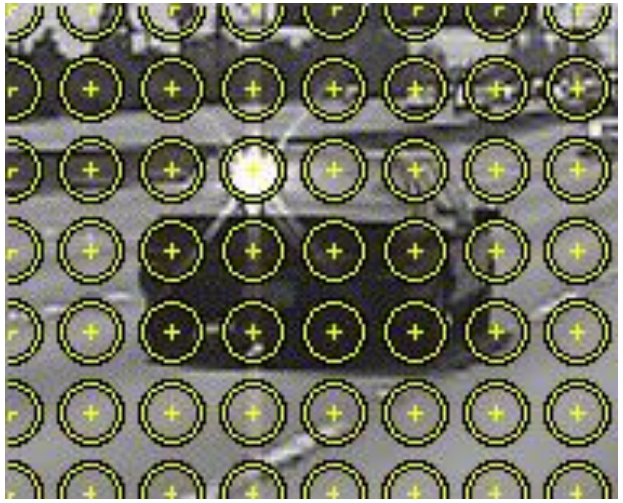
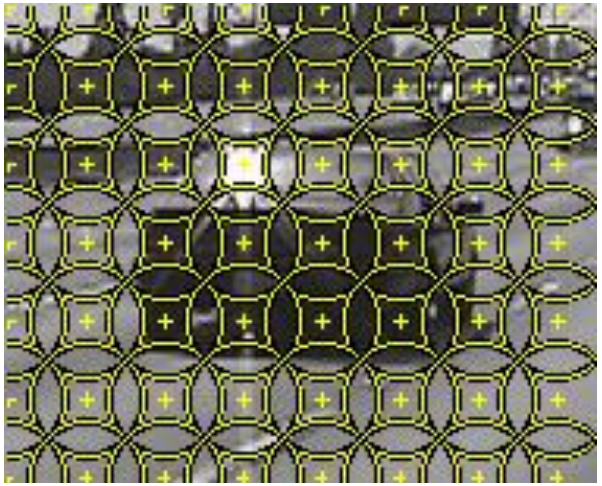
Experiments on Densely Sampling

