Unsupervised scene adaptation for faster multi-scale pedestrian detection

Speaker
Federico Bartoli[1]

Giuseppe Lisanti[1], Svebor Karaman[1], Andrew D. Bagdanov[2] and Alberto Del Bimbo[1]

[1] MICC (Media Integration and Communication Center) - University of Florence, Italy
{firstname.lastname}@unifi.it

[2] CVC (Computer Vision Center) - Autonomous University of Barcelona, Spain
bagdanov@cvc.uab.es

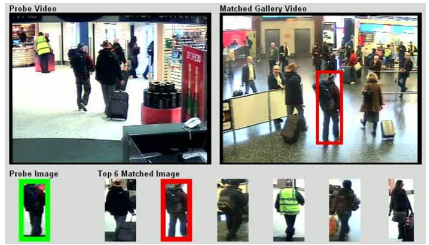
Real-time Pedestrian Detection

Application contexts

1. Video Surveillance
2. Tracking
3. People Re-identification
4. Action Recognition

Main critical factors

1. Changes of scale and strong view-point dependency
   - Different *target* locations can produce high scale changes
   - Lost of scene depth information in the image
2. Variability:
   - Different person poses (e.g. front or side view)
   - Changes in illumination intensity
3. Scene complexity
   - *Indoor* or *Outdoor*
   - *Clutter*, *crowd* and partial occlusion

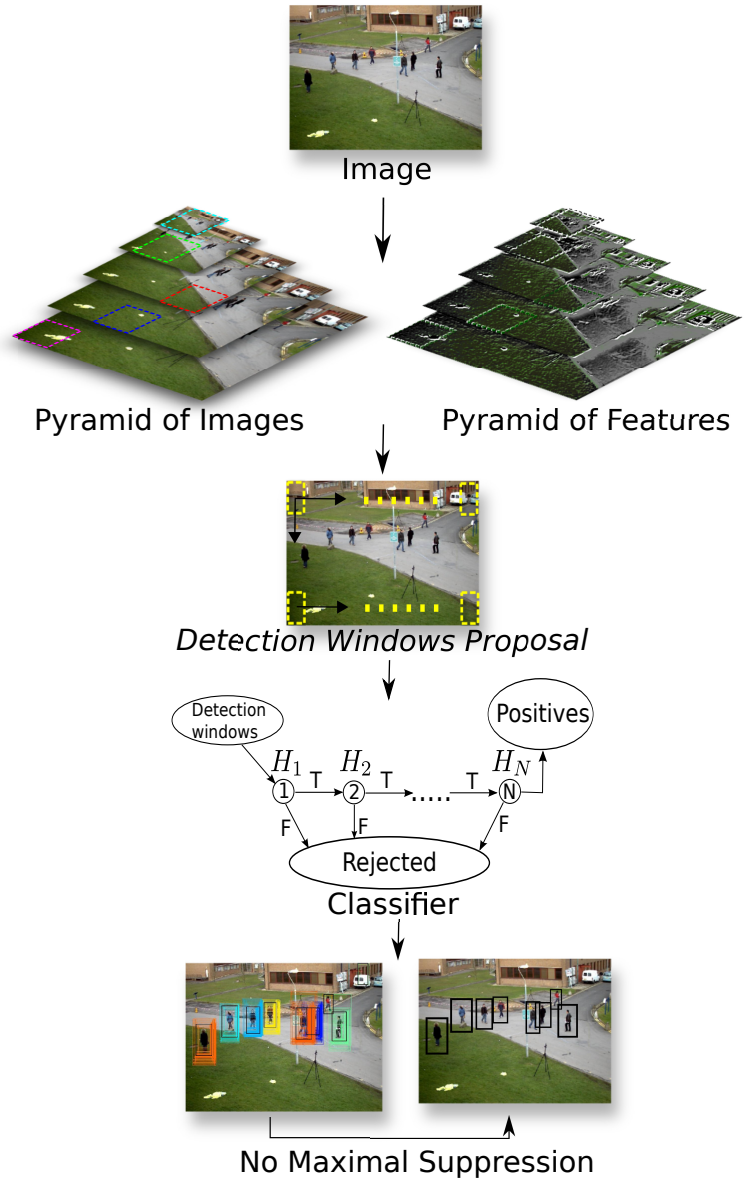


Standard execution pipeline of a multi-scale pedestrian detector

Four principal phases

- Each perform a specific task:

1. Feature Extraction on Pyramid of Image
2. Detection Windows Proposal: Sparse or Dense sampling
3. Classification: Boosting, SVM
4. Non Maximal Suppression

