XRCE Segmentation method

Gabriela Csurka, Florent Perronnin and Yan Liu

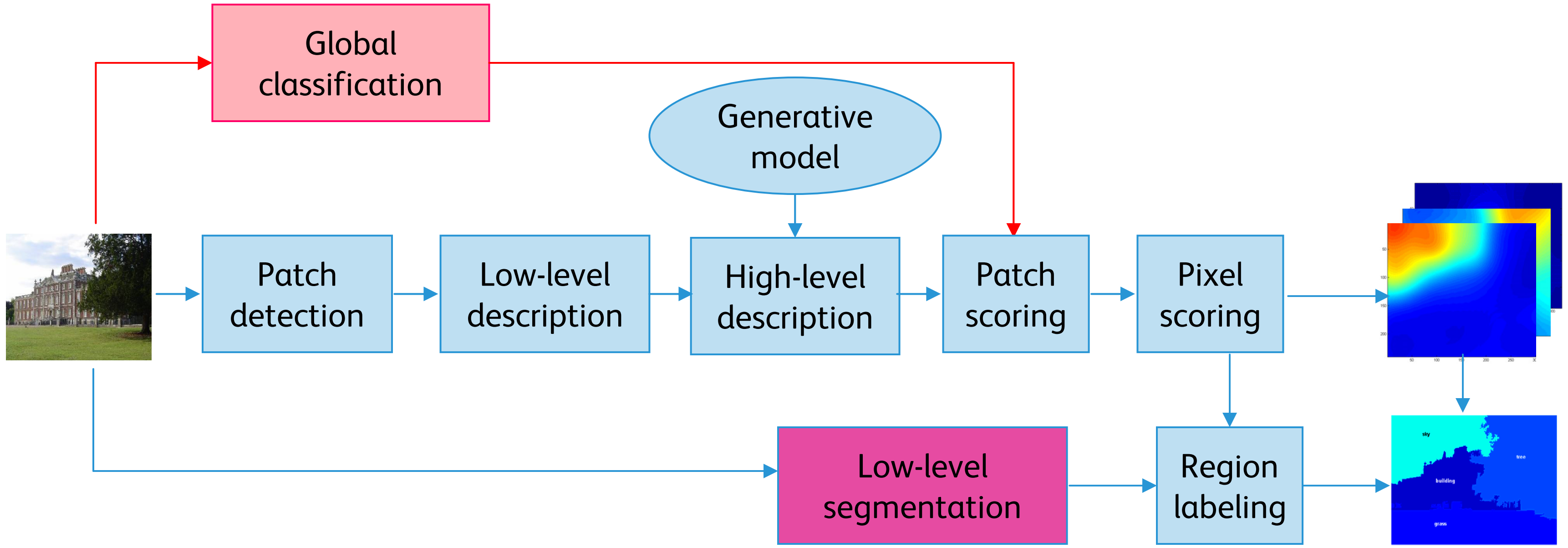
Xerox Research Center Europe

6 chemin de Maupertuis

38240 Meylan, France

xerox

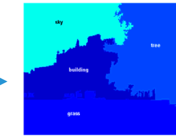
The main idea*



* *A Simple High Performance Approach to Semantic Segmentation*, G. Csurka and F. Perronnin, **BMVC 2008.**

Low-level representation

Patch detection → **Low-level description** → High-level description → Patch scoring → Pixel scoring → Region labeling →



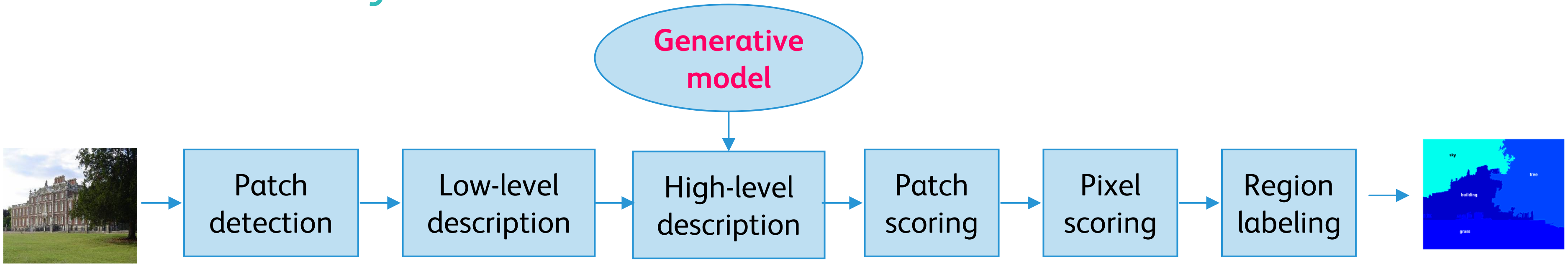
- Patches are extracted on regular grids at 5 different scales.

- Two types of features were considered:
  - Local RGB statistics (mean and standard deviation).
  - Local histograms of gradient orientations (SIFT-like).

- In both cases the dimensionality was reduced to 50 (PCA).

- They are handled independently and fused at late stages.