Xin Sui

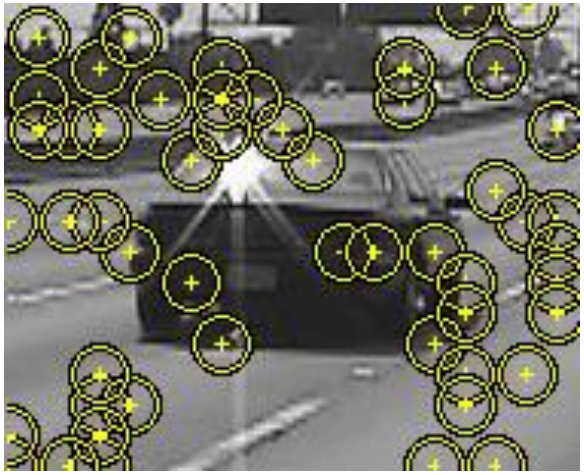
Three random sampling methods



Non-Overlapping

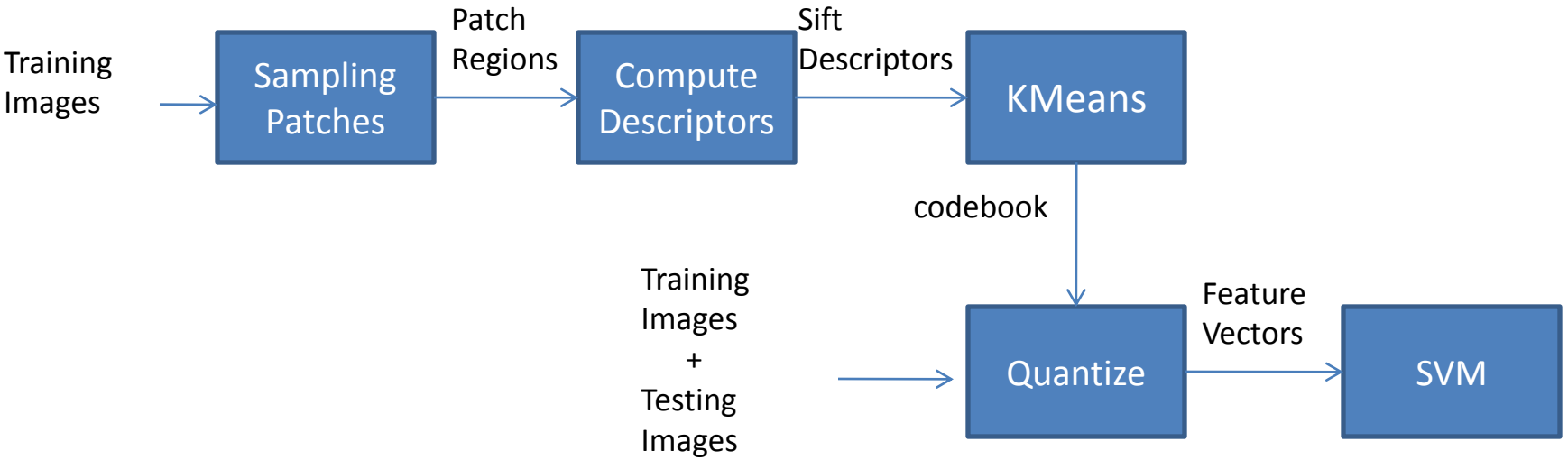


Overlapping



Random-Overlapping

Process Flow



Tools

1. Compute_Descriptor.ln: Given region, compute the sift descriptor http://www.robots.ox.ac.uk/
2. LibSvm
   a convenient SVM tool
   http://www.csie.ntu.edu.tw/~cjlin/libsvm/

SVM: RBF Kernel
With gamma set to the median of the pairwise distances between the training descriptors

Sampling

- Step 1: Generating regions files
  - Ellipse Region
    - a(x-u)(x-u)+2b(x-u)(y-v)+c(y-v)(y-v)=1
  - format:
    1.0
    m
    u_1 v_1 a_1 b_1 c_1
    :
    :
    u_m v_m a_m b_m c_m
- Step 2, call Compute_descriptor.ln to generate sift descriptor files
-
  - **Descriptor output**
  - format:
    N
    m
    u_1 v_1 a_1 b_1 c_1 $d_{1,1}$ $d_{1,2}$ $d_{1,3}$ ... $d_{1,N}$
    :
    :
    u_m v_m a_m b_m c_m $d_{m,1}$ $d_{m,2}$ $d_{m,3}$ ... $d_{m,N}$

Dataset

- Pascal VOC 2005
  - Four Categories
  - Binary classification

	Motorbikes	Bicycles	People	Cars
Training	214	114	84	272
Test(test 1)	216	114	84	275


www.solomotobmw.it
RUS
E
SK


GAZELLE
ROGERIE HOPPEL
FILM + FOTO SERVICE

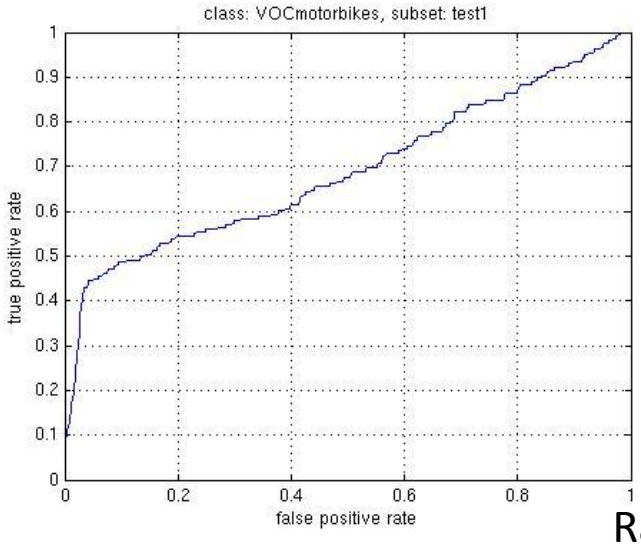

ESTEBE

Results (200 words)

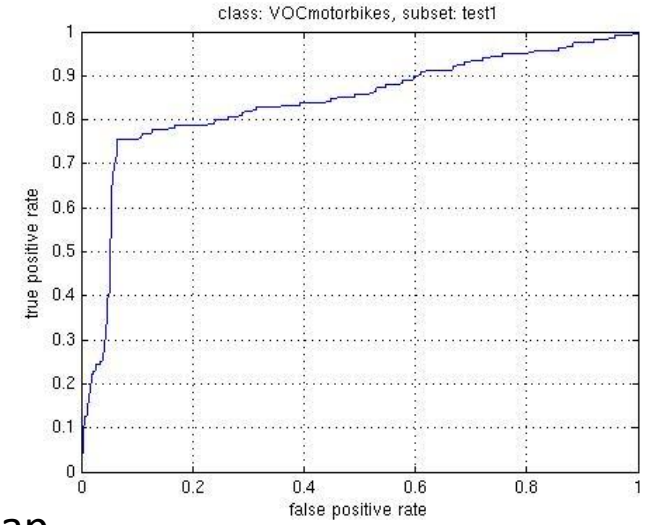
- Accuracy (correct predictions/the total test examples)

	Motorbikes	Bicycles	Cars
Non-Overlap	82.148% (566/689)	85.0508% (586/689)	80.4064% (554/689)
Overlap-rand	79.6807% (549/689)	Predict negative for almost all images	78.2293% (539/689)
Overlap-big	87.8084% (605/689)	88.5341% (610/689)	84.0348% (579/689)