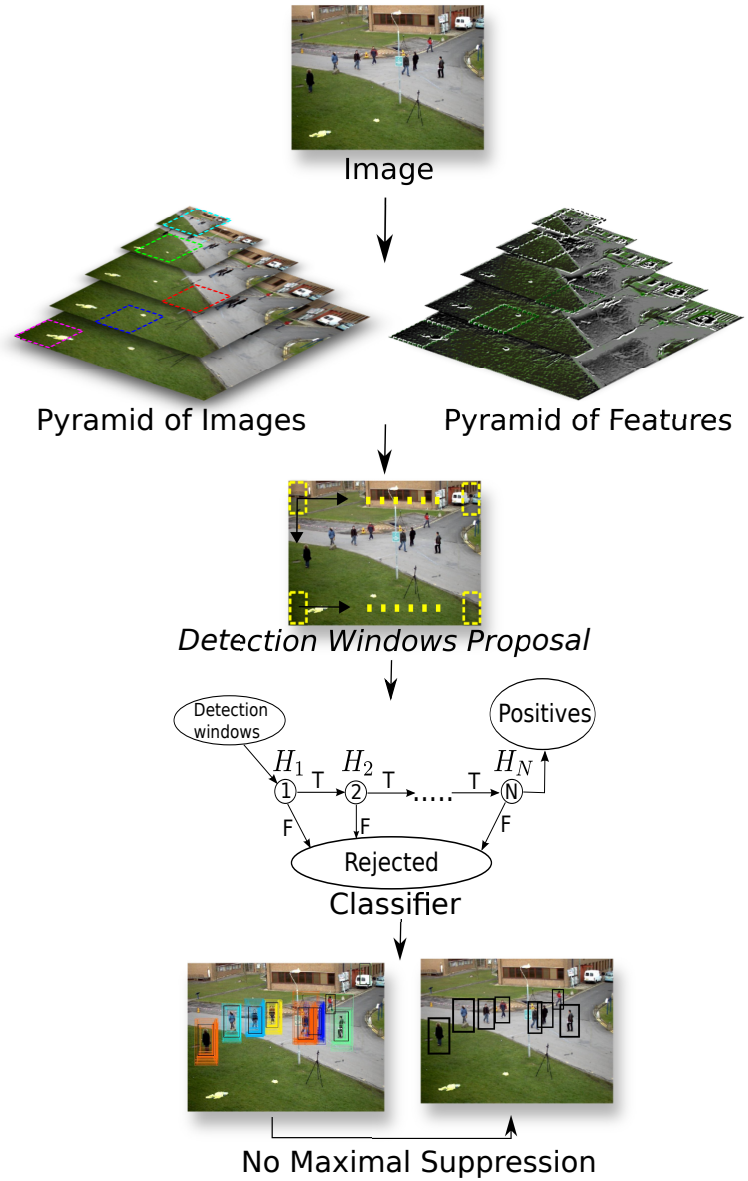


Standard execution pipeline of a multi-scale pedestrian detector

Four principal phases

- Main bottlenecks:

1. Feature Extraction on Pyramid of Image
   Channel features [Dollar'14]
2. Detection Windows Proposal: Sparse or Dense sampling
   Scene adapted detection windows proposal
3. Classification: Boosting, SVM
   Soft cascade approximation
4. Non Maximal Suppression



Faster multi-scale pedestrian detection

Question

How to increase the speed of a pre-trained pedestrian detector on a scene?

Framework Proposed

- Speed up the detection process of a Soft-Cascade pedestrian detector
- No a priori information about the scene required
- All learning done by mining statistics about the detector operating on the scene
- Exploit only ROS (Region of Support) information to build the models
- Strategies:
  1. **Linear Cascade Approximation**: acts on classifier domain, for each sample estimate a final score without calculating all stages
  2. **Generative model** for candidate window proposal: acts on pyramid domain, modelling the scene-dependent statistics of detection windows in terms of both location and scale

- The result is a significant reduction in the total number of stages evaluation required in the soft cascade detection process

Linear Cascade Approximation

Soft Cascade Architecture

Let $x \in \mathbb{R}^D$ be a sample to evaluate and $Y \in \{-1, 1\}$ its class label:

- **Classifier**: $H(x) = \sum_{k=1}^{T} f_k(x)$, where $f_k : \mathbb{R}^D \longrightarrow \mathbb{R}$ is a stage computation
- Partial Score: $H_t(x) = \sum_{k=1}^{t} f_k(x)$ the sum of the first t stage scores
- x is **classified positive** $(Y = 1) \iff \Psi\left(H_t(x), \theta_t\right) \geq 0 \quad \forall t \in [1, T]$
  where Ψ is a *stopping criterion* and $\{\theta_t\}$ are each stage rejection thresholds.