Visual Vocabulary with a GMM



- Modeling the visual vocabulary in the feature space with a GMM:

$$p(x_t \mid \lambda) = \sum_{i=1}^{N} w_i p_i(x_t \mid \lambda) \qquad \text{with} \qquad p_i(x_t \mid \lambda) = \mathcal{N}\left(x_t \mid \mu_i, \Sigma_i\right)$$

- The parameters λ are estimated by EM algorithm maximizing the log-likelihood on the training data *X={ x_t }* *:

$$\log p(X \mid \lambda) = \sum_t \log p(x_t \mid \lambda)$$

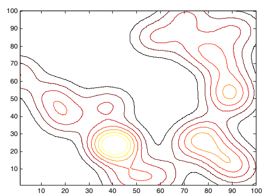
* ***Adapted Vocabularies for Generic Visual Categorization***, **F. Perronnin, C. Dance, G. Csurka and M. Bressan, ECCV 2006.**

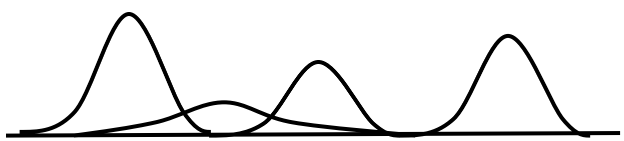
Visual Vocabulary with a GMM

Generative model

Patch detection → Low-level description → **High-level description** → Patch scoring → Pixel scoring → Region labeling →

- Occupancy probability of a patch x_t : $$\gamma_i(x_t) = p(i \mid x_t, \lambda) = \frac{w_i p_i(x_t \mid \lambda)}{\sum_{j=1}^{N} w_j p_j(x_t \mid \lambda)}$$

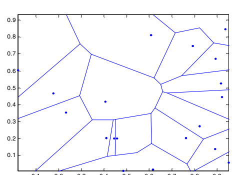


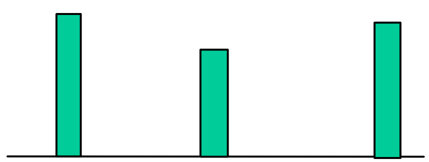


Soft assignment

$$\sum_t \underbrace{[\gamma_1(x_t), \gamma_2(x_t), \ldots, \gamma_N(x_t)]}_{v_t}$$

BOV



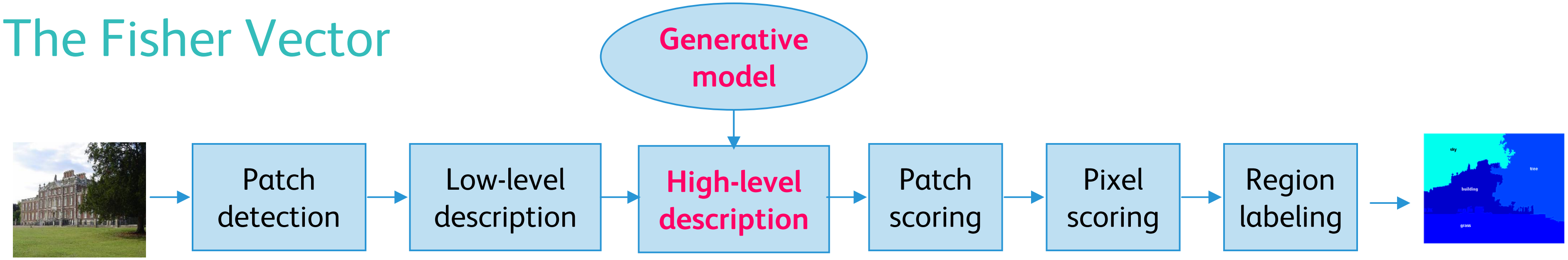


Hard assignment

$$\sum_t \underbrace{[0,0,\cdots,1,\cdots,0]}_{v_t = 1_{\{k \mid x_t \in C_k\}}}$$

xerox

The Fisher Vector



- Given a generative model with parameters λ (GMM)
  - we consider the gradient vector

$$\nabla_\lambda \log p(x_t \mid \lambda)$$

  - and deduce the following formulas for the partial derivatives*:

$$\frac{\partial \log p(x_t \mid \lambda)}{\partial w_i} = \left[\frac{\gamma_i(x_t)}{w_i} - \frac{\gamma_1(x_t)}{w_1} \right]$$

$$\frac{\partial \log p(x_t \mid \lambda)}{\partial \mu_i^d} = \gamma_i(x_t) \left[\frac{x_t^d - \mu_i^d}{(\sigma_i^d)^2} \right]$$

$$\frac{\partial \log p(x_t \mid \lambda)}{\partial \sigma_i^d} = \gamma_i(x_t) \left[\frac{(x_t^d - \mu_i^d)^2}{(\sigma_i^d)^3} - \frac{1}{\sigma_i^d} \right]$$

* ***Fisher Kernels on Visual Vocabularies for Image Categorization*, F. Perronnin and C. Dance, CVPR 2007.**

The Fisher Vector (cont)

Generative model

Patch detection → Low-level description → **High-level description** → Patch scoring → Pixel scoring → Region labeling →

- High level representation of the patch (***Fisher Vector)***

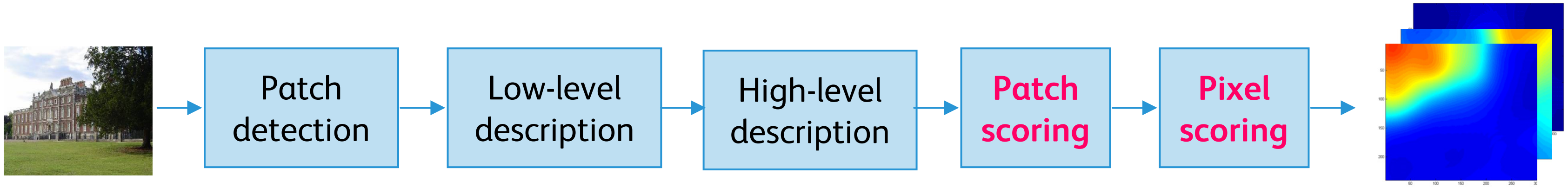
$$f_{t=}\left[\ldots,\frac{\partial \log p(x_t \mid \lambda)}{\partial \mu_i^d},\ldots,\frac{\partial \log p(x_t \mid \lambda)}{\partial \sigma_i^d},\ldots\right]$$

