Master in Computer Vision *Barcelona*

Module 3: Machine learning for computer vision

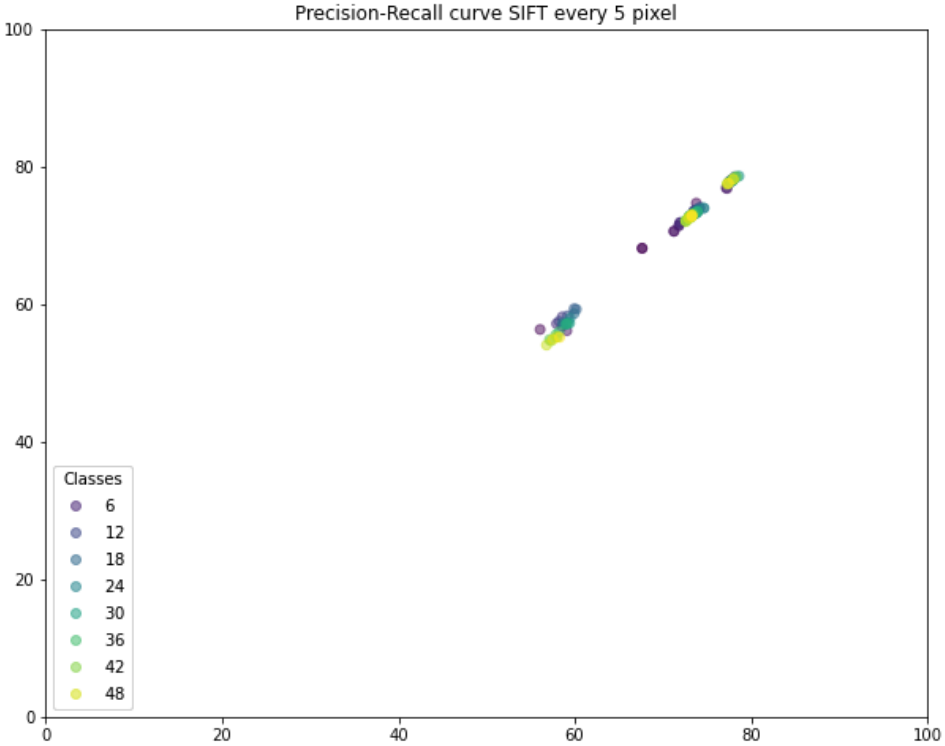
Project: Bag of Visual Words Image Classification

