Supplementary material: Strengthening the Effectiveness of Pedestrian Detection with Spatially Pooled Features

Sakrapee Paisitkriangkrai, Chunhua Shen, Anton van den Hengel
The University of Adelaide, Australia

June 2014

In this supplementary document we provide a complete detail on the structural SVM problem presented in the main paper. We also provide in details the detection performance of different feature parameters and the analysis of selected features.

1 Optimizing the partial area under ROC curve

Notation Vectors are denoted by lower-case bold letters, *e.g.*, $\boldsymbol{x}$, matrices are denoted by upper-case bold letters, *e.g.*, $\mathbf{X}$ and sets are denoted by calligraphic upper-case letters, *e.g.*, $\mathcal{X}$. All vectors are assumed to be column vectors. The (i,j) entry of $\mathbf{X}$ is x_{ij}. Let $\{\boldsymbol{x}_i^+\}_{i=1}^m$ be a set of pedestrian training examples and $\{\boldsymbol{x}_j^-\}_{j=1}^n$ be a set of non-pedestrian training examples. The tuple of all training samples is written as $\mathbf{S} = (\mathbf{S}_+, \mathbf{S}_-)$ where $\mathbf{S}_+ = (\boldsymbol{x}_1^+, \cdots, \boldsymbol{x}_m^+) \in \mathcal{X}^m$ and $\mathbf{S}_- = (\boldsymbol{x}_1^-, \cdots, \boldsymbol{x}_n^-) \in \mathcal{X}^n$. In this paper, we are interested in the partial AUC (area under the ROC curve) within a specific false positive range $[\alpha, \beta]$. Given n negative training samples, we let $j_\alpha = \lceil n\alpha \rceil$ and $j_\beta = \lfloor n\beta \rfloor$. Let $\mathcal{Z}_\beta = \binom{\mathbf{S}_-}{j_\beta}$ denote the set of all subsets of negative training instances of size j_β. We define $\zeta = \{\boldsymbol{x}_{k_j}^-\}_{j=1}^{j_\beta} \in \mathcal{Z}_\beta$ as a given subset of negative instances, where $\boldsymbol{k} = [k_1, \ldots, k_{j_\beta}]$ is a vector indicating which elements of $\mathbf{S}_-$ are included. Our goal is to learn a scoring function $f(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x}$ such that the final detector achieves the optimal area under the ROC curve in the false positive rates (FPR) range $[\alpha, \beta]$.

Approach The pAUC risk for a scoring function $f(\cdot)$ between two pre-specified FPR $[\alpha, \beta]$ can be defined [5] as :

$$\hat{R}_\zeta(f) = \sum_{i=1}^{m} \sum_{j=j_\alpha+1}^{j_\beta} \mathbf{1}(f(\boldsymbol{x}_i^+) < f(\boldsymbol{x}_{(j)_{f|\zeta}}^-)). \tag{1}$$

Here $\boldsymbol{x}_i^+$ denotes the i-th positive training instance and $\boldsymbol{x}_{(j)_{f|\zeta}}^-$ denotes the j-th negative training instance sorted by f in the set $\zeta \in \mathcal{Z}_\beta$. Both $\boldsymbol{x}_i^+$ and $\boldsymbol{x}_{(j)_{f|\zeta}}^-$ represent the output vector of weak classifiers learned from AdaBoost. Clearly (1) is minimal when all positive samples, $\{\boldsymbol{x}_i^+\}_{i=1}^m$, are ranked above $\{\boldsymbol{x}_{(j)_{f|\zeta}}^-\}_{j=j_\alpha+1}^{j_\beta}$, which represent negative samples in our prescribed false positive range $[\alpha, \beta]$ (in this case, the log-average miss rate would be zero). The structural SVM framework can be adopted to optimize the pAUC risk by considering a classification problem of all $m \times j_\beta$ pairs of positive and negative samples. We define a new label matrix $\mathbf{Y} \in \mathcal{Y}_{m,j_\beta} = \{0,1\}^{m \times j_\beta}$ whose value for the pair (i,j) is defined as:

$$y_{ij} = \begin{cases} 0 & \text{if } f(\boldsymbol{x}_i^+) > f(\boldsymbol{x}_j^-) \\ 1 & \text{otherwise.} \end{cases} \tag{2}$$