Notes:

- the Fisher Vector describes in which direction the parameters of the model should be modified to best fit the data
- the gradient with respect to the mixture weights does not contain significant extra information (we ignore them)
- hence, dimension = 2 x D x N, where D is the dimension of low level features (50) and N is the number of Gaussians (128)
- very sparse, as only a few number of components *i* (typically < 5) have a non-negligible "occurrence probability " $\gamma_i(x_t)$ for a given *t*

xerox

Patch and Pixel Scoring



- Patch classifiers (PC) were:
  - trained on labeled Fisher Vectors (using masks and bounding boxes)
  - Linear Sparse Logistic Regression scores transformed in probabilities:

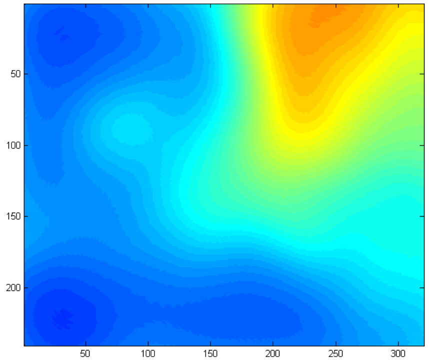
$$p(c \mid f_t) = \frac{1}{1 + \exp(-\alpha^T f_t + \beta)}$$

- The class posterior at pixel level is the weighted average of the class posteriors of patches containing the pixel.

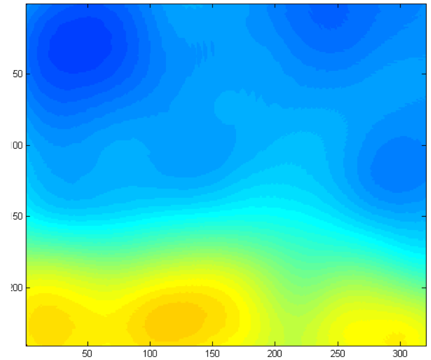
$$p(c \mid z) = \frac{\sum_t p(c \mid f_t)\mathcal{N}(z \mid m_t, C_t)}{\sum_t \mathcal{N}(z \mid m_t, C_t)}$$

- This leads to one class probability map (P_c) per class.

