Strengthening the Effectiveness of Pedestrian Detection with Spatially Pooled Features

Sakrapee Paisitkriangkrai, Chunhua Shen*, and Anton van den Hengel

The University of Adelaide, Australia
chunhua.shen@adelaide.edu.au

Abstract. We propose a simple yet effective approach to the problem of pedestrian detection which outperforms the current state-of-the-art. Our new features are built on the basis of low-level visual features and spatial pooling. Incorporating spatial pooling improves the translational invariance and thus the robustness of the detection process. We then directly optimise the partial area under the ROC curve (pAUC) measure, which concentrates detection performance in the range of most practical importance. The combination of these factors leads to a pedestrian detector which outperforms all competitors on all of the standard benchmark datasets. We advance state-of-the-art results by lowering the average miss rate from 13% to 11% on the INRIA benchmark, 41% to 37% on the ETH benchmark, 51% to 42% on the TUD-Brussels benchmark and 36% to 29% on the Caltech-USA benchmark.

1 Introduction

Pedestrian detection is a challenging but an important problem due to its practical use in many computer vision applications such as video surveillance, robotics and human computer interaction. The problem is made difficult by the inevitable variation in target appearance, lighting and pose, and by occlusion. In a recent literature survey on pedestrian detection [1] the authors evaluated several pedestrian detectors and concluded that combining multiple features can significantly boost the performance of pedestrian detection.

Hand-crafted low-level visual features have been applied to several computer vision applications and shown promising results [2, 3, 4, 5, 6, 7]. Inspired by the recent success of spatial pooling on object recognition and pedestrian detection problems [8, 9, 10, 11], we propose to perform the spatial pooling operation to create the new feature type. Our new detector yields competitive results to the state-of-the-art on major benchmark data sets. A further improvement is achieved when we combine the new feature type and channel features from [12]. We confirm the observation made in [1]: carefully combining multiple features often improves detection performance. The new multiple channel detector outperforms the state-of-the-art by a large margin. Despite its simplicity, our

* Corresponding author.

D. Fleet et al. (Eds.): ECCV 2014, Part IV, LNCS 8692, pp. 546–561, 2014.

new approach outperforms all reported pedestrian detectors, including several complex detectors such as LatSVM [13] (a part-based approach which models unknown parts as latent variables), ConvNet [9] (deep hierarchical models) and DBN-Mut [14] (discriminative deep model with mutual visibility relationship).

Dollár *et al.* propose to compare different detectors using the miss rate performance at 1 false positive per image (FPPI) as a reference point [15]. This performance metric was later revised to the *log-average miss rate* in the range 0.01 to 1 FPPI as this better summarizes practical detection performance [1]. This performance metric is also similar to the average precision reported in text retrieval and PASCAL VOC challenge. As the performance is assessed over the partial range of false positives, the performance of the classifier outside this range is ignored as it is not of practical interest. Many proposed pedestrian detectors optimize the miss rate over the complete range of false positive rates, however, and can thus produce suboptimal results both in practice, and in terms of the log-average miss rate. In this paper, we address this problem by optimizing the log-average miss rate performance measure directly, and in a more principled manner. This is significant because it ensures that the detector achieves its best performance within the range of practical significance, rather than over the whole range of false positive rates, much of which would be of no practical value. The approach proposed ensures that the performance is optimized not under the full ROC curve but only within the range of practical interest, thus concentrating performance where it counts, and achieving significantly better results in practice.

Main Contributions. (1) We propose a novel approach to extract low-level visual features based on spatial pooling for the problem of pedestrian detection. Spatial pooling has been successfully applied in sparse coding for generic image classification problems. The new feature is simple yet outperforms the original covariance descriptor of [5] and LBP descriptor of [7]. (2) We discuss several factors that affect the performance of boosted decision tree classifiers for pedestrian detection. Our new design leads to a further improvement in log-average miss rate. (3) Empirical results show that the new approach, which combines our proposed features with existing features [12, 7] and optimizes the log-average miss rate measure, outperforms all previously reported pedestrian detection results and achieves state-of-the-art performance on INRIA, ETH, TUD-Brussels and Caltech-USA pedestrian detection benchmarks.

