Local Features and Kernels for Classifcation of Texture and Object Categories: A Comprehensive Study

J. Zhang[1] M. Marszałek[1] S. Lazebnik[2] C. Schmid[1]

[1]INRIA Rhône-Alpes, LEAR - GRAVIR
Montbonnot, France

[2]Beckman Institute, University of Illinois
Urbana, USA

Beyond Patches Workshop, 2006

Overview

- We have built an extensible image classification framework
  - sparse local features
  - bag-of-features image representation
  - non-linear Support Vector Machines (SVMs) for classification
- We have evaluated various elements of the framework on
  - 4 texture datasets (UIUCTex, KTH-TIPS, Brodatz, CUReT)
  - 5 object category datasets (Xerox7, Caltech6, Caltech101, Graz, PASCAL 2005)
- The conclusions hold over the datasets
- We have performed a detailed evaluation of the background influence
  - to check whether we can exploit context information
  - to evaluate the robustness against background clutter

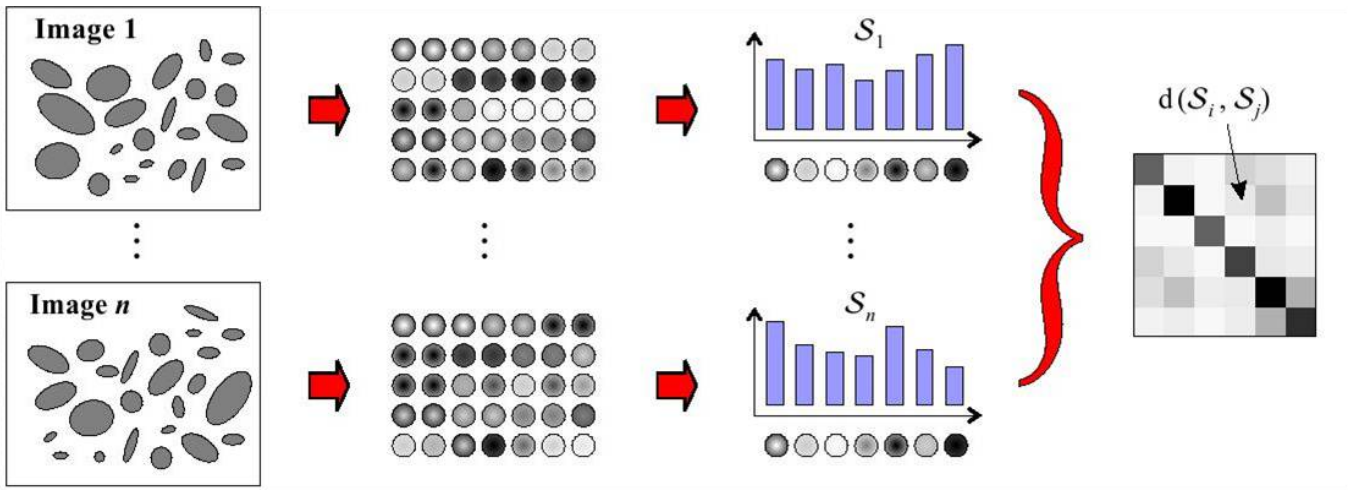
Outline

Outline

Overview

Image → Interest points → Local descriptors → Bag-of-features → Classification

- Salient image regions (interest “points”) are detected
- Regions are locally described with feature vectors
- Features are quantized or clustered
- Histograms or signatures are classified with SVMs



Detectors

Image → Interest points → Local descriptors → Bag-of-features → Classification

- We have evaluated two widely used detectors
  - Harris-Laplace — detects corners
  - Laplacian — detects blobs

- Laplacian demonstrates slightly higher performance...

Detectors

Image → Interest points → Local descriptors → Bag-of-features → Classification

- We have evaluated two widely used detectors
  - Harris-Laplace — detects corners
  - Laplacian — detects blobs

- Laplacian demonstrates slightly higher performance...
- ...the combination, however, performs even better
- The two detectors capture complementary information

Descriptors

Image → Interest points → Local descriptors → Bag-of-features → Classification

- We have evaluated three descriptors
  - SIFT — gradient orientation histogram
  - SPIN — rotation invariant histogram of intensities
  - RIFT — rotation invariant version of SIFT
- SIFT performs the best, SPIN slightly worse, RIFT seems to loose important information
- Again, combining SIFT with SPIN improves performance as those descriptors are complementary
- Adding RIFT does not help

Description invariance

Image → Interest points → Local descriptors → Bag-of-features → Classification

- We need invariance to recognize objects observed under varying conditions
- We have seen that invariance leads to information loss
- How much invariance do we need?