The true pair-wise label is defined as $\mathbf{Y}^*$ where $y_{ij}^* = 0$ for all pairs (i,j). The pAUC loss of $\mathbf{Y}$ with respect to $\mathbf{Y}^*$ can then be written as:

$$\Delta(\mathbf{Y}, \mathbf{Y}^*) = \frac{1}{mn(\beta - \alpha)} \sum_{i=1}^{m} \sum_{j=j_\alpha+1}^{j_\beta} y_{i,(j)_Y}, \tag{3}$$

where $(j)_Y$ denotes the index of the negative instance in $\mathbf{S}_-$ ranked in the j-th position by any fixed ordering consistent with the matrix $\mathbf{Y}$. We define a joint feature map, $\phi_\zeta : (\mathcal{X}^m \times \mathcal{X}^n) \times \mathcal{Y}_{m,j_\beta} \rightarrow \mathbb{R}^t$, which takes a set of training instances (m positive samples and n negative samples) and an ordering matrix of dimension $m \times j_\beta$ and produce a vector output in $\mathbb{R}^t$ as:

$$\phi_\zeta(\mathbf{S}, \mathbf{Y}) = \sum_{i=1}^{m} \sum_{j=1}^{j_\beta} (1 - y_{ij})(\boldsymbol{x}_i - \boldsymbol{x}^-_{k_j}). \tag{4}$$

This feature map ensures that the variable $\boldsymbol{w}$ that optimizes $\boldsymbol{w}^\top \phi_\zeta(\mathbf{S}, \mathbf{Y})$ will also produce the optimal pAUC score for $\boldsymbol{w}^\top \boldsymbol{x}$. We can summarize the above problem as the following convex optimization problem [5]:

$$\min_{\boldsymbol{w}, \xi} \quad \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C\,\xi \quad \text{s.t. } \boldsymbol{w}^\top(\phi_\zeta(\mathbf{S}, \mathbf{Y}^*) - \phi_\zeta(\mathbf{S}, \mathbf{Y})) \geq \Delta(\mathbf{Y}, \mathbf{Y}^*) - \xi, \tag{5}$$

$\forall \zeta \in \mathcal{Z}_\beta$, $\forall \mathbf{Y} \in \mathcal{Y}_{m,j_\beta}$ and $\xi \geq 0$. Here $\mathbf{Y}^*$ denotes the correct relative ordering. $\mathbf{Y}$ denotes any arbitrary orderings and C controls the amount of regularization. A cutting plane method can be used to solve (5). The cutting plane algorithm begins with an empty initial constraint set and adds the most violated constraint set at each iteration. For our problems, finding the most violated constraint involves a combinatorial search over an exponential number of orderings of positive and negative training instances, $\mathcal{Y}_{m,j_\beta}$. The cutting plane algorithm continues until no constraint is violated by more than ϵ. Since the cutting plane method converges in a constant number of iterations [4] and (5) is solved once during training, most of the computation time in our experiments is spent in bootstrapping hard negative samples and weak learner training.

2 Experiments on spatially pooled covariance

In this section, we conduct additional experiments on sp-Cov with different subset of low-level features, multi-scale patches and spatial pooling parameters.

2.1 Feature representation

In the main paper we extract a covariance matrix from nine low-level visual features:

$$[x,\ y,\ |I_x|,\ |I_y|,\ |I_{xx}|,\ |I_{yy}|,\ M,\ O_1,\ O_2]$$

where x and y represent the pixel location, and I_x and I_{xx} are first and second intensity derivatives along the x-axis. The last three terms are the gradient magnitude ($M = \sqrt{I_x^2 + I_y^2}$), edge orientation as in [6] ($O_1 = \arctan(|I_x|/|I_y|)$) and an additional edge orientation O_2 in which,

$$O_2 = \begin{cases} \texttt{atan2}(I_y, I_x) & \text{if } \texttt{atan2}(I_y, I_x) > 0, \\ \texttt{atan2}(I_y, I_x) + \pi & \text{otherwise.} \end{cases}$$

In this supplementary, we measure the performance improvement of our detectors with sp-Cov features. The baseline detector is trained with LUV and 7 low-level visual features as described above (we exclude pixel location x and y). We then combine baseline features with covariance features extracted from a different combination of low-level features. We categorize low-level features, which we use to extract covariance features, into four categories: pixel locations $[x, y]$ (`LOC`), magnitude and orientations, $[M, O_1, O_2]$ (`ORI`), first-order intensity derivative of two-dimensional images, $[|I_x|, |I_y|]$ (`G1`) and second-order intensity derivative of two-dimensional images, $[|I_{xx}|, |I_{yy}|]$ (`G2`). We train these detectors[1] using the INRIA training set and evaluate these detectors on the INRIA, ETH and TUD-Brussels test sets. Log-average miss rates on three benchmark data sets are reported in table 1. Comparing `BASELINE` + `LOC` + `ORI`, `BASELINE` + `LOC` + `G1` and `BASELINE` + `LOC` + `G2`, we observe that covariance features extracted from `LOC` + `ORI` performs best and covariance features extracted from `LOC` + `G2` performs worst on all

[1] We set the shrinkage parameter to be 0.1 and the depth of decision trees to be 3.

three benchmark data sets. This indicates that first-order derivative features are more discriminative than second-order derivative features for the task of human detection. One interesting observation is that applying a non-linear transformation to the first-order derivative features can further improve the final detection performance, *i.e.*, `ORI` is derived from `G1` but it outperforms `G1`, *e.g.*, $M = \sqrt{I_x^2 + I_y^2}$ and $O_1 = \arctan(|I_x|/|I_y|)$. The best detection performance is observed when we extract sp-Cov from all low-level visual features.