Lecturer: Ramon Baldrich, ramon.baldrich@uab.cat

Credit to Marçal Rossinyol

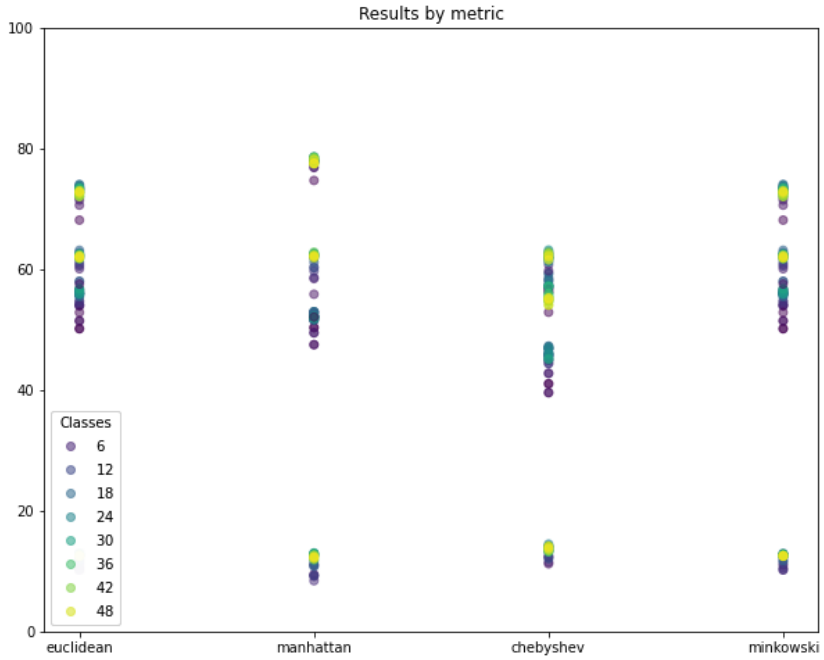
S01 discussion

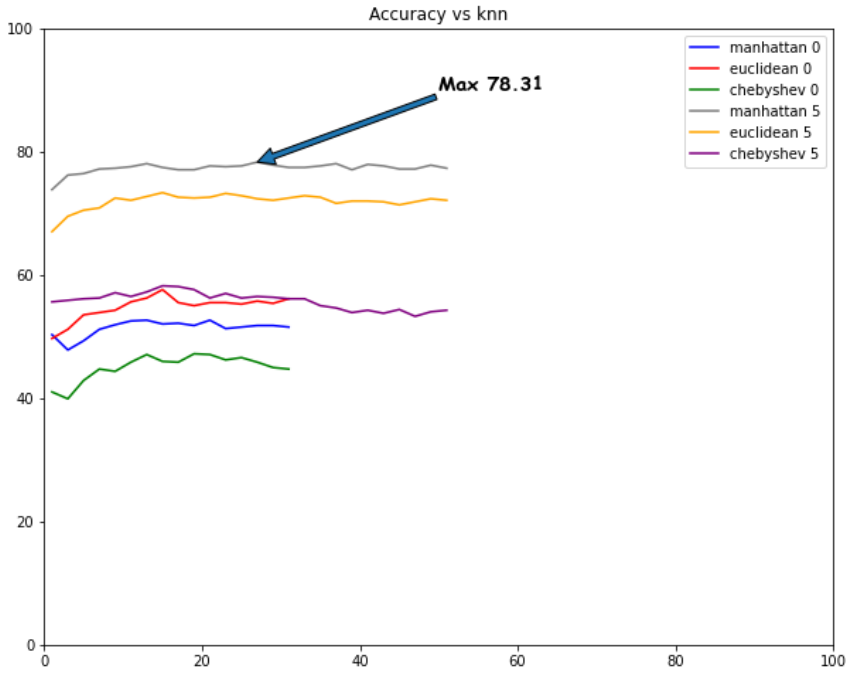
- Number of keypoints in SIFT
  - The more the better
- Dense SIFT
  - Nearly nobody tried different scales!
- Codebook sizes / k-nn value
- k-nn and distances
  - Just slight differences found between point-wise distances
  - Which distance would work better for HISTOGRAMS?
- Dimensionality reduction
- Precompute stuff, store to disk!

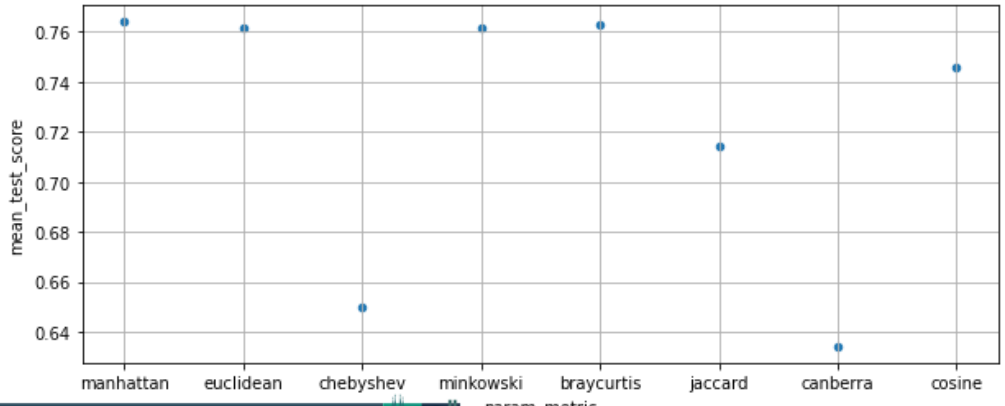


When DENSE is done with step size of 5, the maximum value stays at KNN equals to 27 however it reaches 78%, what happens if we increase step size