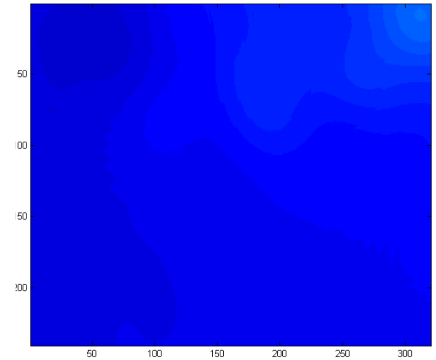
Examples of class probability maps



Tree Map

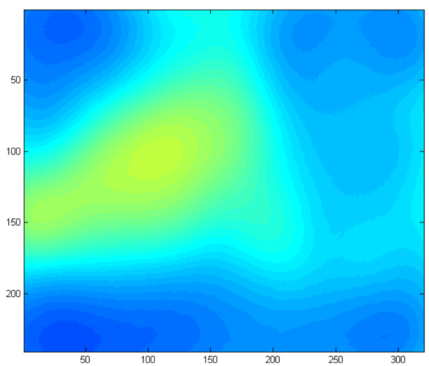


Grass Map

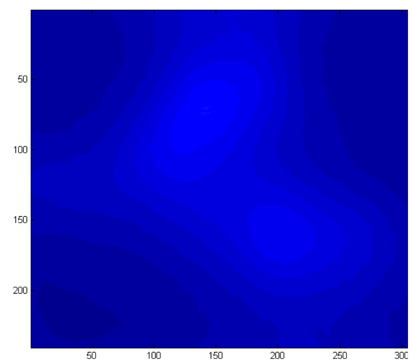


Dog Map

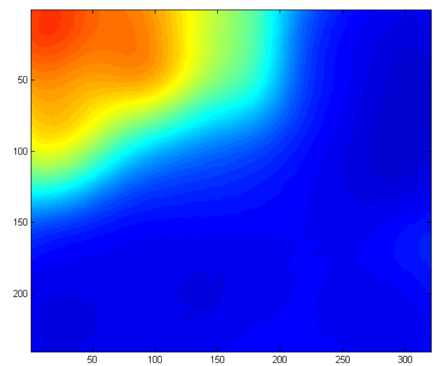
Pixel labeling



Building Map

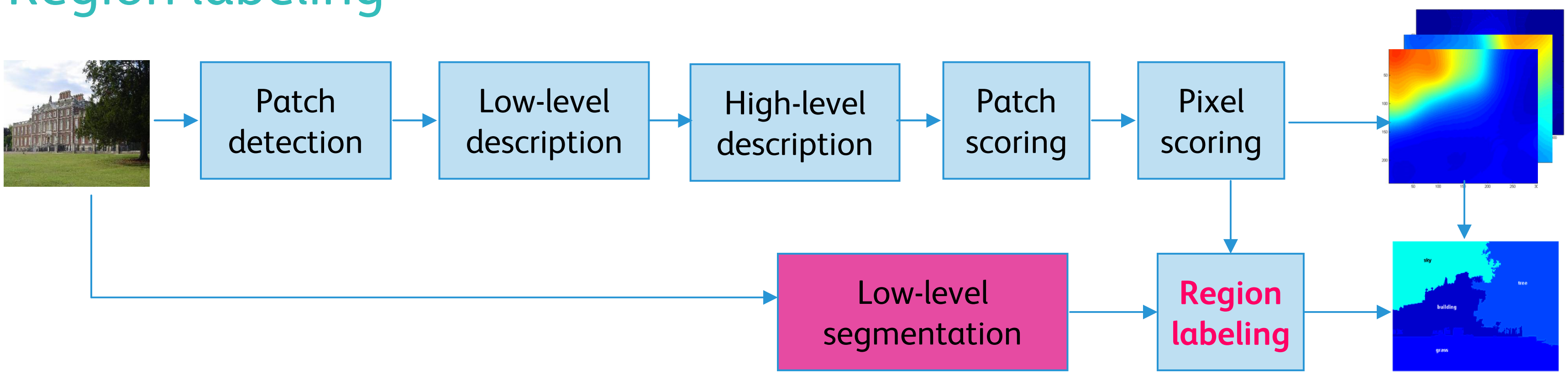


Boat Map



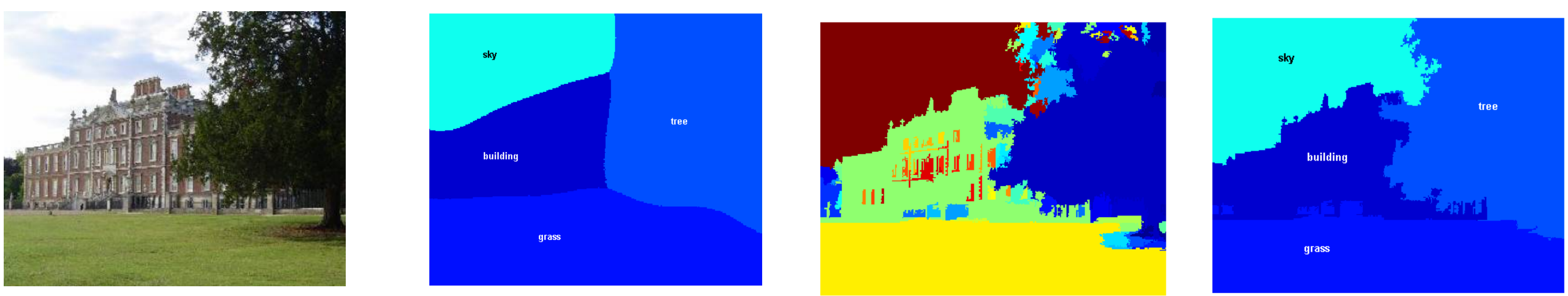
Sky Map

Region labeling



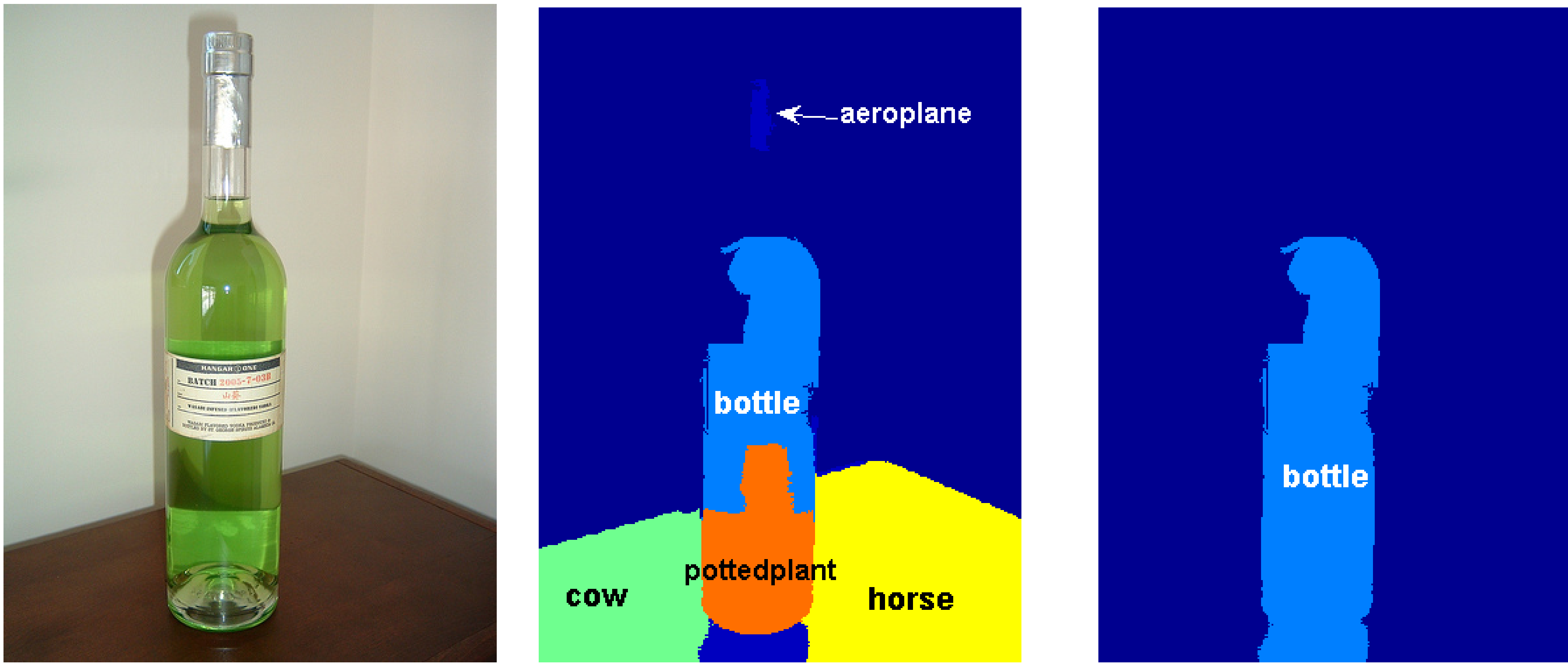
- Class probabilities are averaged over low level (Mean Shift) images segments and each segment R is labeled with:

$$c^* = \begin{cases} \arg\max_c \left(P_c(R)\right) & \text{if } \ P_c(R) > \text{Thr} \\ \text{background} & \end{cases}$$

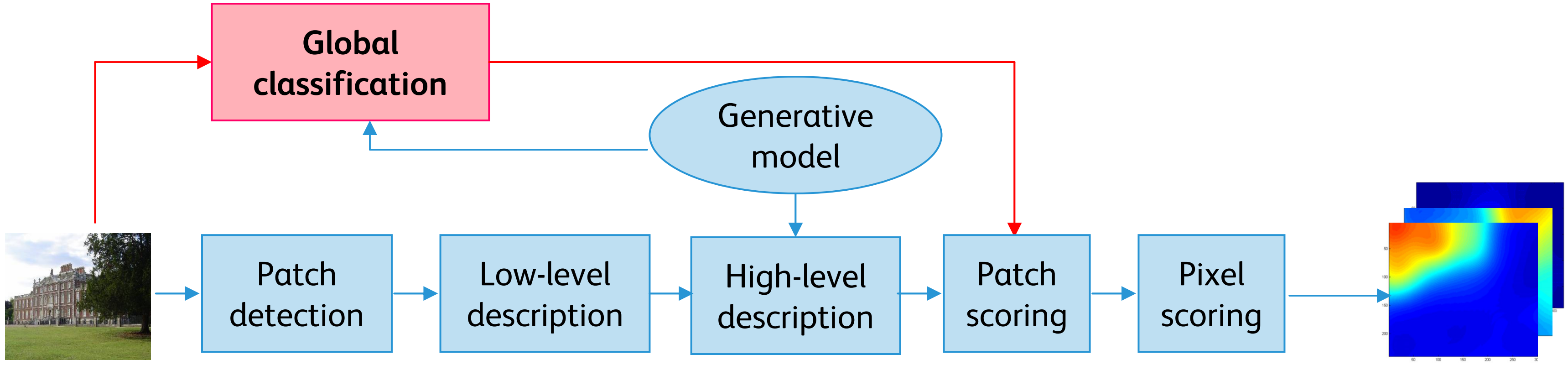


However

- Using all probability masks might introduce many local False Positives !!



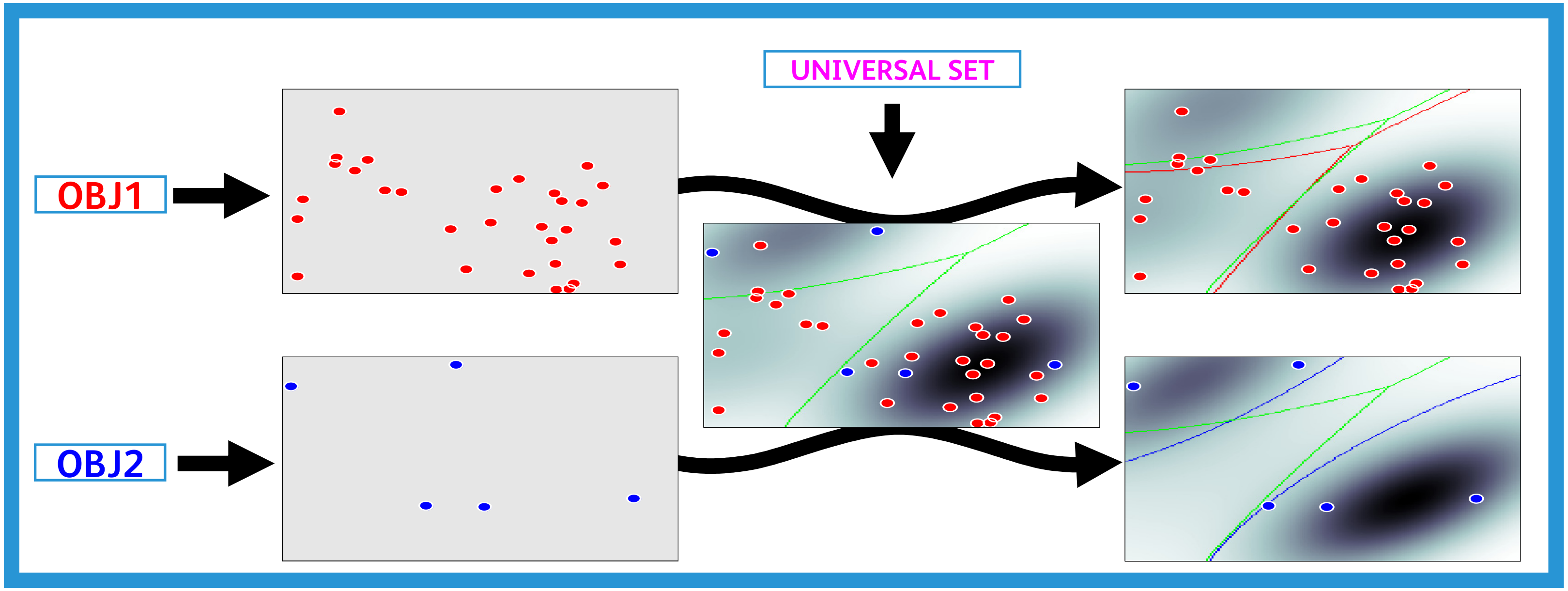
Fast Rejection with Global Classification