Description invariance

Image → Interest points → Local descriptors → Bag-of-features → Classification

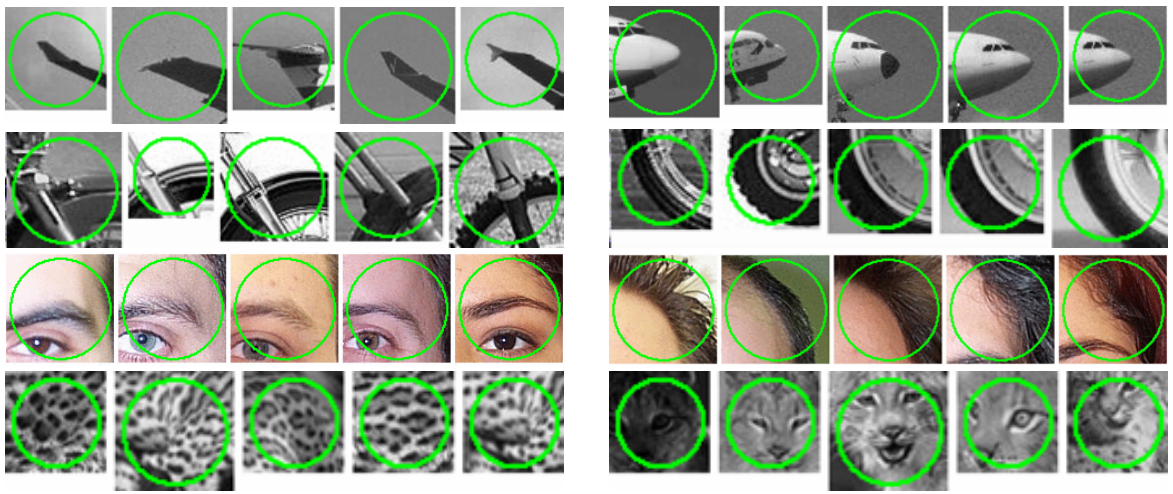
- We need invariance to recognize objects observed under varying conditions
- We have seen that invariance leads to information loss
- How much invariance do we need?

- No more than necessary
- We needed scale invariance in our experiments
- Rotation invariance helped only for UIUCTex
- We have not observed any improvement due to affine adaptation of interest “points” — object recognition is different from matching

Visual vocabulary

Image → Interest points → Local descriptors → Bag-of-features → Classification

- Bag-of-words representation has proven its usefullness in text classification
- Visual words are created by clustering the observed features



Bag-of-Features

Image → Interest points → Local descriptors → Bag-of-features → Classification

- Given a vocabulary, we can quantize the feature vector space by assigning each observed feature to the closest visual word
- Given an image, we can create a histogram of words' occurence
- Alternatively, we can cluster the set of features
- Note that there are no common underlying words in this case, the words are adapted to an image

- Note that both approaches ignore spatial relationships between features

Support Vector Machines

Image → Interest points → Local descriptors → Bag-of-features → Classification

- We use non-linear Support Vector Machines to classify histograms and signatures
- The decision function has the following form

$$g(x) = \sum_i \alpha_i y_i K(x_i, x) - b$$

- We use extended Gaussian kernels

$$K(x_j, x_k) = \exp\Big(-\frac{1}{A} D(x_j, x_k)\Big)$$

- $D(x_j, x_k)$ is a similarity measure

χ^2 kernel

- To compare histograms, we use χ^2 distance

$$D(U, W) = \frac{1}{2} \sum_{i=1}^{m} \frac{(u_i - w_i)^2}{u_i + w_i}$$

- Efficient to compute
- It is bin-to-bin measure, so common underlying words are necessary

EMD kernel

Image → Interest points → Local descriptors → Bag-of-features → Classification

- To compare signatures, we use Earth Mover's Distance

$$D(U,W) = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}\, d(u_i, w_j)}{\sum_{i=1}^{m}\sum_{j=1}^{n} f_{ij}}$$