Related Work. Numerous pedestrian detectors have been proposed over the past decade along with newly created pedestrian detection benchmarks such as INRIA, ETH, TUD-Brussels, Caltech and Daimler Pedestrian data sets. We refer the reader to [1] for an excellent review on pedestrian detection frameworks and benchmark data sets. In this section, we briefly discuss several recent state-of-the-art pedestrian detectors that are not covered in [1].

Sermanet *et al.* train a pedestrian detector using a convolutional network model [9]. Instead of using hand designed features, they propose to use unsupervised sparse auto encoders to automatically learn features in a hierarchy. Experimental results show that their detector achieves competitive results on major benchmark data sets. Benenson *et al.* investigate different low-level aspects of pedestrian detection [16]. The authors show that by properly tuning low-level features, such as feature selection, pre-processing the raw image and classifier training, it is possible to reach state-of-the-art results on major benchmarks. From their paper, one key observation that significantly improves the detection performance is to apply image normalization to the test image before extracting features.

Lim *et al.* propose novel mid-level features, known as sketch tokens [17]. The feature is obtained from hand drawn sketches in natural images and captures local edge structure such as straight lines, corners, curves, parallel lines, *etc.* They combine their proposed features with channel features of [18] and train a boosted detector. By capturing both simple and complex edge structures, their detector achieves the state-of-the-art result on the INRIA test set. Park *et al.* propose new motion features for detecting pedestrians in a video sequence [8]. By factoring out camera motion and combining their proposed motion features with channel features [18], the new detector achieves a five-fold reduction in false positives over previous best results on the Caltech pedestrian benchmark.

2 Our Approach

Despite several important work on object detection, the most practical and successful pedestrian detector is still the sliding-window based method of Viola and Jones [6]. Their method consists of two main components: feature extraction and the AdaBoost classifier. For pedestrian detection, the most commonly used features are HOG [2] and HOG+LBP [7]. Dollár *et al.* propose Aggregated Channel Features (`ACF`) which combine gradient histogram (a variant of HOG), gradients and LUV [12]. ACF uses the same channel features as `ChnFtrs` [18], which is shown to outperform HOG [16, 18].

To train the classifier, the procedure known as bootstrapping is often applied, which harvests hard negative examples and re-trains the classifier. Bootstrapping can be repeated several times. It is shown in [19] that at least two bootstrapping iterations are required for the classifier to achieve good performance. In this paper, we build our detection framework based on [12]. We first propose the new feature type based on a modified low-level descriptor and spatial pooling. We then discuss how the miss rate performance measure can be further improved using structural SVM. Finally, we discuss our improvements to [12] in order to achieve state-of-the-art detection results on most benchmark data sets.

2.1 Spatially Pooled Features

Spatial pooling has been proven to be invariant to various image transformations and demonstrate better robustness to noise [20, 21, 22]. Several empirical results have indicated that a pooling operation can greatly improve the recognition performance. Pooling combines several visual descriptors obtained at nearby locations into some statistics that better summarize the features over some region of interest (pooling region). The new feature representation preserves visual information over a local neighbourhood while discarding irrelevant details and noises. Combining max-pooling with unsupervised feature learning methods have led to state-of-the-art image recognition performance on object recognition. Although these feature learning methods have shown promising results over hand-crafted features, computing these features from learned dictionaries is still a time-consuming process for many real-time applications. In this section, we further improve the performance of low-level features by adopting the pooling operator commonly applied in unsupervised feature learning and supervised convolutional neural networks. This simple operation can enhance the feature robustness to noise and image transformation. In the following section, we investigate two visual descriptors which have shown to complement HOG in pedestrian detection, namely covariance descriptors and LBP. It is important to point out here that our approach is not limited to these two features, but can be applied to any low-level visual features.