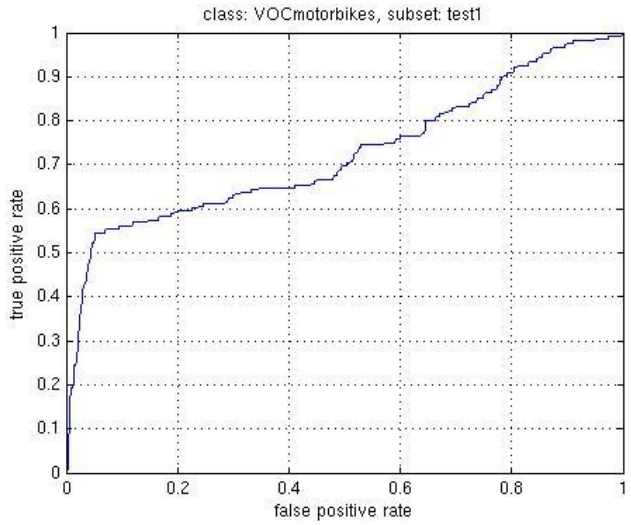
ROC curves_motorbike



Random-Overlap

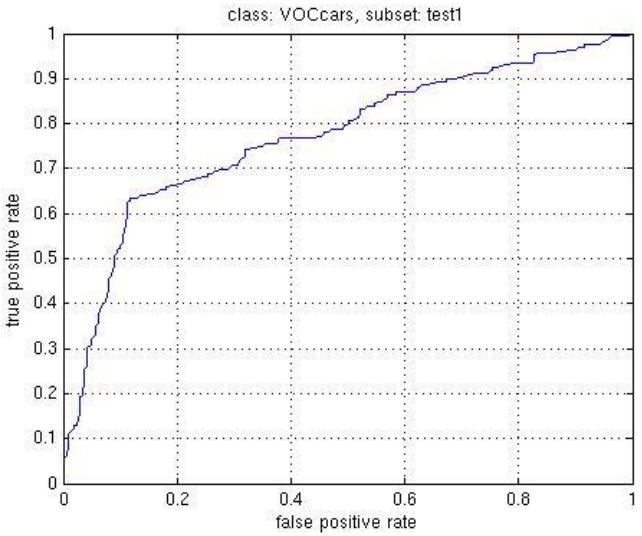


Overlap-big

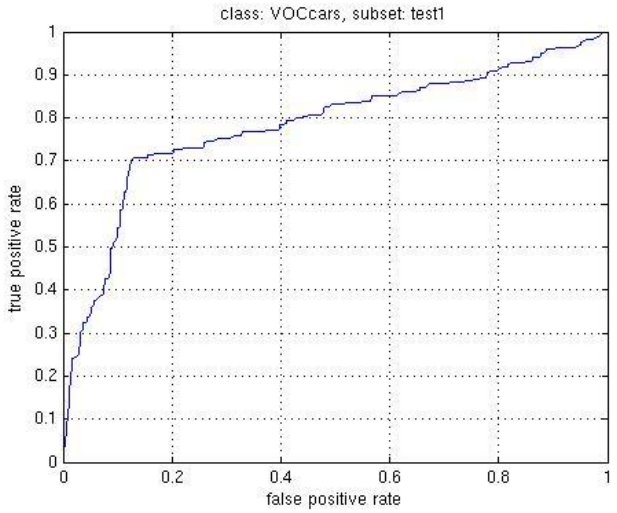


Non-overlap

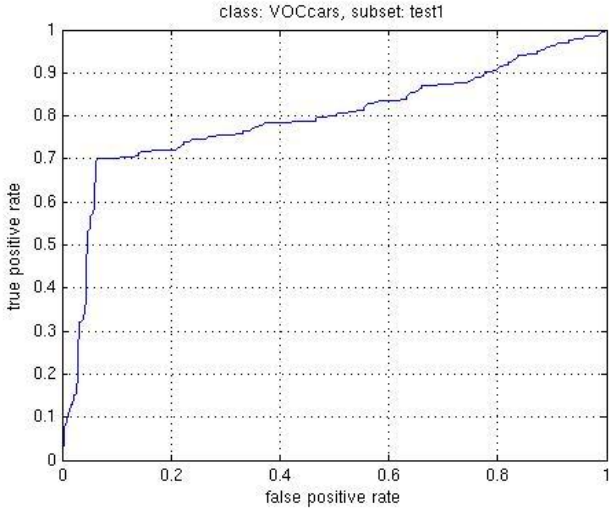
ROC curves_cars



Random-Overlap



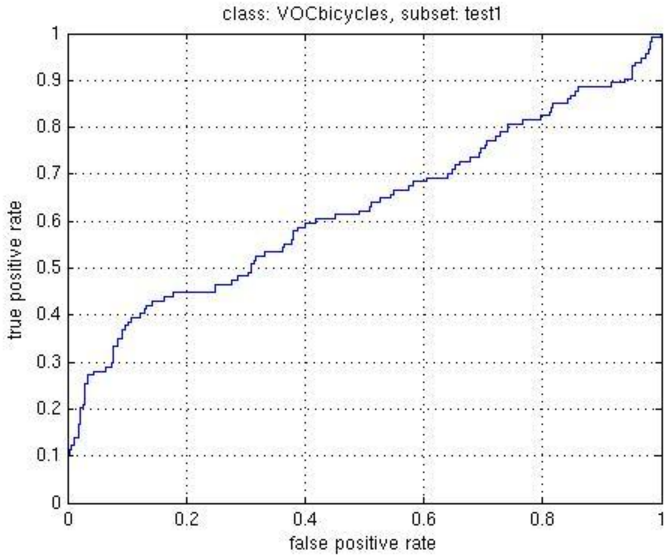
Non-overlap



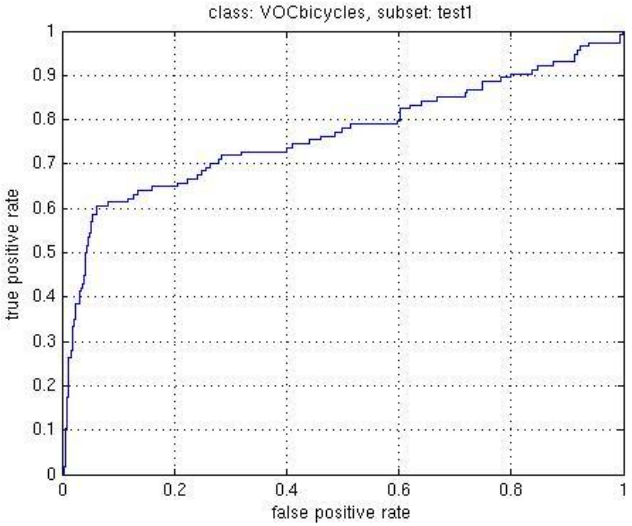
Overlap-big

ROC curves_bikes

Overlap_rand: predicting negative for most of all test images