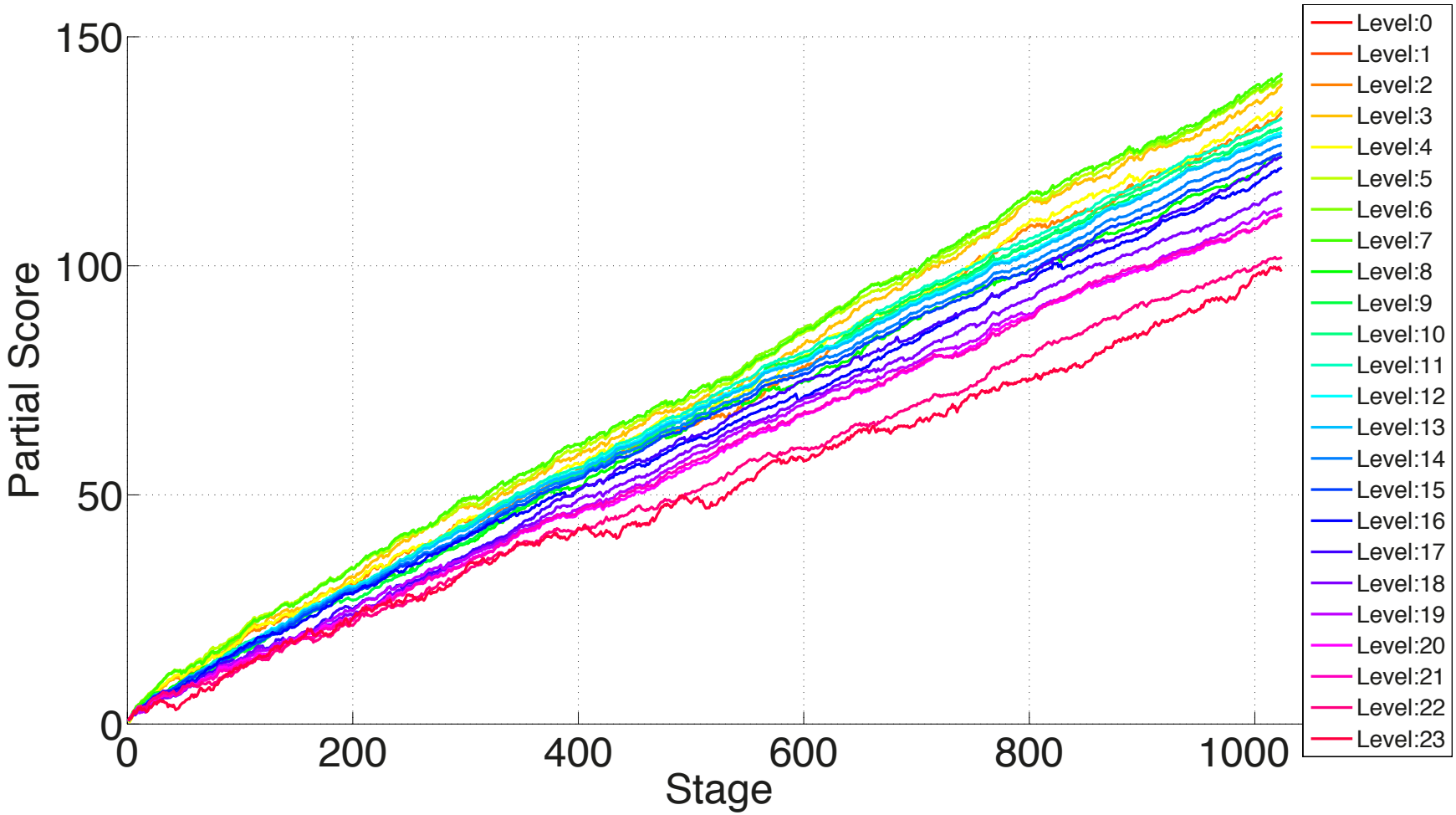
Linear Cascade Approximation

- Objective: For a given test sample x, we want to consider only a reduced number $t < T$ of stages of $H(x)$ in order to assign a score to a *detection window*
- Find $\tilde{H}_{t \to T} \in \mathbb{R}$ that estimates H using only the first t stages of the soft cascade, such that:

$$H(x) \simeq \tilde{H}_{t \to T}(x) \quad \forall x \in \mathcal{P}(I)$$

Linear Cascade Approximation

- ex. Average positive traces extracted from a soft cascade of 1024 stages on the Oxford dataset. Traces are colored based on their level membership in the pyramid



Linear Cascade Approximation

Strategy

- Grouping all traces respect to their level
- Linear regression to estimate the parameters (slope and intercept) for the interpolation
- Compute the average trace for each group

Final score approximation takes the following form:

$$H_{t\to T}(x) = \bar{\mathbf{w}}_l \cdot \begin{bmatrix} 0 & T-t \end{bmatrix} + H_t(x) + \bar{\epsilon}_l$$

where

- l: level of x
- $\bar{\mathbf{w}}_l \equiv \mathbf{E}[\{\mathbf{w}_l^i\}]$ are the average trace parameters for the level l:
  - $\mathbf{w}_l^i = \arg\min_{\mathbf{w}} ||\mathbf{S}^\mathsf{T}\mathbf{w} - \mathbf{h}_{t\to T}(x^{(i)})||$
  - $\mathbf{w} \in \mathbb{R}^2$, $\mathbf{w} = \begin{bmatrix} w_0 & w_1 \end{bmatrix}$ with w_0 the intercept and w_1 the slope
  - $\mathbf{S} = \begin{bmatrix} 1 & \cdots & 1 & \cdots & 1 \\ t & t+\Delta & t+2\Delta & \cdots & T \end{bmatrix}$
  - $\mathbf{h}_{t\to T}^\mathsf{T}(x^{(i)}) = \begin{bmatrix} H_t(x^{(i)}) & H_{t+\Delta}(x^{(i)}) & \cdots & T \end{bmatrix}$
  - Δ: sampling step for the stages used in regression
- $\bar{\epsilon}_l$= average interpolation error on the stage T