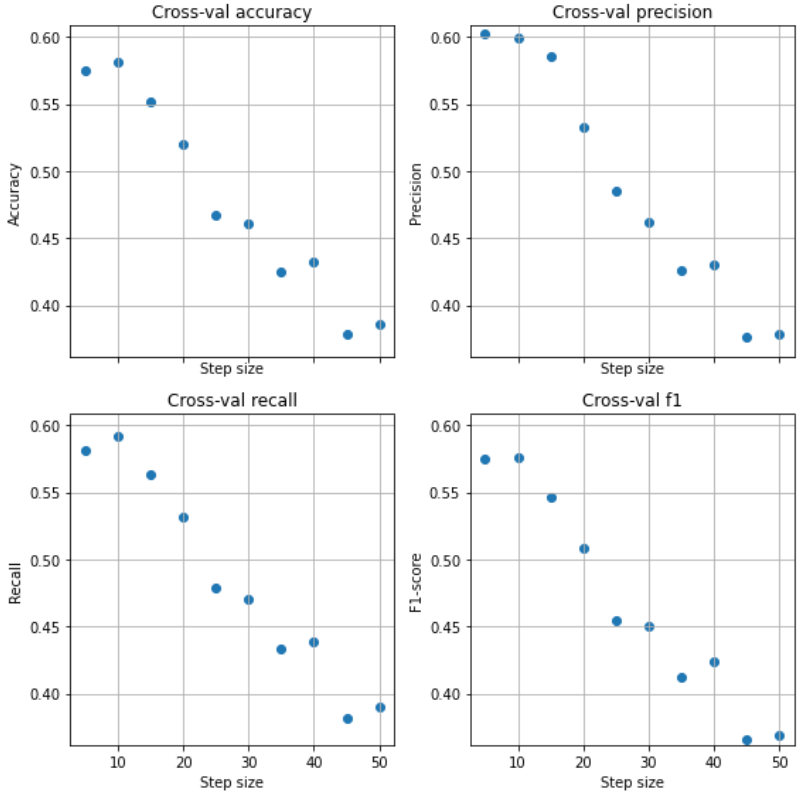
Distance?