- Requires solving a linear programming problem to determine the f_{ij} flow
- We have to define the ground distance $d(u_i, w_j)$ between features
- No vocabulary construction is necessary

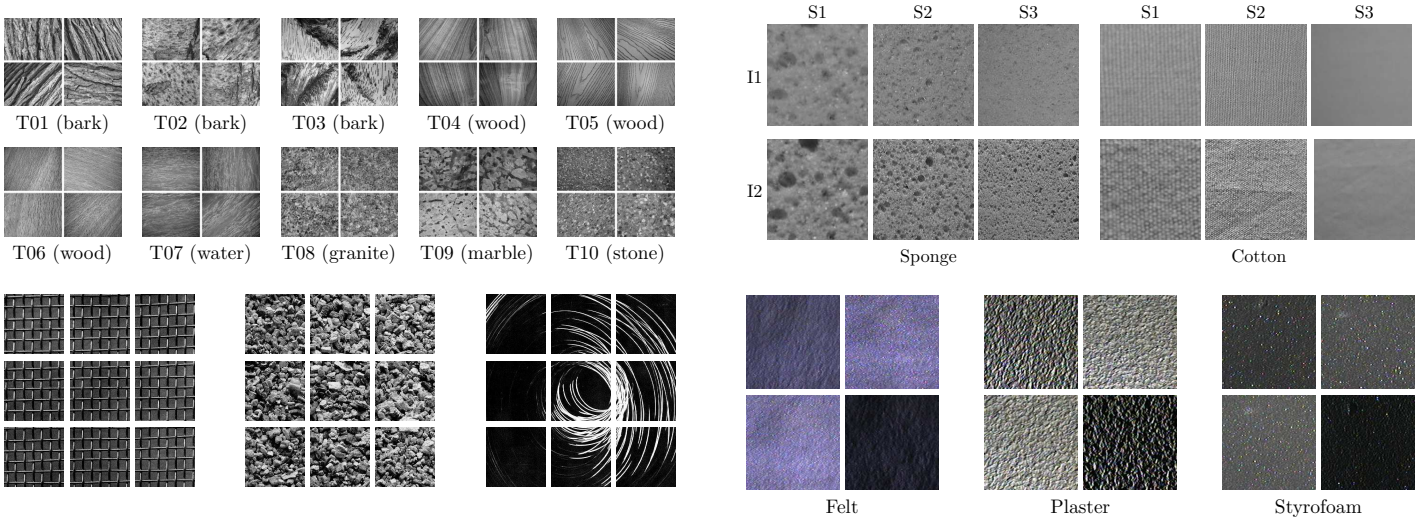
Which kernel to choose?

Image → Interest points → Local descriptors → Bag-of-features → Classification

- Both perform comparably
- EMD kernel does not require an expensive vocabulary construction — short training times
- χ^2 kernel is faster to compute — short testing times

