



Master in Computer Vision *Barcelona*

Module 3: Machine learning for computer vision

Project: Bag of Visual Words Image Classification

Lecturer: Marçal Rossinyol, marcal@cvc.uab.es



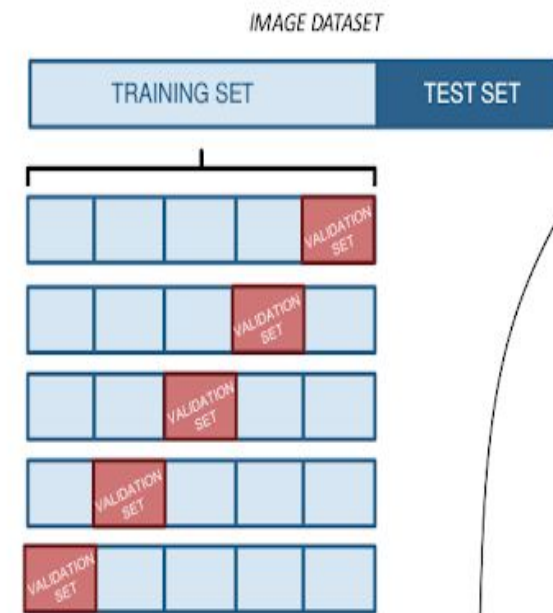
S01 discussion

- Number of keypoints in SIFT
 - The more the merrier
- Dense SIFT
 - nobody tried different scales!
 - A couple of groups didn't manage to get it working (i.e. they obtained worse results)
- Codebook sizes / k-nn value
- k-nn and distances
 - Just slight differences found
 - Which distance would work better for HISTOGRAMS?
- Dimensionality reduction
- Precompute stuff, store to disk!

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

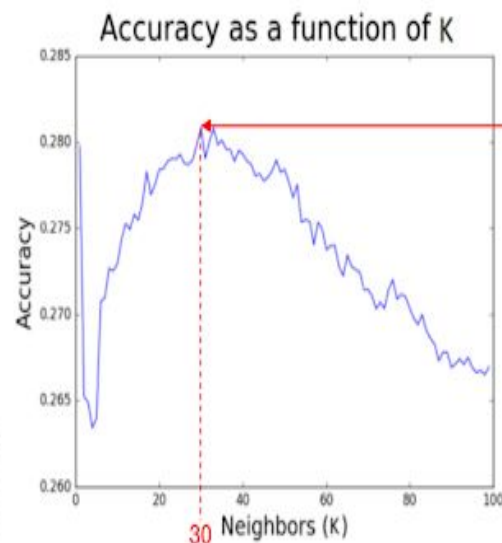


We use K-fold cross validation to pick suitable parameters for the classifier while **avoiding overfitting**.



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).



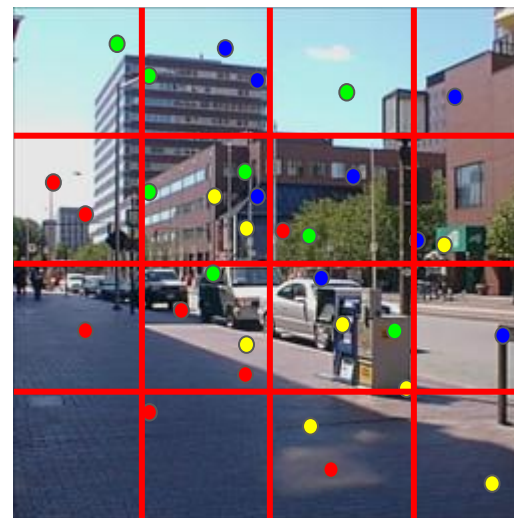
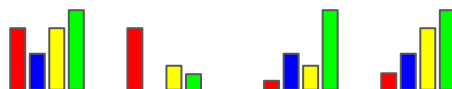
Best average accuracy with $K = 30$!

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 16th @ 10 A.M. by email (marcal@cvc.uab.es)
 - 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