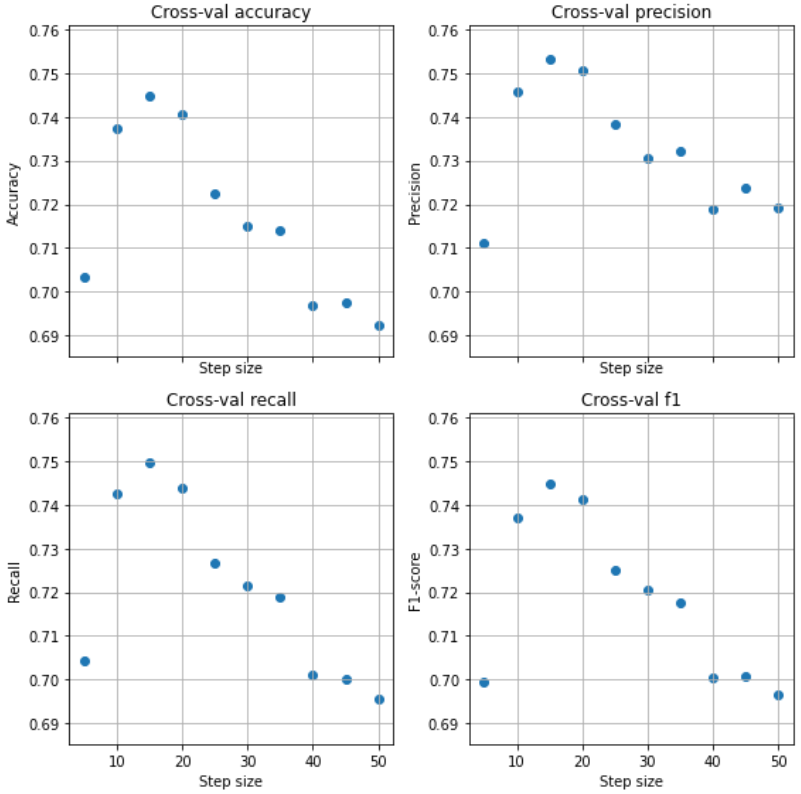

Results by metric
Classes
6
12
18
24
30
36
42
48
euclidean
manhattan
chebyshev
minkowski


Accuracy vs knn
Max 78.31
manhattan 0
euclidean 0
chebyshev 0
manhattan 5
euclidean 5
chebyshev 5

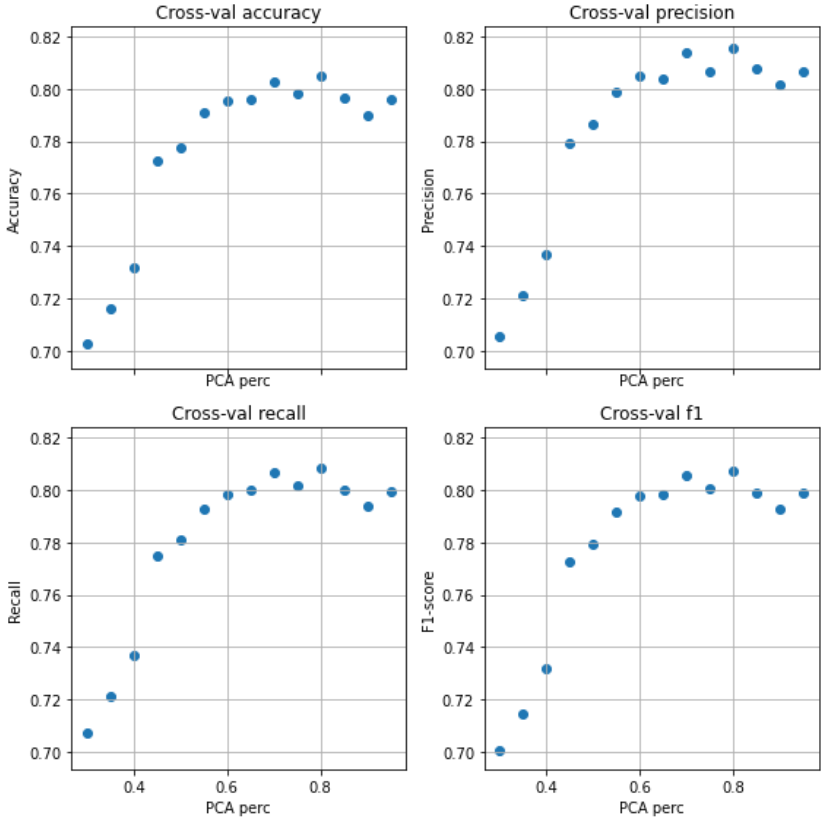

mean_test_score
manhattan
euclidean
chebyshev
minkowski
braycurtis
jaccard
canberra
cosine
param_metric