- A visual categorizer is trained on weakly labeled data to detect visual concepts/objects (any classifier can be used) and transform scores in probabilities (image level prior).
- Then image level prior (ILP) is used to fast reject “non relevant” probability maps :
- ☺ Reduce computational cost.
- ☺ Decrease false positive regions.
- ☹ Prevent the discovery of objects rejected by the global classifier.

Global Image Classifier (used in the Classification Task)

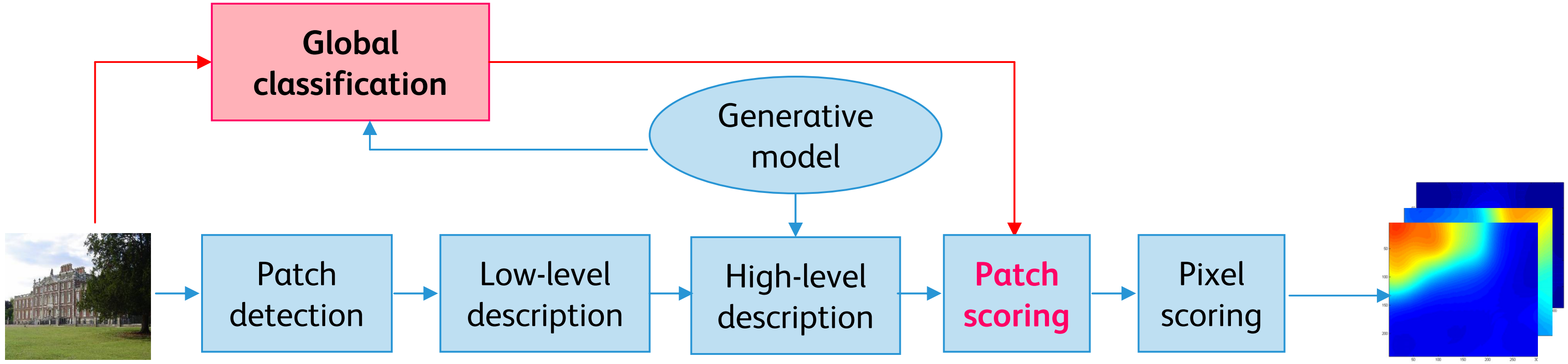
- MAP adaptation of the Universal GMM (Vocabulary) for each image.
- Fast kernel computation between adapted GMMs (approximate Probability Product Kernel),
- One-against-all Kernel Sparse Logistic Regression (KSLR) to classify.



* A Similarity Measure between Unordered Vector Sets with Application to Image Categorization, Y. Liu and F. Perronnin, CVPR 2008.

xerox

Modified Patch Classifier (MPC)

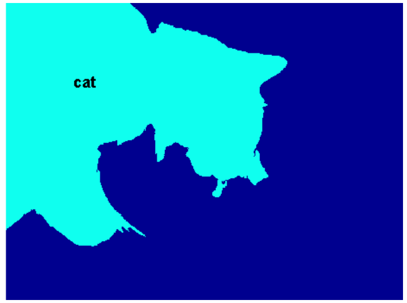


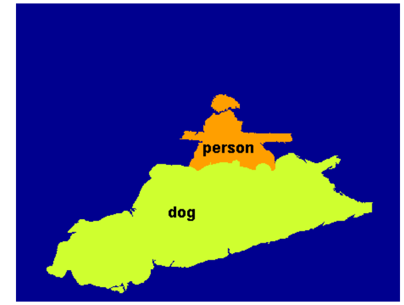
- Main idea:
  - Global image classifier rejects the improbable context/background.
  - Patch classifier separates the “object” from its usual context.
- How:
  - Train the patch classifier only with images containing the object:
    - positive patches from object masks (segments and bounding boxes)
    - negative patches from the inversed masks

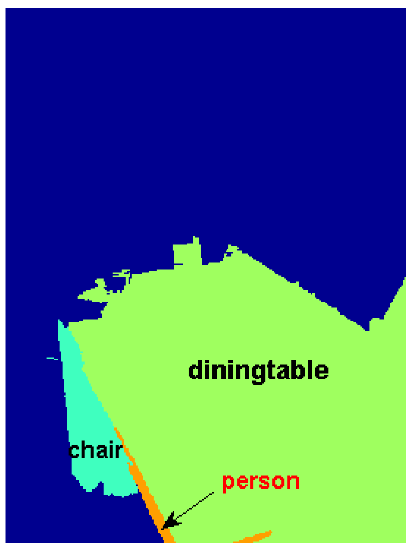
Note: In the challenge both type of patch classifiers (PC and MPC) were used and the four (2 color and 2 texture) corresponding probability maps averaged.