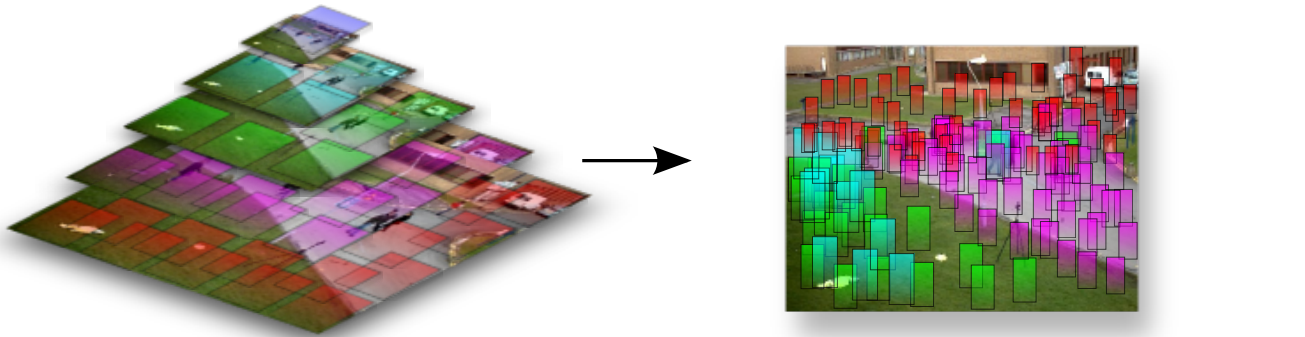
Generative Model for candidate window proposal

Observations:

- The presence and scale of targets is highly dependent on the geometry of the scene.
- Only detection windows in a limited scale range can be detected in a sub-region of frame
- The complete evaluation of all possible scales in all sub-regions of the image is wasteful

Idea

Only evaluate detection windows with a high likelihood to be a local maxima considering the geometric and scale statistics on the scene



Sliding windows Candidate Window Proposal

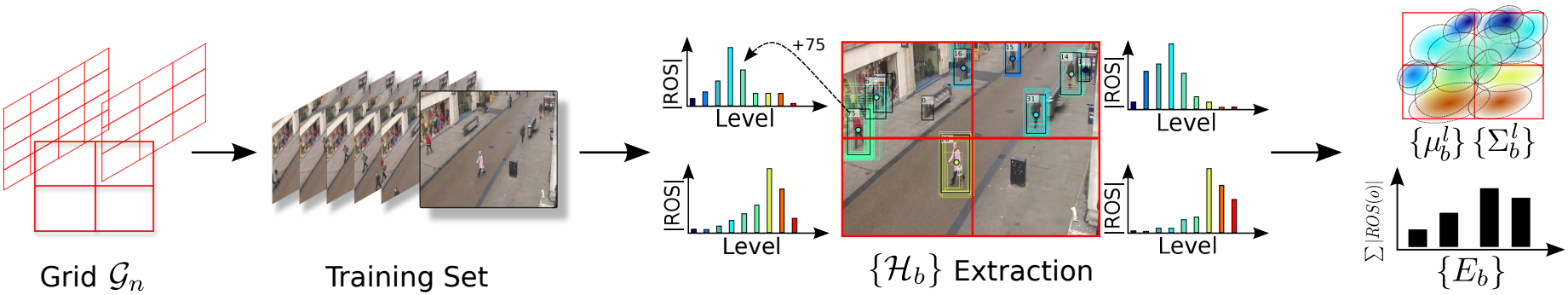
Generative Model for candidate window proposal

1. Leveraging Region of Support (ROS) information:
   - The ROS is indicative of both the detector precision and the scene geometry:
     - The cardinality of each ROS is a good estimate of true positive: objects with a low rank are often false positive.
     - The location and scale of strongs can be considered to learn a model able to describe the geometry and perspective of the scene
   - ROS information are discriminative and can be extracted at no additional cost during the non maximum suppression process.

ex. Some strongs (and their ROS) from a soft cascade classifier on a frame from Oxford:

Generative Model for candidate window proposal

2. Scene Model (M_n)



$$M_n = (\mathcal{G}_n, \{\tilde{\mathcal{H}}_b^l\}, \{\mu_b^l, \Sigma_b^l\}, \{E_b\})$$

where:

- n: grid of n^2 blocks
- $1 \leq l \leq L$ pyramid levels
- $\tilde{\mathcal{H}}_b^l$: $\mathcal{H}_b^l$ normalized over all levels l in block b
- $E_b = \frac{\sum_{l=1}^{L} \mathcal{H}_b^l}{\sum_{\tilde{b} \in \mathcal{G}_n} \sum_{l=1}^{L} \mathcal{H}_{\tilde{b}}^l}$