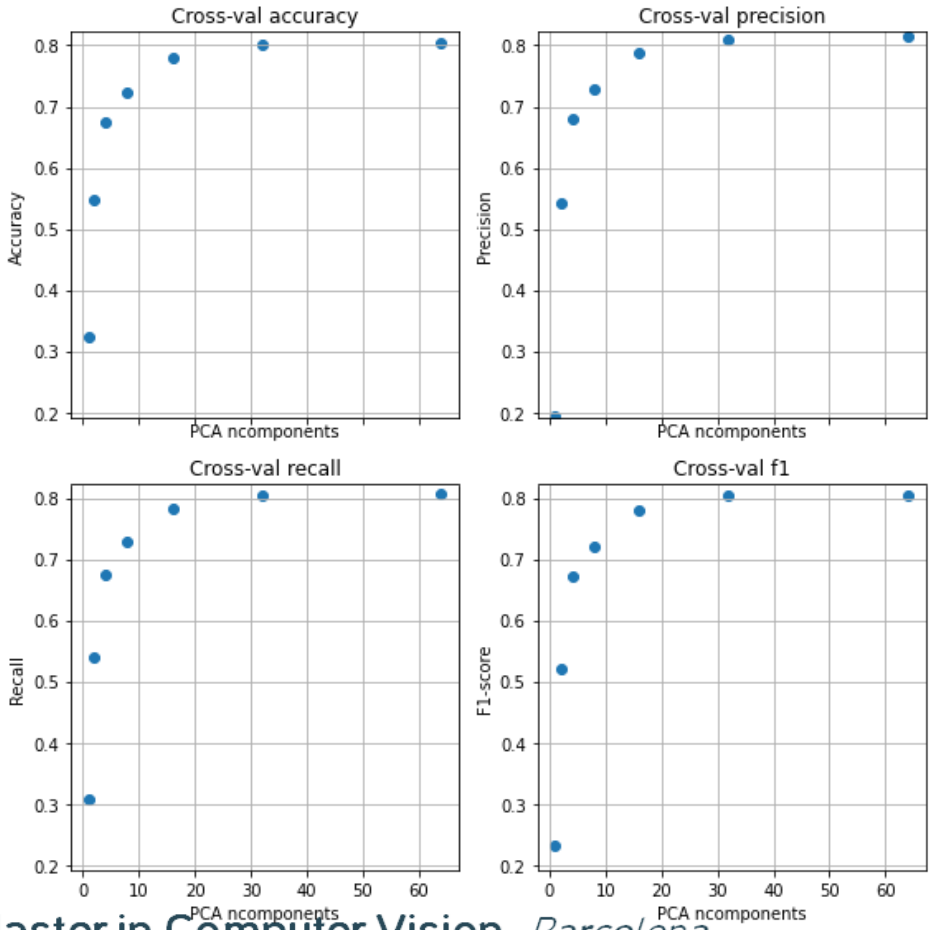
k (for batch_size=k * 20)	Accuracy	PCA	LDA
64	54.64	54.77	57.62
128	59.85	59.97	61.21
256	56.87	62.85	60.84
512	52.29	60.84	59.85
1024	39.40	61.58	48.20
2048	23.91	61.46	21.80