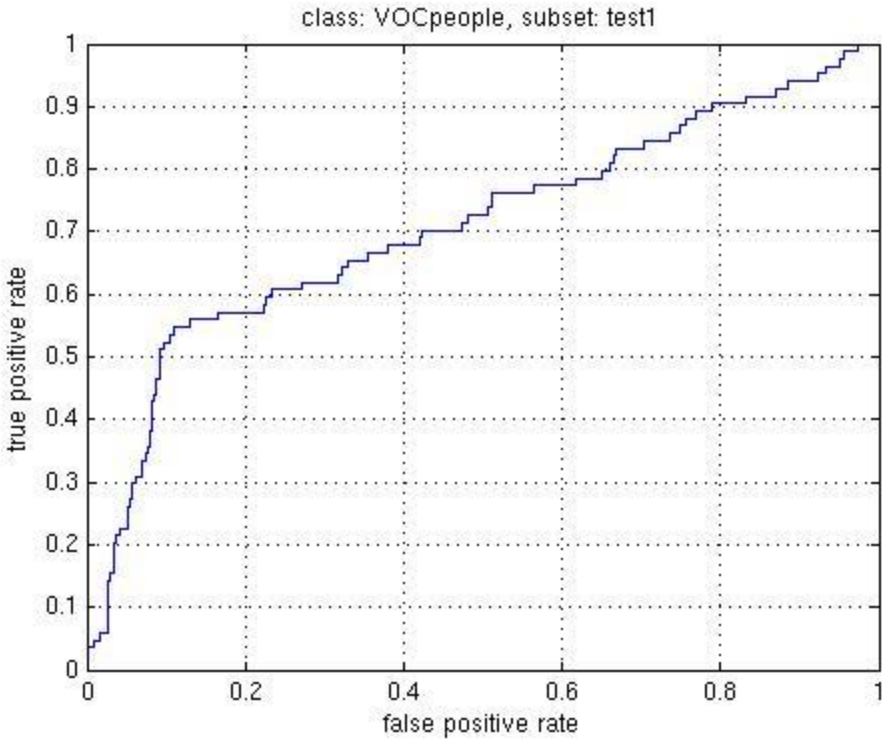
Non-overlap



Overlap-big

ROC curves_people

Overlap_rand: predicting negative for most of all test images
Non-Overlap: predicting negative for most of all test images



Overlap-big

Conclusion

- According to the above experiment, it seems:
  - Uniformly distributed patches with overlapping seems better
  - Uniformly distributed patches are better than random distributed patches