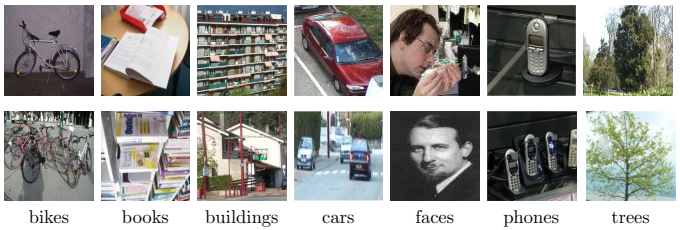
Outline

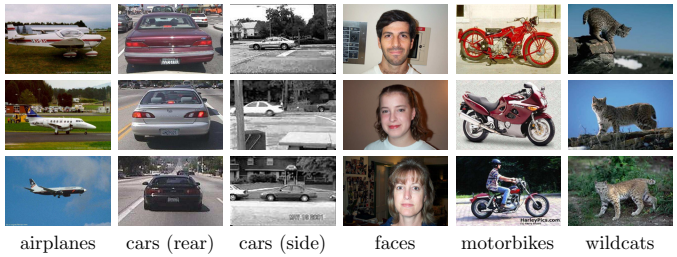
Texture datasets

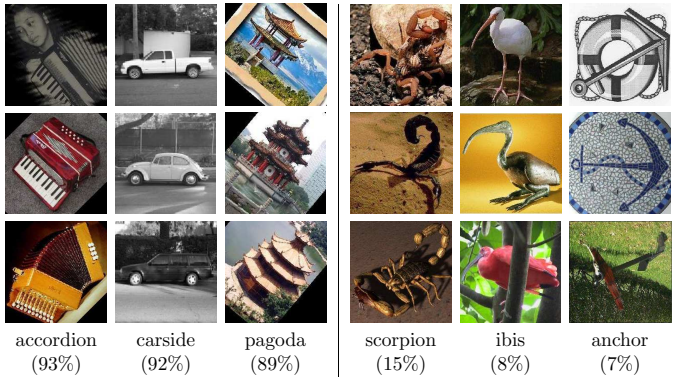


Methods	UIUCTex	KTH-TIPS	Brodatz	CUReT
our	98.3 ± 0.5	95.5 ± 1.3	95.4 ± 0.3	95.3 ± 0.4
Hayman	92.0 ± 1.3	94.8 ± 1.2	95.0 ± 0.8	98.6 ± 0.2
Lazebnik	96.4 ± 0.9	91.3 ± 1.4	89.8 ± 1.0	72.5 ± 0.7
VZ-joint	78.4 ± 2.0	92.4 ± 2.1	92.9 ± 0.8	96.0 ± 0.4
G. Gabor	65.2 ± 2.0	90.0 ± 2.0	87.9 ± 1.0	92.4 ± 0.5