Observations:

- Training of Model *weakly-supervised*
- Search differentiated according to the sub-region of frame (block)
- Generation of detection windows based on: spatial position ($\{\mu_{b,l}\}, \{\Sigma_{b,l}\}$), scale ($\{\mathcal{H}_b\}$) and energy($\{E_b\}$)
- No need of calibration

Generative Model for candidate window proposal

3. Candidate windows proposal at detection time

Algorithm

For each block b and scale l of Images Pyramid $\mathcal{P}(I)$:

- Compute the total number of detection windows to genereate: $N = \gamma|\mathcal{P}(\mathcal{I})|E_b\mathcal{H}_b^l$
- If not enough information ($\mathcal{H}_b^l < \tau$) $\Longrightarrow$ uniform extraction in the block region
- Else randomly sample from normal distribution $\mathcal{N}(\mu_b^l, \Sigma_b^l)$ with covariance expansion:
  - Strategy round-based
  - For each round the covariance matrix is expanded by a factor (using $\mathcal{X}_\alpha^2$ distribution)
  - Iteration until the total number of obtained detection windows is approximately N
  - Reduction of duplicate samples

Parameter $\gamma \in [0, 1]$: Proportion of detection windows of a pyramid to be evaluated

- An estimate of the final speedup we want from the resulting detector
- Tradeoff between between speed ($\gamma \to 0$) and accuracy ($\gamma \to 1$) of the detector

Test and Results

Baseline:

- 3 *Soft cascade* with 1024 *stages*, Images Pyramid from 3 octaves with 8 levels each
- Features used by stages:
  - HOG with 6 bin for orientation $(0^\circ - 360^\circ)$
  - Gradient Histogram
  - Color Channels *LUV*

Dataset:

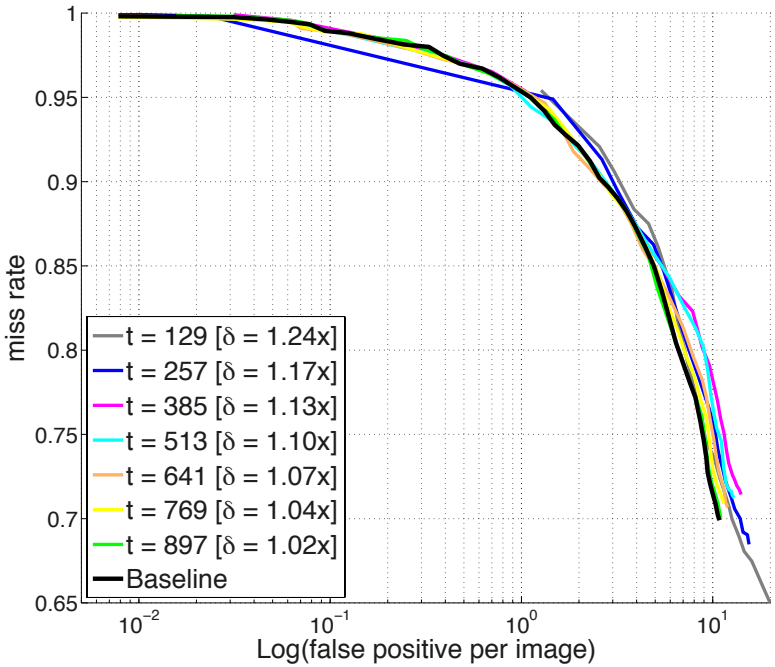
- seq. *Oxford*: sampling 1 fps from video Oxford (3 min) and frame reshape at 640×480
- seq. *PETS*: uniform extraction of 200 images from PETS (795 frames) and reshape at a 640×480

Speed of proposed Framework in terms of stages saving:

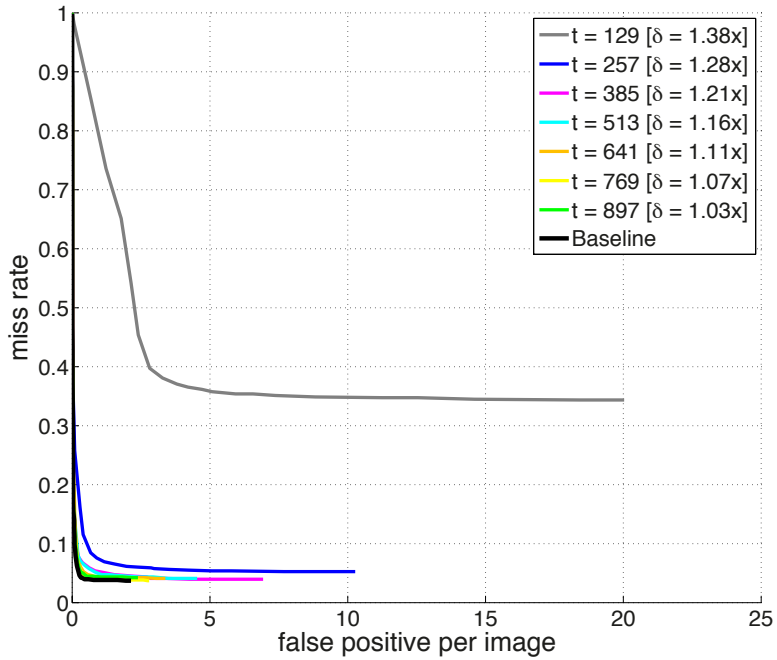
$$\delta = \frac{\sum_{\forall x \in \mathcal{P}} [H(x)]}{\sum_{\forall x \in \mathcal{X}} 1_{\{c=0\}} [H(x)] + 1_{\{c=1\}} [\tilde{H}_{t \to T}(x)]}$$