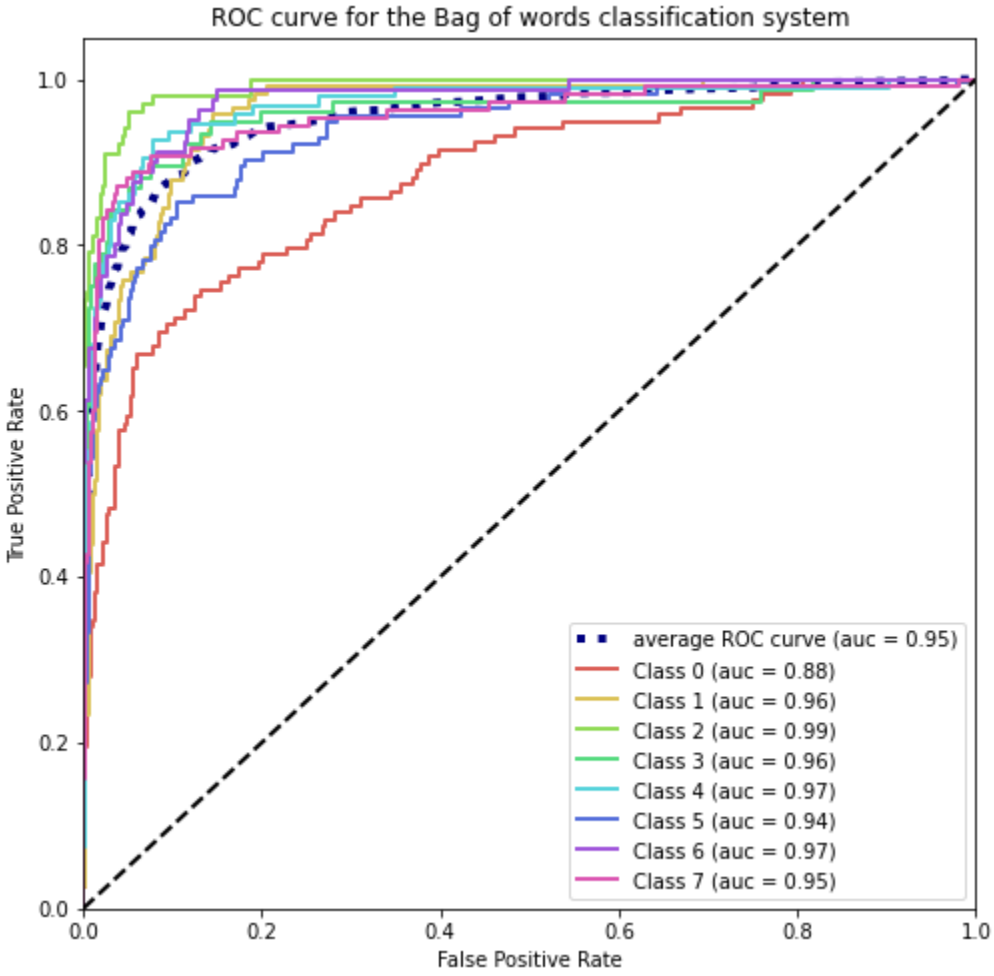

Cross-val accuracy
Accuracy
Step size
Cross-val precision
Precision
Step size
Cross-val recall
Recall
Step size
Cross-val f1
F1-score
Step size


Cross-val accuracy
Accuracy
Step size
Cross-val precision
Precision
Step size
Cross-val recall
Recall
Step size
Cross-val f1
F1-score
Step size

PCA on visual words domain





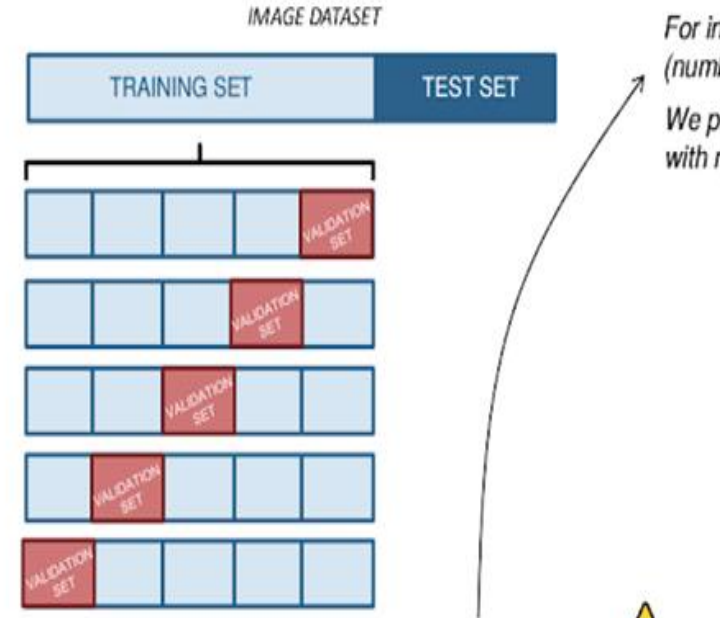

ROC curve for the Bag of words classification system
True Positive Rate
False Positive Rate
0.0
0.2
0.4
0.6
0.8
1.0
average ROC curve (auc = 0.95)
Class 0 (auc = 0.88)
Class 1 (auc = 0.96)
Class 2 (auc = 0.99)
Class 3 (auc = 0.96)
Class 4 (auc = 0.97)
Class 5 (auc = 0.94)
Class 6 (auc = 0.97)
Class 7 (auc = 0.95)