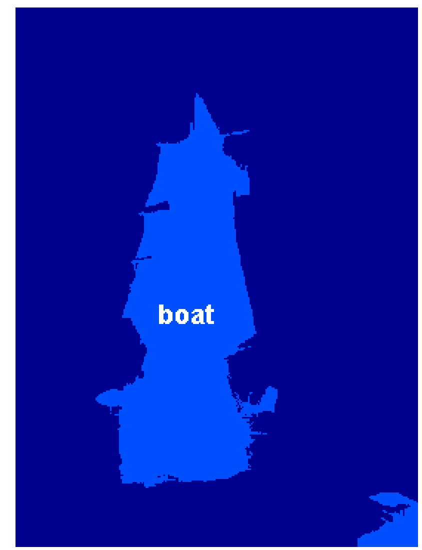
xerox

Examples where it “rather” succeeded



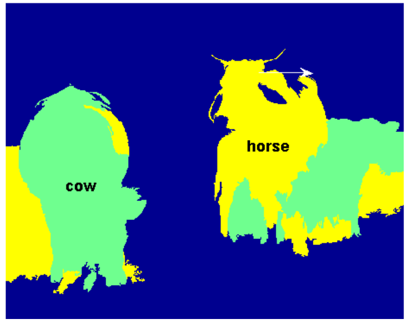


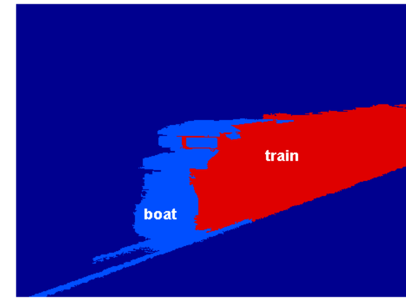




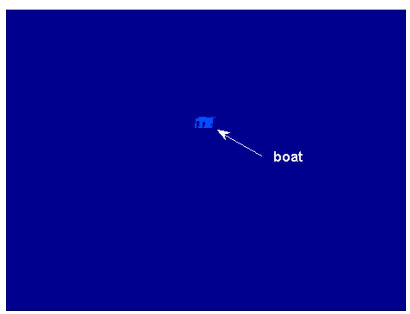
Examples where it “had difficulties”

- Confused classes

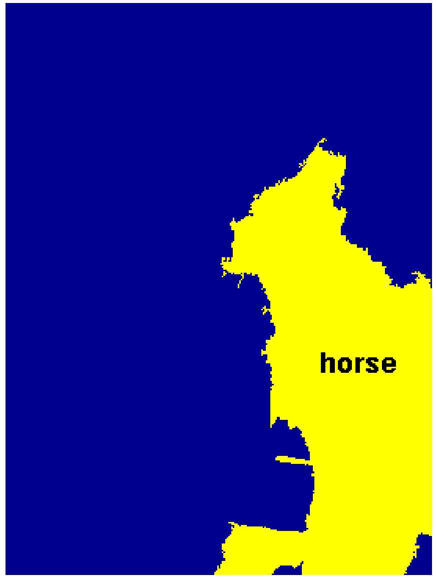


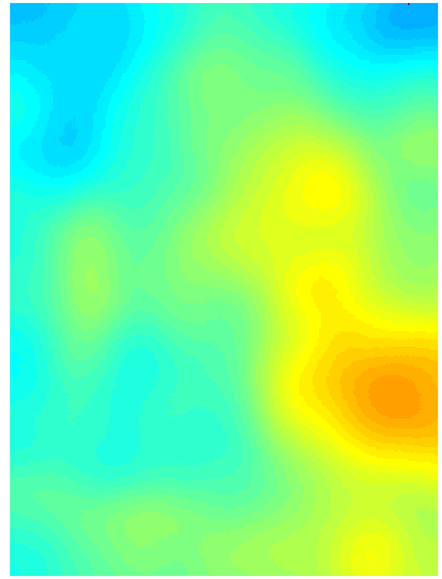


- Under and over estimation (too low or too high probability value in P_c)

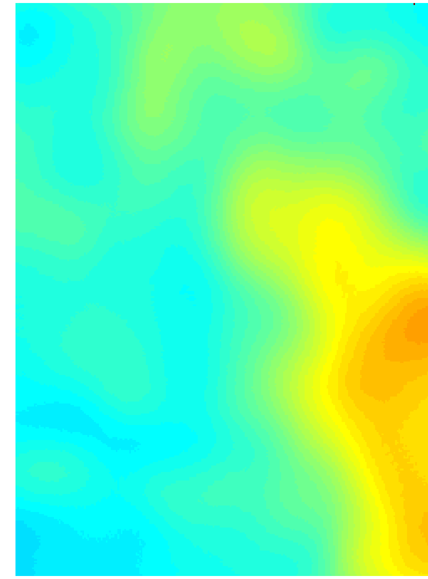


Examples where it failed (due to fast rejection ???)