Performance with Linear Cascade Approximation

- *Dataset* considerati: seq. *Oxford* e *PETS*
- 7 values for t, uniform extracted from $[64, 961]$



seq. *Oxford*



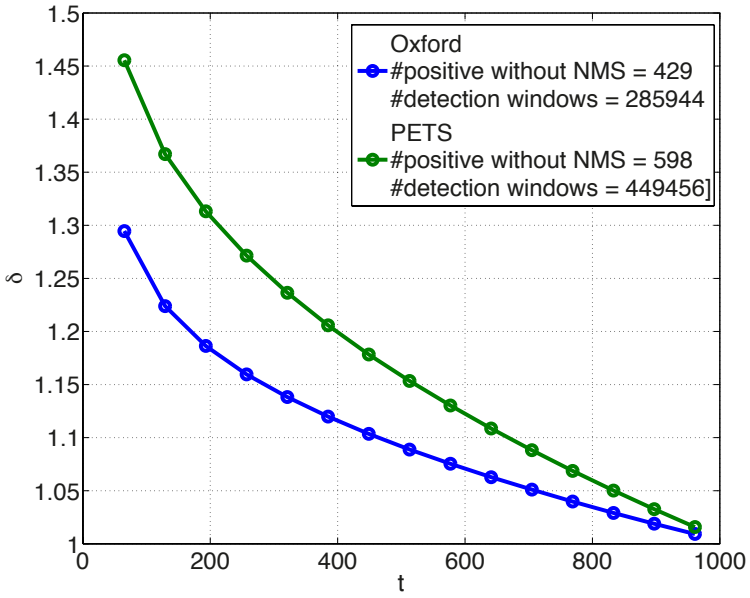
seq. *PETS*

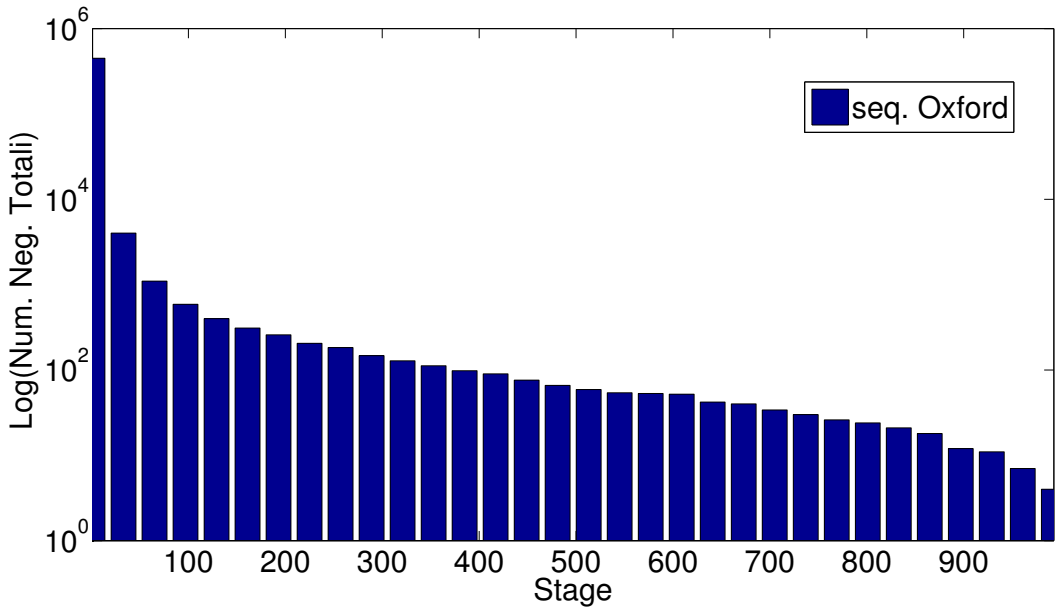
- Results:
  - seq. *Oxford*: reductions between $1\% - 24\%$ with maxima accuracy loss lower 5%
  - seq. *PETS*: same results of saving, but error not more than 4% with $t > 257$

Limited savings with Linear Cascade Approximation

Considering only stages reduction during detecteion windows evaluation is not enough to obtain high saving values:

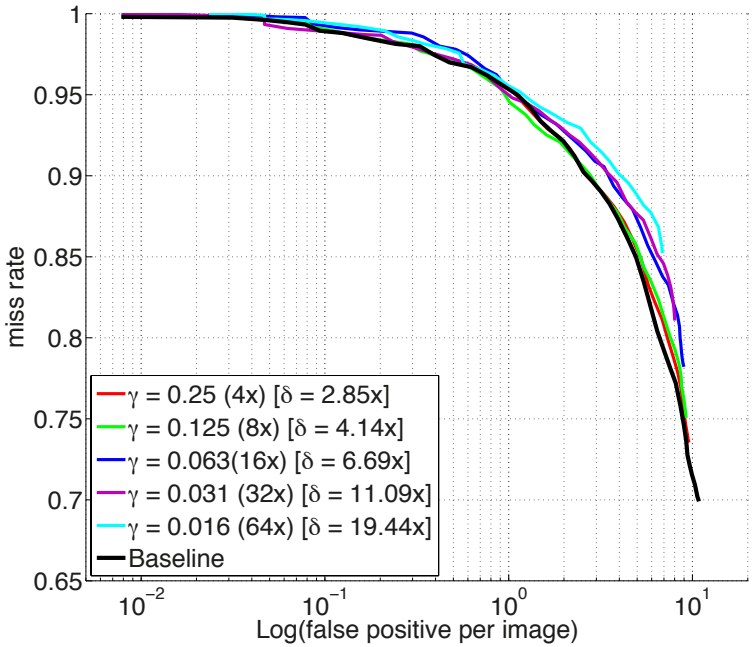
- $|\mathcal{X}_P| \ll |\mathcal{X}_N|$ (two order of magnitude)
- Same cost on evaluation of $\mathcal{X}_P$ and $\mathcal{X}_N$
- Maximum savings lower than 50% respect to full evaluation of Pyramid



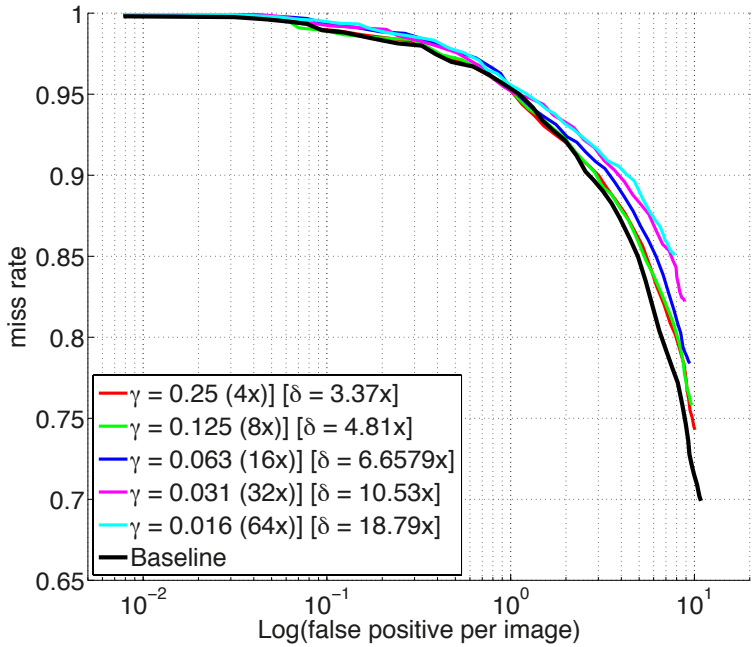


Performance with Generative Model

- *Dataset*: seq. *Oxford*
- $\mathcal{G}_n \in \{2, 3, 4, 5, 6\}$
- $SpeedUp \in [4\times, 8\times, 16\times, 32\times, 64\times]$