Covariance Matrix. A covariance matrix is positive semi-definite. It provides a measure of the relationship between two or more sets of variates. The diagonal entries of covariance matrices represent the variance of each feature and the non-diagonal entries represent the correlation between features. The variance measures the deviation of low-level features from the mean and provides information related to the distribution of low-level features. The correlation provides the relationship between multiple low-level features within the region. In this paper, we follow the feature representation as proposed in [5]. However, we introduce an additional edge orientation which considers the sign of intensity derivatives. Low-level features used in this paper are:

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

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 [5] ($O_1 = \arctan(|I_x|/|I_y|)$) and an additional edge orientation O_2 in which,

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

The orientation O_2 is mapped over the interval $[0, \pi]$. Although some O_1 features might be redundant after introducing O_2, these features would not deteriorate the performance as they are unlikely to be selected by the boosting learner. Our preliminary experiments show that using O_1 alone yields slightly worse performance than combining O_1 and O_2. With the defined mapping, the input image is mapped to a 9-dimensional feature image. The covariance descriptor of a region is a 9×9 matrix, and due to symmetry, only the upper triangular part is stored, which has only 45 different values.

LBP. Local Binary Pattern (LBP) is a texture descriptor that represents the binary code of each image patch into a feature histogram [23]. The standard version of LBP is formed by thresholding the 3×3-neighbourhood of each pixel with the centre pixel's value. All binary results are combined to form an 8-bit binary value (2^8 different labels). The histogram of these 256 different labels can be used as texture descriptor. The LBP descriptor has shown to achieve good performance in many texture classification [23]. In this work, we transform the input image from the RGB color space to LUV space and apply LBP to the luminance (L) channel. We adopt an extension of LBP, known as the uniform LBP, which can better filter out noises [7]. The uniform LBP is defined as the binary pattern that contains at most two bitwise transitions from 0 to 1 or vice versa.

Spatially Pooled Covariance. In this section, we improve the spatial invariance and robustness of the original covariance descriptor by applying the operator known as spatial pooling. There exist two common pooling strategies in the literature: average pooling and max-pooling. We use max-pooling as it has been shown to outperform average pooling in image classification [22, 20]. We divide the image window into *dense patches* (refer to Fig. 1). For each patch, covariance features are calculated over pixels within the patch. For better invariance to translation and deformation, we perform spatial pooling over a fixed-size spatial region (*pooling region*) and use the obtained results to represent covariance features in the pooling region. The pooling operator thus summarizes multiple covariance matrices within each pooling region into a single matrix which represents covariance information. We refer to the feature extracted from each pooling region as spatially pooled covariance (sp-Cov) feature. Note that extracting covariance features in each patch can be computed efficiently using the integral image trick [24]. Our sp-Cov differs from covariance features in [5] in the following aspects:

1. We apply spatial pooling to a set of covariance descriptors in the pooling region. To achieve this, we ignore the geometry of covariance matrix and stack the upper triangular part of the covariance matrix into a vector such that pooling is carried out on the vector space. For simplicity, we carry out pooling over a square image region of fixed resolution. Considering pooling over a set of arbitrary rectangular regions as in [25] is likely to further improve the performance of our features.

2. Instead of normalizing the covariance descriptor of each patch based on the whole detection window [5], we calculate the correlation coefficient within each patch. The correlation coefficient returns the value in the range $[-1, 1]$. As each patch is now independent, the feature extraction can be done in parallel on the GPU.

Implementation. We extract sp-Cov using multi-scale patches with the following sizes: 8×8, 16×16 and 32×32 pixels. Each scale will generate a different set of visual descriptors. Multi-scale patches have also been used in [26]. In this paper, the use of multi-scale patches is important as it expands the richness of our feature representations and enables us to capture human body parts at different scales. In our experiments, 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 in our experiments.