Table 1: An improvement in log-average miss rate by combining `BASELINE` features with sp-Cov features extracted from different low-level visual features: `LOC` (pixel locations), `ORI` (magnitude and orientations), `G1` (first-order intensity derivative) and `G2` (second-order intensity derivative). `BASELINE` features consists of LUV, $|I_x|$, $|I_y|$, $|I_{xx}|$, $|I_{yy}|$, M, O_1 and O_2 (no variance and correlation information)

Features	INRIA	ETH	TUD-Brussels	Average
`BASELINE`	24.5%	49.5%	61.4%	45.1%
`BASELINE` + `LOC` + `ORI`	13.7%	43.6%	53.1%	36.8%
`BASELINE` + `LOC` + `G1`	14.9%	45.4%	55.7%	38.7%
`BASELINE` + `LOC` + `G2`	18.9%	47.7%	58.5%	41.7%
`BASELINE` + `LOC` + `ORI` + `G1`	13.8%	42.3%	51.0%	35.7%
`BASELINE` + `LOC` + `ORI` + `G2`	12.9%	42.8%	50.2%	35.3%
`BASELINE` + `LOC` + `G1` + `G2`	16.8%	47.5%	53.2%	39.2%
`ALL`	12.8%	42.0%	47.8%	**34.2**%

2.2 Multi-scale patches

In the main paper we independently extract sp-Cov features from multi-scale patches with size 8×8, 16×16 and 32×32 pixels. Each scale generates a different set of visual descriptors. In this experiment, we compare the performance of pedestrian detectors trained using sp-Cov features extracted from single-scale patches. We set the patch spacing stride (step-size) to be 1 pixel. The pooling region is set to be 4×4-pixels and the pooling spacing stride is set to 4 pixels. Log-average miss rates are reported in table 2. We observe that the patch size of 16×16 pixels performs best for single-scale patches. The best detection performance is observed when we extract sp-Cov from multi-scale patches.

Table 2: Log-average miss rate of sp-Cov using patches at multiple scales

Patch size (pixels)	INRIA	ETH	TUD-Brussels	Average
8×8	15.1%	44.8%	52.2%	37.4%
16×16	13.7%	43.3%	47.8%	34.9%
32×32	20.7%	45.5%	52.2%	39.5%
8×8 and 16×16	13.8%	42.5%	49.1%	35.1%
8×8 and 32×32	13.7%	43.6%	49.3%	35.5%
16×16 and 32×32	15.0%	43.9%	47.5%	35.5%
`ALL`	12.8%	42.0%	47.8%	**34.2**%

2.3 Spatial pooling parameters

Spatial pooling has been proven to be invariant to various image transformations and demonstrate better robustness to noise. Several empirical results have indicated that a pooling operation can greatly improve the recognition performance [1, 2, 3]. There exist two common pooling strategies in the literature: average pooling and max-pooling. Table 3 compares the detection results of both pooling operations. Similar to results reported in [3, 1], we observe that max-pooling slightly outperforms average pooling on average.

In the next experiment, we compare pedestrian detection results by varying the pooling region size from 4×4-pixels to 10×10-pixels and report the log-average miss rate in table 4. From the table, the pooling size of 4×4-pixels achieves the best detection result on average.

Table 3: Log-average miss rate of sp-Cov with average pooling and max-pooling

Patch size (pixels)	INRIA	ETH	TUD-Brussels	Average
average pooling	16.3%	41.7%	46.8%	34.9%
max-pooling	12.8%	42.0%	47.8%	**34.2**%

Table 4: Log-average miss rate of sp-Cov using different pooling sizes

Pooling size (pixels)	INRIA	ETH	TUD-Brussels	Average
4×4	12.8%	42.0%	47.8%	**34.2**%
6×6	11.9%	43.1%	50.6%	35.2%
8×8	13.5%	43.9%	49.9%	35.8%
10×10	12.1%	45.7%	51.6%	36.5%

2.4 Selected features and their spatial distribution

In this section, we illustrate the proportion of feature types selected by AdaBoost. We count the number of times each feature type is selected during the weak learner training and show its proportion in Fig. 1. From the figure, both sp-LBP and sp-Cov features are often selected by AdaBoost. Fig. 2 shows the spatial distribution of regions selected by different feature types. White pixels indicate that a large number of features are selected in that region. From the figure, most selected regions typically contain human contours (especially the head and shoulders). Colour features are selected around the human face (skin colour) while edge features are mainly selected around human contours (head, shoulders and feet). sp-LBP features are selected near human head and human hips while sp-Cov features are selected around human chest and regions between two human legs.