Accuracy
KNN Accuracy
63
62
61
60
59
58
0
20
40
60
80
100
Different values for K neighbours.

S02

- We'll start with BoVW computed with Dense SIFT with a large enough codebook size
- We'll normalize descriptors
  - L2-norm, Power-norm, etc..
- Cross-validation
  - Sklearn functions: StratifiedkFold, GridsearchCV
- Spatial Pyramids
- SVM and kernels
  - Use sklearn standardScaler to project every dimension to [0, 1]!
  - linear kernel
  - RBF kernel
  - our own histogram intersection kernel
- OPTIONAL: Fisher Vectors (http://yael.gforge.inria.fr/tutorial/tuto_imgindexing.html)

Cross Validation



We use K-fold cross validation to pick suitable parameters for the classifier while **avoiding overfitting**.

*For instance, in the KNN classifier we want to find the most suitable **K** (number of neighbours used to classify each descriptor)*

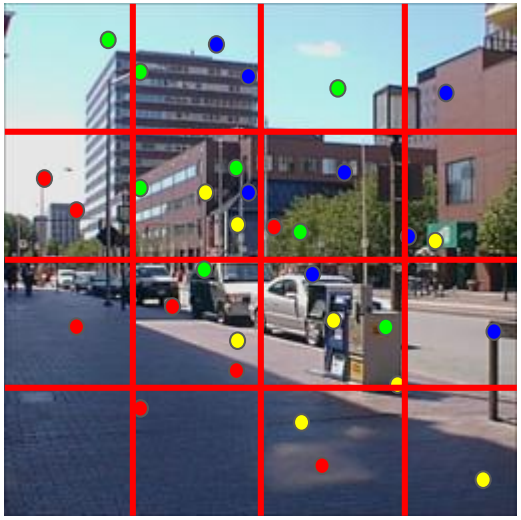
*We plotted the average of the accuracies (Y axis) obtained with 5 folds with respect to varying **K** from 1 to 100 (X axis).*

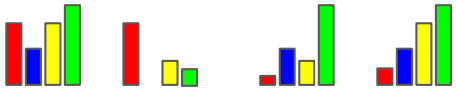
Best average accuracy with K = 30 !

So we obtain that individual SIFT descriptors are best classified using K = 30 (for another kind of descriptor the 'best' K will vary)

Spatial Pyramids

Spatial Pyramids





Histogram Intersection kernel

```
def histogramIntersection(M, N):
```

$$K_{int}(A, B) = \sum_{i=1}^{m} \min\{a_i, b_i\}.$$

```
    return K_int
```

Tasks to do

Improve the BoVW code with:

- Dense SIFT (with tiny steps and different scales!)
- L2-norm - power norm
- SVM classifier
- StandardScaler
- Cross-validation
- Linear, RBF and histogram intersection kernels
- Spatial Pyramids
- Fisher Vectors (OPTIONAL)

Deliverable

- A **single Python notebook file per group** reporting all the work done,
  - with the different experiments,
  - code,
  - plots,
  - explanations, etc.
  - **EVERYTHING EXECUTED!**

- To deliver by Monday 18th @ 10 A.M.
  - Please, state clearly your group.

Warning: provided code might not work out of the box depending on the used versions (OpenCV, numpy, sklearn…) do not panic, and ~~RTFM~~ read the documentation