Spatially Pooled LBP. Similar to sp-Cov, we divide the image window into small patches and extract LBP over pixels within the patch. The histogram,

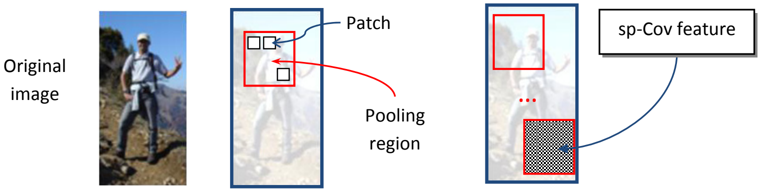

Fig. 1. Architecture of our pooled features. In this example, sp-Cov are extracted from each fixed sized pooling region.

which represents the frequency of each pattern occurring, is computed over the patch. For better invariance to translation, we perform spatial pooling over a pooling region and use the obtained results to represent the LBP histogram in the pooling region. We refer to the new feature as spatially pooled LBP (sp-LBP) feature.

Implementation. We apply the LBP operator on the 3 × 3-neighbourhood at each pixel. The LBP histogram is extracted from a patch size of 4 × 4 pixels. We extract the 58-dimension LBP histogram using a C-MEX implementation of [27]. For sp-LBP, the patch spacing stride, the pooling region and the pooling spacing stride are set to 1 pixel, 8 × 8 pixels and 4 pixels, respectively. We also experiment with combining the LPB histogram extracted from multi-scale patches but only observe a slight improvement in detection performance at a much higher feature extraction time. Instead of extracting LBP histograms from multi-scale patches, we combine sp-LBP and LBP as channel features.

Discussion. Although we make use of spatial pooling, our approach differs significantly from the unsupervised feature learning pipeline, which has been successfully applied to image classification problem [26, 11]. Instead of pooling encoded features over a pre-trained dictionary, we compute sp-Cov and sp-LBP by performing pooling directly on covariance and LBP features extracted from local patches. In other words, our proposed approach removes the dictionary learning and feature encoding from the conventional unsupervised feature learning [26, 11]. The advantage of our approach over conventional feature learning is that our features have much less dimensions than the size of visual words often used in generic image classification [11]. Using too few visual words can significantly degrade the recognition performance as reported in [21] and using too many visual words would lead to very high-dimensional features and thus make the classifier training become computationally infeasible.

2.2 Optimizing the Partial Area under ROC Curve

As the performance of the detector is usually measured using the log-average miss rate, we optimize the pAUC (the partial AUC) between any two given false positive rates $[\alpha, \beta]$, similar to the work of [28]. Unlike [28], in which weak learners are selected based on the pAUC criterion, we use AdaBoost to select weak learners as it is more efficient. In order to achieve the best performance, we build a feature vector from the weak learners' output and learn the pAUC classifier in the final stage. For each predicted positive patch, the confidence score is re-calibrated based on this pAUC classifier. This post-learning step is similar to the work of [29], in which the authors learn the asymmetric classifier from the output of AdaBoost's weak learners to handle the node learning goal in the cascade framework.

The pAUC risk for a scoring function $f(\cdot)$ between two pre-specified FPR $[\alpha, \beta]$ can be defined [30] as :

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

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. In our experiments, the pAUC classifier is trained once at the final bootstrapping iteration and most of the computation time is spent in extracting features and bootstrapping hard negative samples. See the supplementary material for more details on the structural SVM problem.

2.3 Design Space

In this section, we further investigate the experimental design of the ACF detector [12]. For experiments on shrinkage and spatial pooling, we use the proposed sp-Cov as channel features. For experiments on the depth of decision trees, we use channel features of [12]. All experiments are carried out using AdaBoost with the shrinkage parameter of 0.1 as a strong classifier and level-3 decision trees as weak classifiers (if not specified otherwise). We use three bootstrapping stages and the final model consists of 2048 weak classifiers with soft cascade. We heuristically set the soft cascade's reject threshold to be -10 at every node. We trained all detectors using the INRIA training set and evaluated the detector on INRIA, ETH and TUD-Brussels benchmark data sets.