References

[1] Y. Boureau, N. L. Roux, F. Bach, J. Ponce, and Y. LeCun. Ask the locals: multi-way local pooling for image recognition. In *Proc. IEEE Int. Conf. Comp. Vis.*, 2011.

[2] K. Chatfield, V. Lempitsky, A. Vedaldi, and A. Zisserman. The devil is in the details: an evaluation of recent feature encoding methods. In *Proc. of British Mach. Vis. Conf.*, 2011.

[3] A. Coates and A. Ng. The importance of encoding versus training with sparse coding and vector quantization. In *Proc. Int. Conf. Mach. Learn.*, 2011.

[4] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Mach. Learn.*, 77(1):27–59, 2009.

[5] H. Narasimhan and S. Agarwal. $\text{SVM}^{\text{tight}}_{\text{pAUC}}$: a new support vector method for optimizing partial auc based on a tight convex upper bound. In *ACM Int. Conf. on Knowl. disc. and data mining*, 2013.

[6] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian manifolds. *IEEE Trans. Pattern Anal. Mach. Intell.*, 30(10):1713–1727, 2008.

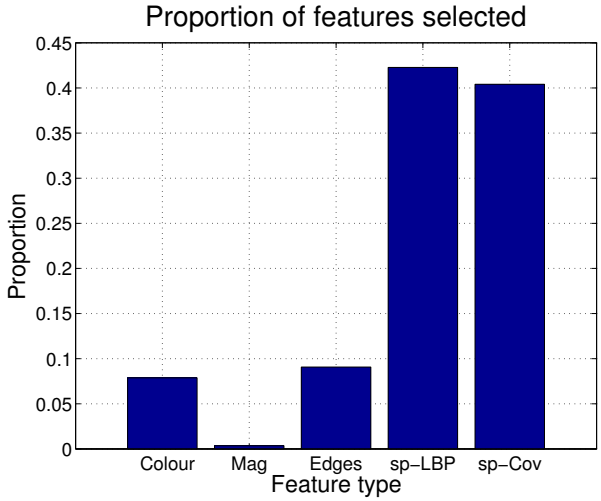
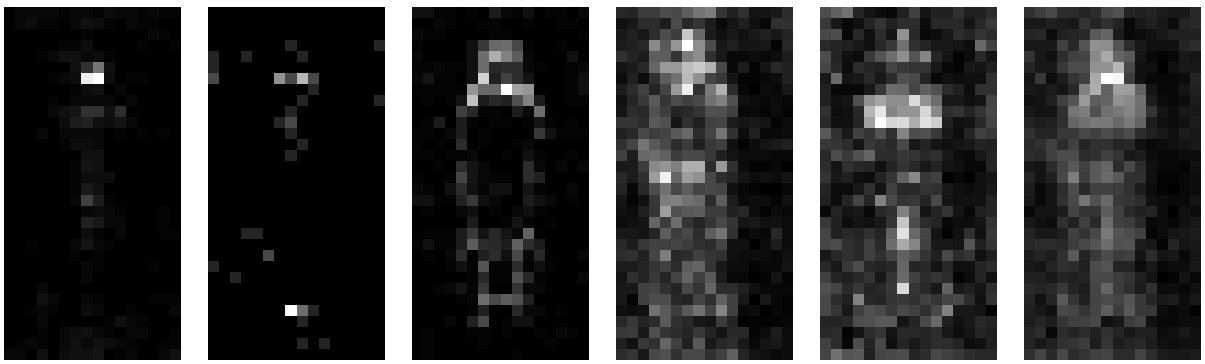


Figure 1: Proportion of features selected by AdaBoost. Colour represents LUV channels (3 channels). Mag reresents gradient magnitude (1 channel). Edges represents vertical lines, diagonal line and horizontal lines (6 channels). sp-LBP represents LBP features and spatially pooled LBP features (116 channels). sp-Cov represents spatially pooled covariance features (133 channels).



(a) Colour (b) Magnitude (c) Edges (d) sp-LBP (e) sp-Cov (f) ALL

Figure 2: Spatial distribution of selected features based on their feature types. White pixels indicate that a large number of features are selected in that area. Often selected regions correspond to human contour and human body.