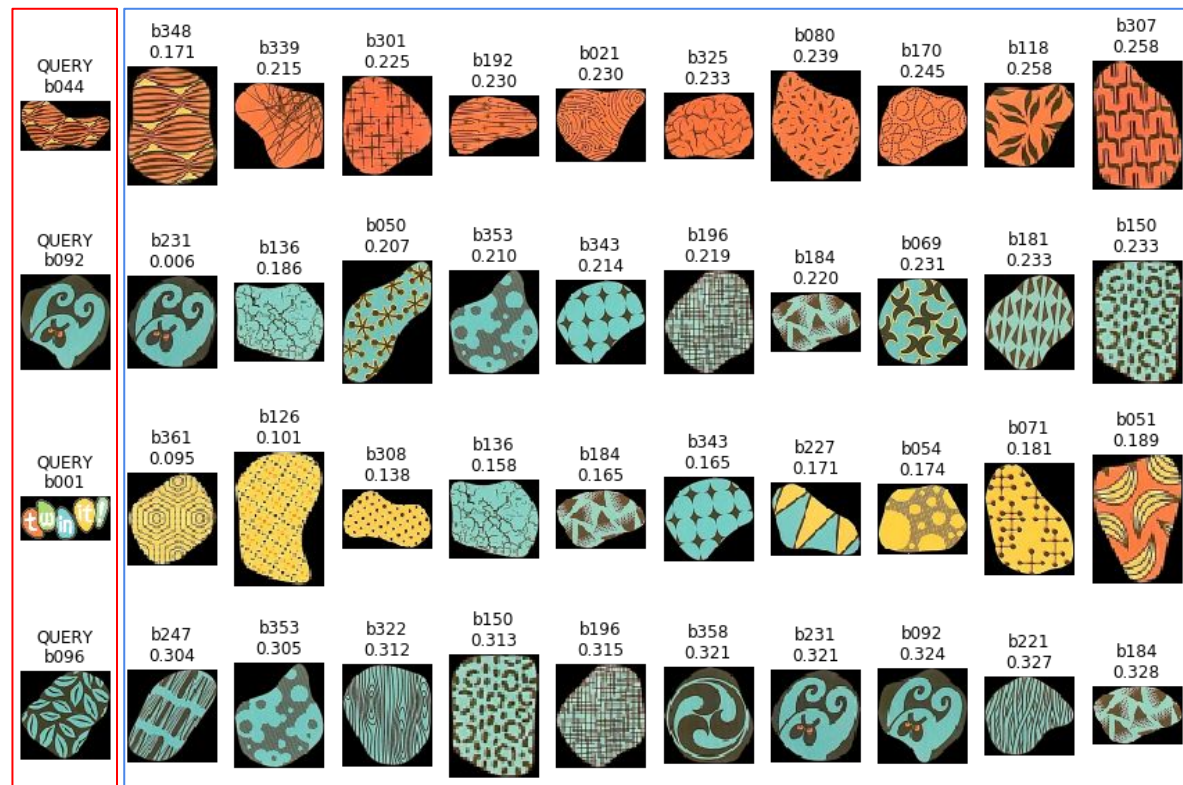
*How could we have automated it?*

# Practice session 1: Take home messages (2/2)

Results with method  
SQDIFF\_NORMED →  
(lower is better)

*Strengths and  
weaknesses of  
template matching  
for the Twin it! case?*

*Effects of normalization?*



query images

result images (closest to query according to method)



Next practice session(s)

# Next practice session

**Memory/storage (RAM, partitions) issues should be fixed.**

They were due to:

1. A background update of Arch filling the root partition (RAM)
2. `pip install --user` installing to `~/.local`: used to be on local partition (RAM)
3. *Twin it!* resources taking too much space on local partition (RAM again)

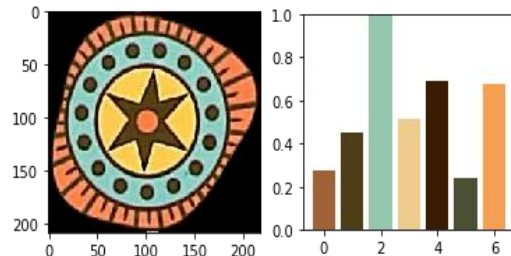
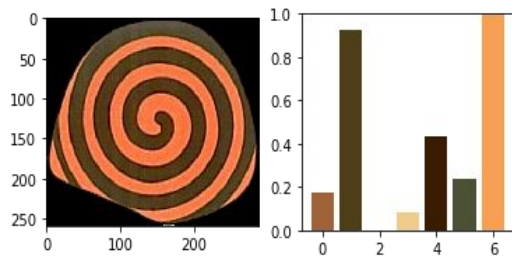
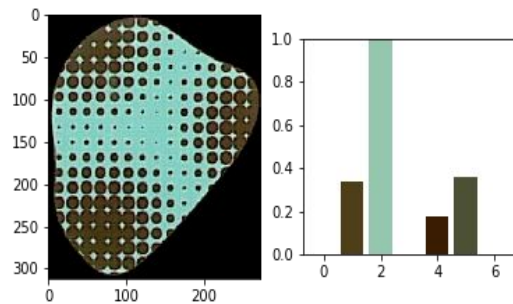
Solutions:

1. Marin fixed the image 👍
2. Tricks to store Python packages on AFS: **see the README.md!!**  
`/afs/cri.epita.net/resources/teach/bigdata/install.sh` ← run once for all  
`/afs/cri.epita.net/resources/teach/bigdata/configure.sh` ← each session
3. Resources are now stored on AFS too: **see the README.md!!**  
`/afs/cri.epita.net/resources/teach/bigdata/mlrf19/`

# Next practice session

*Twin it!*, again, with a slightly more elaborated approach:

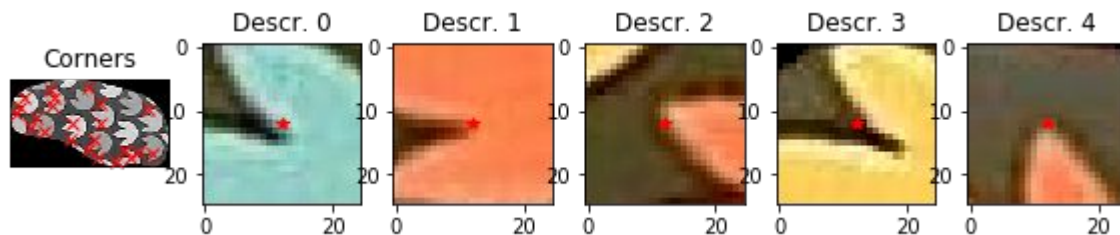
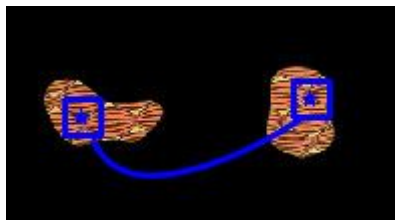
## 1. Pre-select bubbles based on their colors $\Rightarrow$ Color histograms



# Next practice session

*Twin it!*, again, with a slightly more elaborated approach:

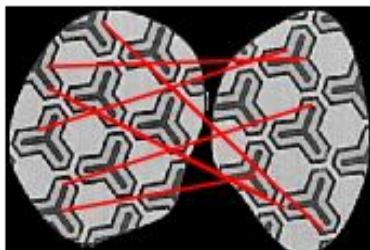
1. Pre-select bubbles based on their colors  $\Rightarrow$  Color histograms
2. **For the pre-selected bubbles, check their content is similar**  
 $\Rightarrow$  **Detect stable points and extract the patches around them**



# Next practice session

*Twin it!*, again, with a slightly more elaborated approach:

1. Pre-select bubbles based on their colors  $\Rightarrow$  Color histograms
2. **For the pre-selected bubbles, check their content is similar**  
 $\Rightarrow$  Detect stable points and extract the patches around them  
 $\Rightarrow$  **Compare (match) those patches**



# Exams

# About the exam(s)

We have to grade you...

We are still working on the best option(s):

- Final exam (QCM or other pen/paper) ⇐ for sure ✓
- function(Kaggle ranking on practice 4, 5, 6) ⇐ you were OK +
- Project ⇐ you preferred to avoid ✗
- **Practice exam on machines** ⇐ **is this a good idea?** ? ?
  - Like for Olivier's course: any feedback on this experience?
  - Would replace / complement grading on practice sessions
  - Offline machines but all documents tolerated

# Image descriptors



# Issues with methods based on pixel comparison

What is important? What do they consider? **Raw pixels!**

⇒ We want to be able to make use of **domain knowledge!**

*Like sensitivity to shape, or dominant color information.*

They are terribly **slow** and works **only for small images.**

⇒ We want to **summarize an image** to a much smaller vector.

They are **sensible to rotation, scaling**, and many other perturbations.

⇒ We want to adjust sensitivity/invariance to perturbations.

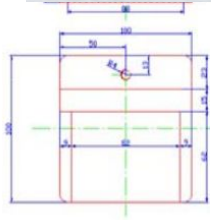
*Do we tolerate translation? Rotation? Intensity shift?*

How can we compare different pairs of images? **Metric issues.**

⇒ We want to be able to achieve **more than 1 vs all comparisons.**

# Image descriptors: Overview

Different sizes and contents  $\Rightarrow$  Different kind of descriptors

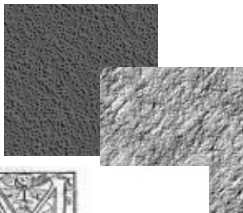


Large images, many parts

$\approx 500 \times 500$  px and more



Complex  
small images



La parole se feroit la fondue dedans  
enoy, de le poule cette sans mouvoir. Si se  
ra'auis, rendus les honneurs furebren, de  
les 'dormiens' offrandes à la monnaie  
de ce grand Chevalier François, en fou-

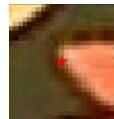
Textured  
areas

Office for

Words



Logos



Local image patches,  
lines, etc.



Isolated  
symbols and  
letters

$\approx 20 \times 20$  px and less

# Image descriptors: Overview

**Different sizes and contents  $\Rightarrow$  Different kind of descriptors**

**Different problems  $\Rightarrow$  Different choices**

- Computation / memory constraints
- Which perturbations to we have to tolerate?  
*rotation, translation...*
- What is the expected output?  
*classification, detection, ranking, segmentation...*

**Many, many approaches  $\Rightarrow$  Impossible to list them all**

- Examples of several categories
- Focus on very useful or instructive ones