Shrinkage. Hastie *et al.* show that the accuracy of boosting can be further improved by applying a weighting coefficient known as shrinkage [31]. The explanation given in [32] is that a shrinkage version of boosting simply converges to the ℓ_1 regularized solution. It can also be viewed as another form of regularization for boosting. At each iteration, the weak learner's coefficient is updated by

$$F_t(\boldsymbol{x}) = F_{t-1}(\boldsymbol{x}) + \nu \cdot \omega_t h_t(\boldsymbol{x}) \tag{2}$$

Here $h_t(\boldsymbol{x})$ is AdaBoost's weak learner at the t-th iteration and ω_t is the weak learner's coefficient at the t-th iteration. $\nu \in (0, 1]$ can be viewed as a learning

Table 1. Log-average miss rate when varying shrinkage parameters. Shrinkage can further improve the final detection performance. † The model consists of 4096 weak classifiers while other models consist of 2048 weak classifiers.

Shrinkage	INRIA	ETH	TUD-Br.	Avg.
None	14.4%	40.8%	48.7%	34.6%
$\nu = 0.5$	12.5%	43.7%	50.3%	35.5%
$\nu = 0.2$	11.6%	41.4%	50.4%	34.4%
$\nu = 0.1$	12.8%	42.0%	47.8%	**34.2%**
$\nu = 0.05$	14.0%	43.1%	51.4%	36.2%
$\nu = 0.05^{\dagger}$	12.8%	42.6%	48.6%	34.7%

Table 2. Log-average miss rate of our features with and without applying spatial pooling

	Covariance			LBP		
	INRIA	ETH	TUD-Brussels	INRIA	ETH	TUD-Brussels
without pooling	14.2%	42.7%	48.6%	25.8%	47.8%	**55.5%**
with pooling	**12.8%**	**42.0%**	**47.8%**	**23.7%**	**46.2%**	55.8%

rate parameter. The smaller the value of ν, the higher is the overall accuracy as long as the number of iterations is large enough. The authors of [32] report that shrinkage often produces better generalization performance compared to linear search algorithms.

We compare four different shrinkage parameters from {0.05, 0.1, 0.2, 0.5} with the conventional AdaBoost. When applying shrinkage, we lower the soft cascade's reject threshold by a factor of ν as weak learners' coefficients have been diminished by a factor of ν. The log-average miss rate of different detectors is shown in Table 1. We observe that applying a small amount of shrinkage ($\nu \leq 0.2$) often improves the detection performance. From Table 1, setting the shrinkage value to be too small ($\nu = 0.05$) without increasing the number of weak classifiers can hurt the performance as the number of boosting iterations is not large enough for the boosting to converge. For the rest of our experiments, we set the shrinkage parameter to be 0.1 as it gives a better trade-off between the performance and the number of weak classifiers.

Spatial Pooling. In this section, we compare the performance of the proposed feature with and without spatial pooling. For sp-Cov and sp-LBP without pooling, we extract both low-level visual features with the patch spacing stride of 4 pixels and no pooling is performed. Using these low-level features and LUV colour features, we trained four detectors using the INRIA training set. Log-average miss rates of both features are shown in Table 2. We observe that it is beneficial to apply spatial pooling as it increases the robustness of the features against small deformations and translations. We observe a reduction in miss rate by more than one percent on the INRIA test set. Since we did not combine sp-LBP with HOG as in [7], sp-LBP performs slightly worse than sp-Cov.

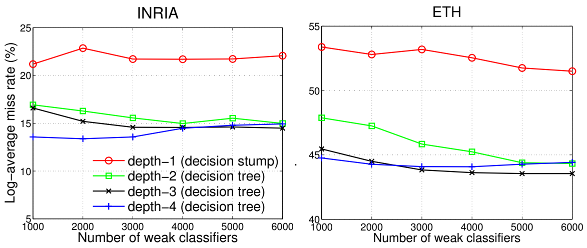

Fig. 2. Log-average miss rates of different tree depths on INRIA (left) and ETH (right) benchmark data sets