Object category datasets

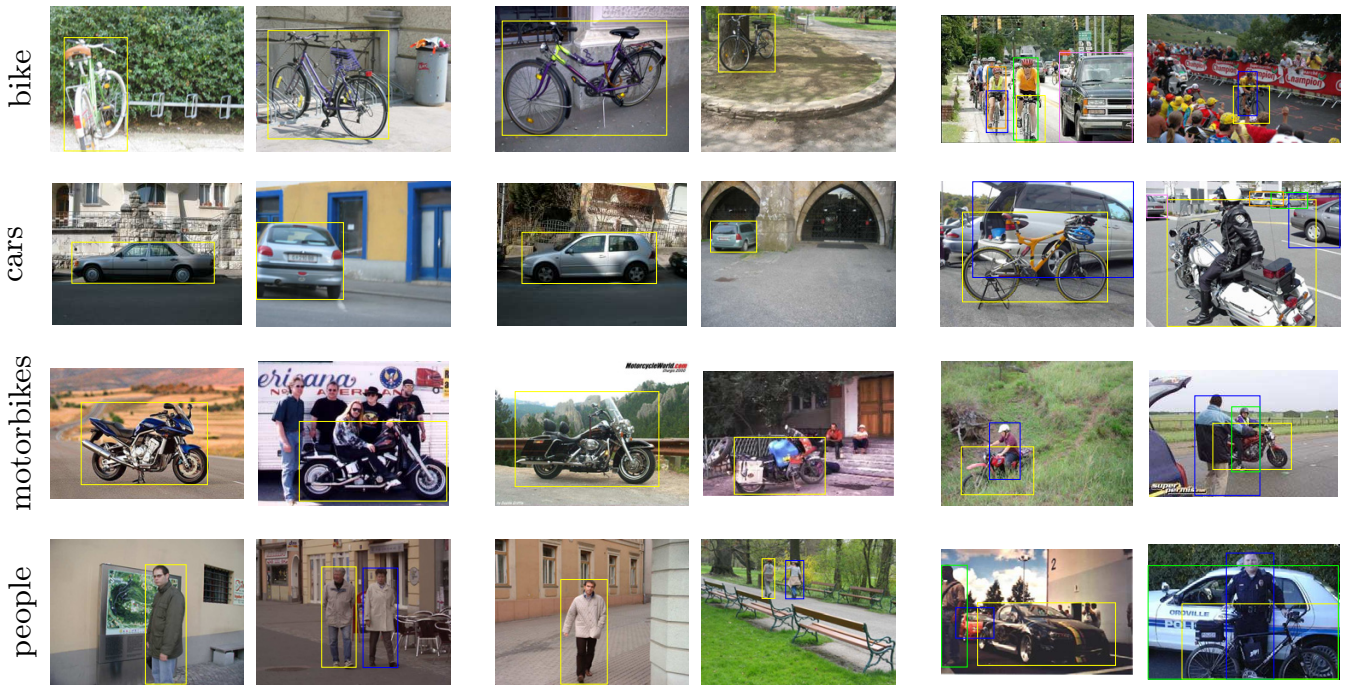






Methods	Xerox7	CalTech6	Graz	Pascal		CalTech101
				test set1	test set2	
our	94.3	97.9	90.0	92.8	74.3	53.9
others	82.0	96.6	83.7	94.6	70.5	43

PASCAL VOC challenge 2005 dataset



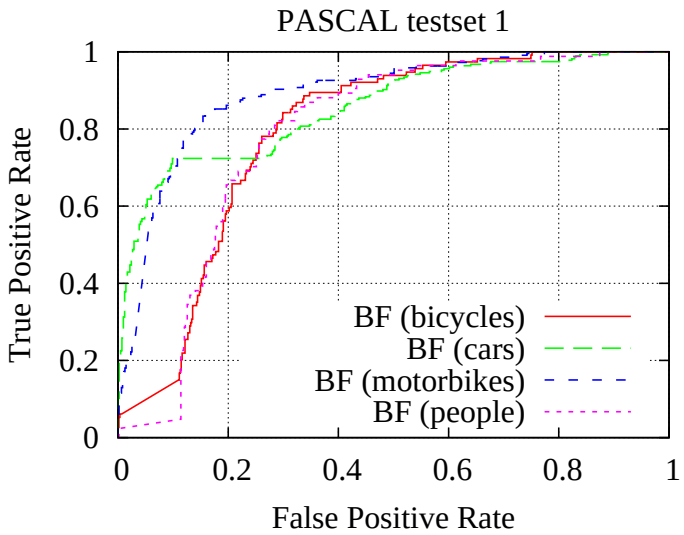
- Note the object annotations

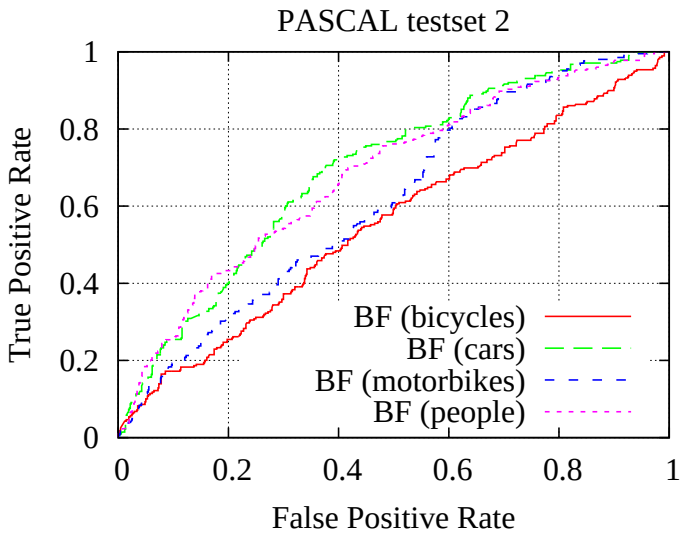
Outline

Overview

- PASCAL VOC challenge 2005 dataset comes with detailed object annotation
- Using the provided bounding boxes we can approximately separate foreground and background features
- We perform experiments on five meta-datasets
  - FF — foreground features
  - BF — background features
  - AF — all features (FF + BF)
  - AF-CONST — foreground features with static scene background features
  - AF-RAND — foreground features with background features of a random