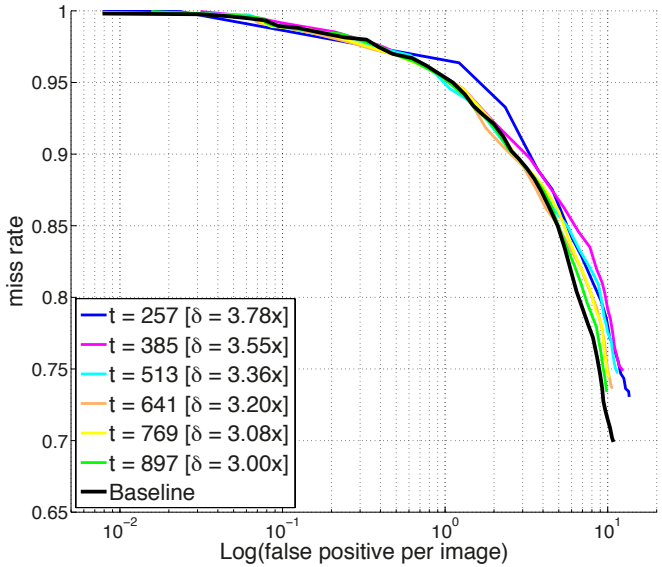
Grid 2×2



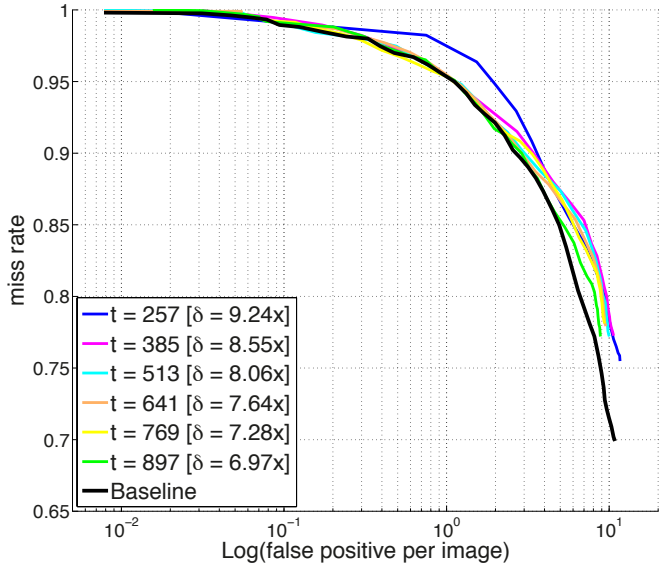
Grid 4×4

- Results:
  - With all configurations we obtain a savings greater then 50%
  - For grid size of 2×2, the minimum and maximum saving values is 65% $(2.85\times)$ and 95% $(19.44\times)$ respectively.

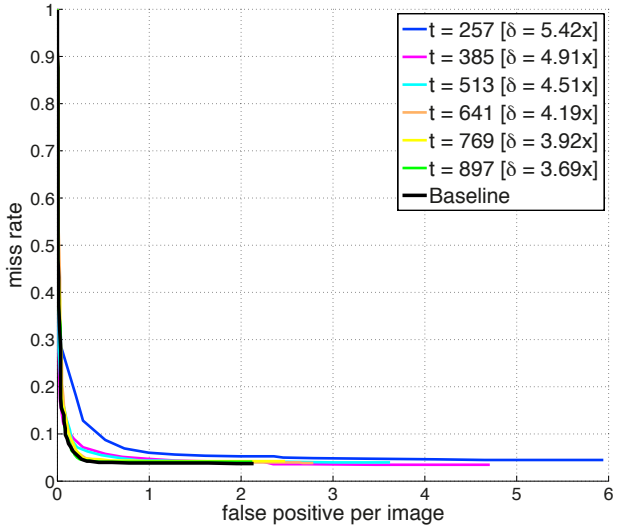
Performance with our Framework



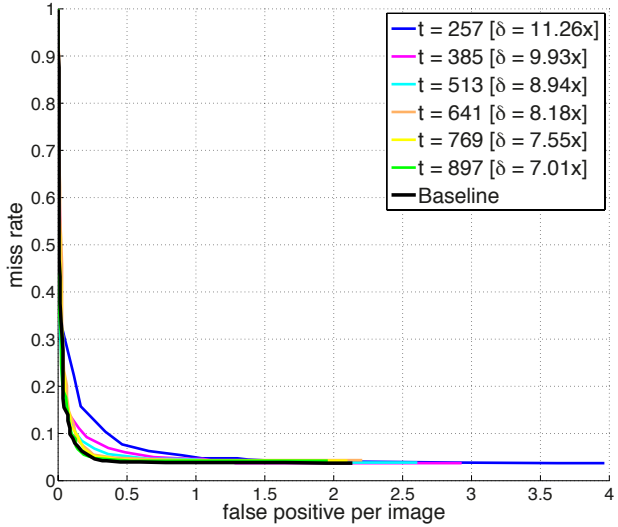
$\gamma = 0.25\ (4\times)$



$\gamma = 0.0625\ (16\times)$



$\gamma = 0.25\ (4\times)$



$\gamma = 0.0625\ (16\times)$

Performance Comparison: Our Framework *vs* Main Person Detectors

	Oxford		PETS	
Detectors	Miss-rate(%)	Savings(δ)	Miss-rate(%)	Savings(δ)
DPM	80	-	34	-
ACF	97	1×	51	1×
Baseline	99	1×	9	1×
Linear Cascade App.	**98.1**	1.17×	10.4	1.28×
Candidate Windows Pro.	98.9	11.09×	**7.4**	3.37×
With both	98.5	12.73×	11.4	4.19×

Conclusions

- In the classifier and Pyramid domains can be applied approximation strategies for complexity reduction
- Maximum saving is obtained means sparse sampling of detection windows to evaluate by classifier
- The ROS information proves to be effective data for modelling both geometry and statistics for a scene

Our Framework

- The proposed strategies are *weakly-supervised*
- The great reduction of stages to evaluate allows to run the detector in real-time
- The framework implementation results very easy, also no need of dedicated hardware to run (ex. GPU)