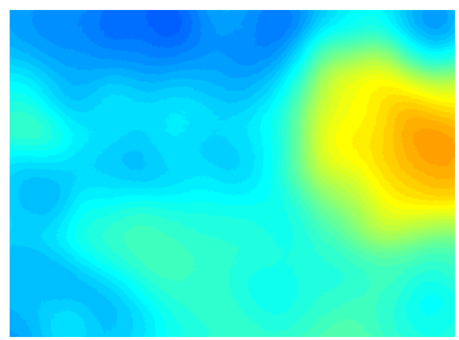




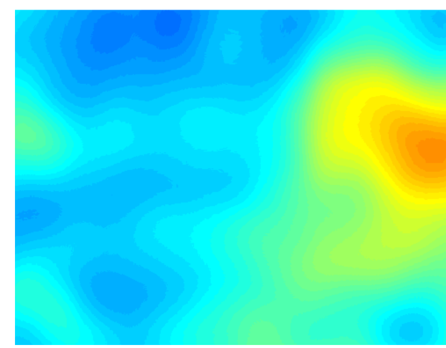
Horse Map



Cat Map – Not considered



Dog Map



Cat Map – Not considered

Discussion

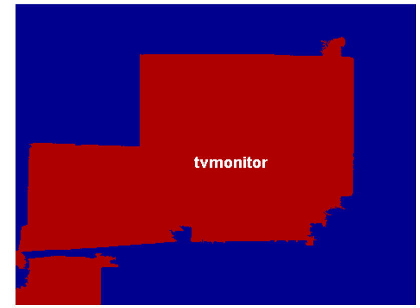
- Its strengths:
  - Simplicity
    - Simple patch classification with high level descriptors
    - Combined with Low level segmentation and ILP
  - Low computational cost:
    - The most costly bit (Mean Shift segmentation 30 s vs 1-2 s for the rest) can be avoided for many applications (where no need for accurate object boundaries).
  - Can be a good starting point for further processing or integration in more complex system (future research)

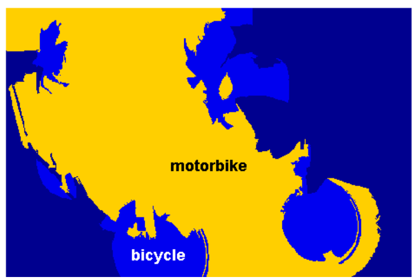
- Its limitations
  - The method is maybe too simple to give excellent results:
    - Still remains at the “bag-of-visual word” level.
    - No geometry, no knowledge of shape, no global object model.
    - Not suitable for object detection (see next slide).

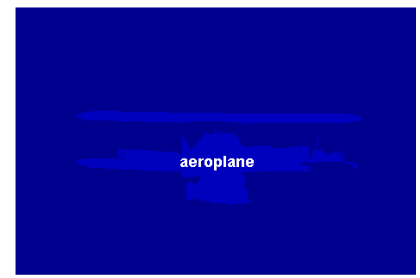
Object Detection Task

- Indeed the approach is not well suited for detection (XRCE_det)
  - Not able to separate multiple instances or fuse separated object parts, …





- XRCE_Det had low (7.1 %) detection rate compared to the winner (22.6 %)
- However, when segmentation from detection
  - we got 18.9 % , and they got 3.7 % segmentation accuracy
  - even with bounding boxes (both input and output), it was the third best segmentation result (not counting UIUC_CMU which used their own training data).

BACKUP SLIDES

GMM estimation - EM algorithm

- Definition: $$\gamma_i(x) = \frac{w_i p_i(x|\lambda)}{\sum_{j=1}^N w_j p_j(x|\lambda)} .$$

- Universal model: MLE

$$\hat{w}_i^u = \frac{1}{T}\sum_{t=1}^T \gamma_i(x_t) ,$$
$$\hat{\mu}_i^u = \frac{\sum_{t=1}^T \gamma_i(x_t)x_t}{\sum_{t=1}^T \gamma_i(x_t)} ,$$
$$\hat{\Sigma}_i^u = \frac{\sum_{t=1}^T \gamma_i(x_t)x_t x_t'}{\sum_{t=1}^T \gamma_i(x_t)} - \hat{\mu}_i^u \hat{\mu}_i^{u\prime} .$$

- Adapted image model: MAP

$$\hat{w}_i^a = \frac{\sum_{t=1}^T \gamma_i(x_t) + \tau}{T + N \times \tau} ,$$

relevance factor

$$\hat{\mu}_i^a = \frac{\sum_{t=1}^T \gamma_i(x_t)x_t + \tau\mu_i^u}{\sum_{t=1}^T \gamma_i(x_t) + \tau} ,$$
$$\hat{\Sigma}_i^a = \frac{\sum_{t=1}^T \gamma_i(x_t)x_t x_t' + \tau[\Sigma_i^u + \mu_i^u \mu_i^{u\prime}]}{\sum_{t=1}^T \gamma_i(x_t) + \tau} - \hat{\mu}_i^a \hat{\mu}_i^{a\prime} .$$

- Advantages of adapted GMMs:
  - MAP more robust than MLE when training data is scarce
  - MAP faster than MLE to train (smaller number of EM iterations required)

Kernel computation: PPK

Formula:

$$K^{\rho}_{ppk}(p,q) = \int_{x\in\Omega} p(x)^{\rho} q(x)^{\rho} dx \ .$$

ρ=1, Expected Likelihood Kernel
ρ =0.5, Bhattacharyya Kernel

Existing approximations [JK03]:

$$K^{\rho}_{ppk}(p,q) \approx \sum_{i=1}^{N}\sum_{j=1}^{M} \alpha_i \beta_j K^{\rho}_{ppk}(p_i, q_j)$$

Our proposed **MAP_OTO**:

$$K^{\rho}_{ppk}(p,q) \approx \sum_{i=1}^{N} \alpha_i \beta_i K^{\rho}_{ppk}(p_i, q_i)$$

With and without Fast Rejection – Pascal VOC 2007

Method	BOV	FV
No global rejection	0.12	0.15
Using patches from all images to train (PC)	0.19	0.21
Using patches from images containing the object to train (MPC)	0.24	0.26