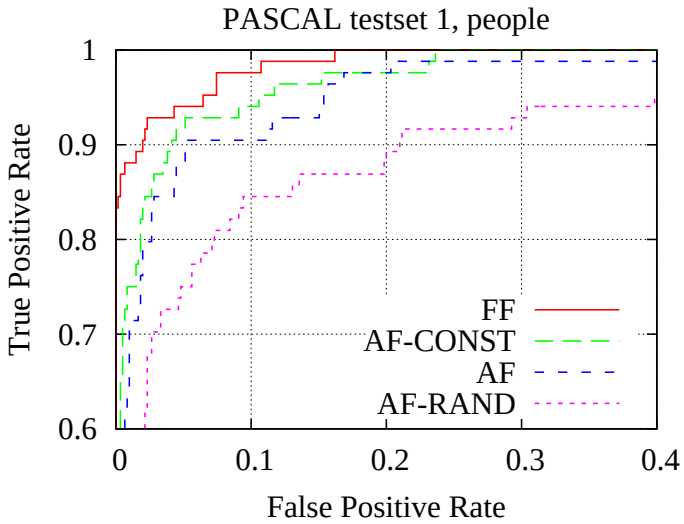
How important is the context?

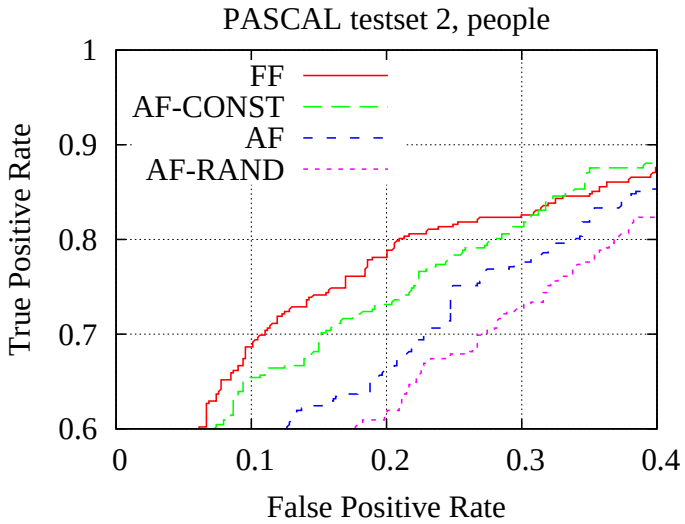




- We train and test on BF — still better than random method
- Background features carry a significant ammount of context information

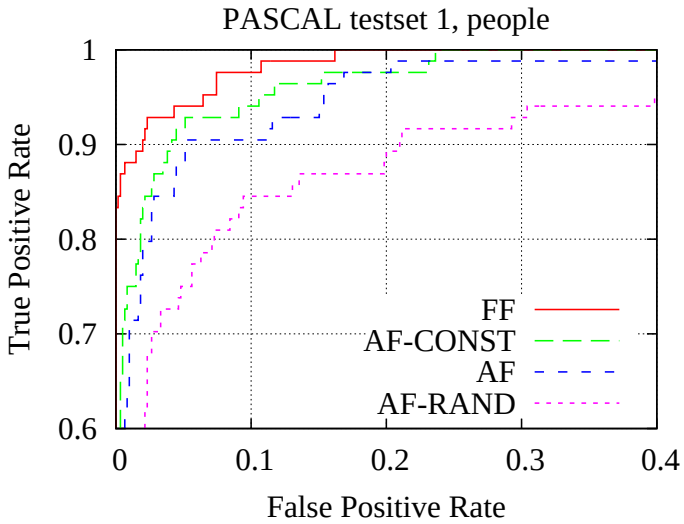
Can we use this information?

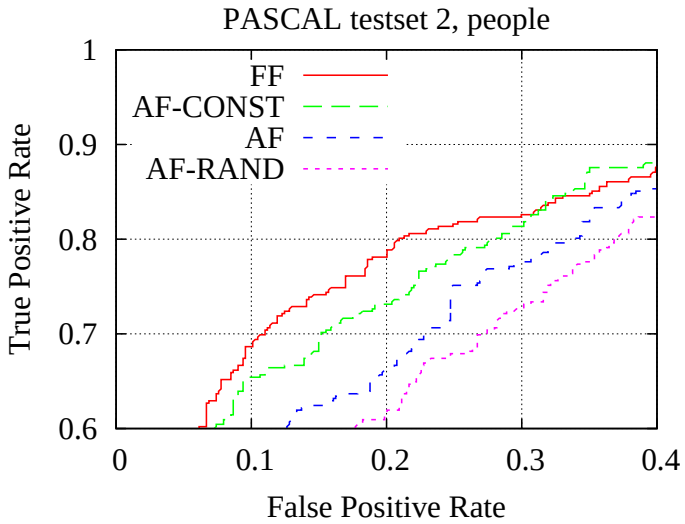




- Training and testing on FF gives better results than AF
- Due to background clutter we cannot use the context information to improve the classification results

Can we deal with background clutter at all?



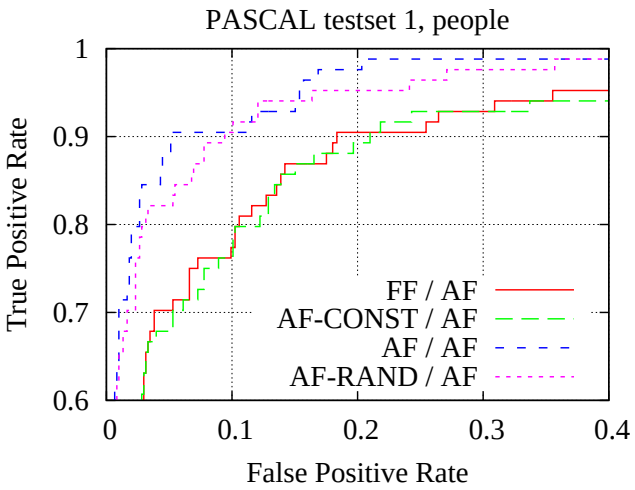


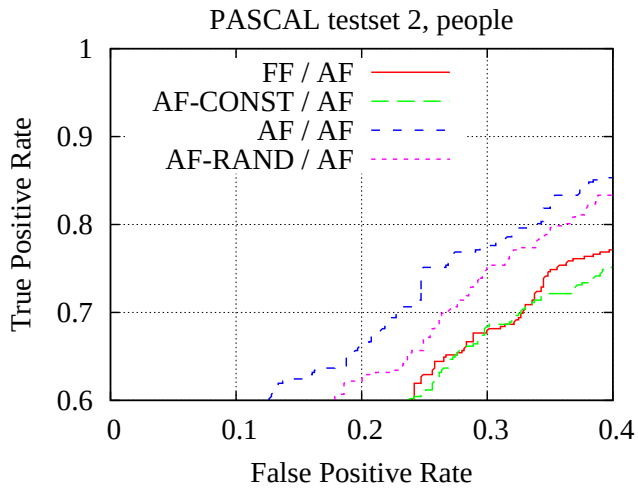
- Training and testing on AF-CONST is still better than AF and often close to FF
- We can easily deal with static background

Outline

It's easy to get biased

- What happens if the test images are not represented well by the training set?





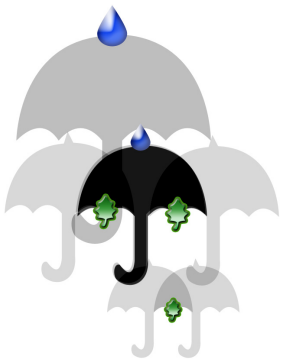
- Training on AF and AF-RAND is significantly better than on FF or AF-CONST
- One should not train on too easy examples, it is better to choose too hard ones

Summary

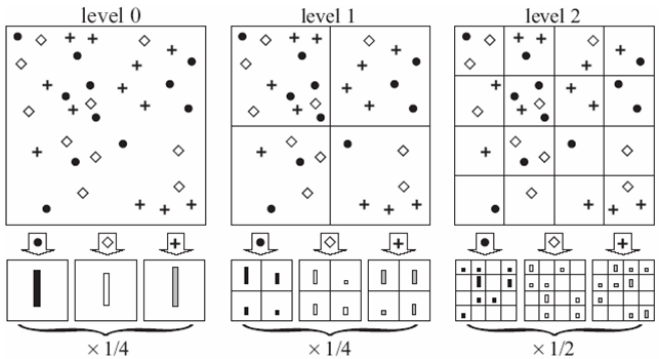
- We have created an effective image classification framework that outperforms the state-of-the-art
- We have evaluated the parameters of the framework on a wide range of datesets and were able to deliver general design conclusions
- We have evaluated the influence that the background has on bag-of-features methods

Future work

- Spatial Weigting for object recognition



- Spatial Pyramid Matching for scene classification



Thank you for your attention

I will be glad to answer your questions