Depth of Decision Trees. The authors of [18] and [16] report that the depth-2 decision tree produces the best performance in their experiments. However, we observe that the depth-3 decision tree offers better generalization performance. To conduct our experiments, we trained 4 different pedestrian detectors with decision trees of depth 1 (decision stump) to 4 (containing 15 stumps). Our experiments are based on the `ACF` detector of [12] which combines gradient histogram (O), gradient (M) and LUV features. The `ACF` detector linearly quantizes feature values into 256 bins to speed up the conventional decision tree training [33]. We trained the pedestrian detector using the INRIA training set and evaluated the detector on both INRIA and ETH benchmark data sets. Fig. 2 plots the log-average miss rate on the vertical axis and the number of weak classifiers on the horizontal axis. We observe that the pedestrian detection performance improves as we increase the depth of decision trees. Similar to [16], we observe that using decision stumps as weak learners can lead to significant underfitting, *i.e.*, the weak learner can not separate pedestrian patches from non-pedestrian patches. On the other hands, setting the tree depth to be larger than two can lead to a performance improvement, especially on the ETH data set. For the rest of our experiments, we set the depth of decision trees to be three as it achieves good generalization performance and is faster to train than the depth-4 decision tree.

3 Experiments

We train two detectors: one using the INRIA training set and one using the Caltech-USA training set. For INRIA, each pedestrian training sample is scaled to a resolution of 64 × 128 pixels. Negative patches are collected from INRIA background images. We follow the work of [12] to train the boosted pedestrian detector. Each detector is trained using three bootstrapping stages and consists of 2048 weak classifiers. The detector trained on the INRIA training set is evaluated on all benchmark data sets except the Caltech-USA test set[1]. On both

[1] Park et al. [8] report a performance improvement on the Caltech-USA when they retrain the detector using the Caltech-USA training set. We follow the setup discussed in [8].

ETH and TUD-Brussels data sets, we apply the automatic colour equalization algorithms (ACE) [34] before we extract channel features [16]. We upscale both ETH and TUD-Brussels test images to 1280 × 960 pixels. For Caltech-USA, the resolution of the pedestrian model is set to 32 × 64 pixels. We exclude occluded pedestrians from the Caltech training set [8]. Negative patches are collected from the Caltech-USA training set with pedestrians cropped out. To obtain final detection results, greedy non-maxima suppression is applied with the default parameter as in Addendum of [18]. We use the log-average miss rate to summarize the detection performance. For the rest of our experiments, we evaluate our pedestrian detectors on the reasonable subset (pedestrians are at least 50 pixels in height and at least 65% visible).

3.1 Improved Covariance Descriptor

In this experiment, we evaluate the performance of the proposed sp-Cov. sp-Cov consists of 9 low-level image statistics. We exclude the mean and variance of two image statistics (pixel locations at x and y co-ordinates) since they do not capture discriminative information. We also exclude the correlation coefficient between pixel locations at x and y co-ordinates. Hence there is a total of 136 channels (7 low-level image statistics + $3 \cdot 7$ variances + $3 \cdot 35$ correlation coefficients + 3 LUV color channels)[2]. It is important to note here that our features and weak classifiers are different from those in [5]. There the authors calculate the covariance distance in the Riemannian manifold. As eigen-decomposition is performed, the approach of [5] is computationally expensive. We speed up the weak learner training by proposing our modified covariance features and train the weak learner using the decision tree. The new weak learner is not only simpler than [5] but also highly effective.

We compare our detector with the original covariance descriptor [5] in Fig. 3. We plot HOG [2] and HOG+LBP [7] as the baseline. Similar to the result reported in [16], where the authors show that `HOG+BOOSTING` reduces the average miss-rate over `HOG+SVM` by more than 30%, we observe that applying our sp-Cov features as the channel features significantly improves the detection performance over the original covariance detector (a reduction of more than 5% miss rate at 10^{-4} false positives per window). More experiments on sp-Cov with different subset of low-level features, multi-scale patches and spatial pooling parameters can be found in the supplementary.

Next we compare the proposed sp-Cov with `ACF` features (M+O+LUV) [12]. Since `ACF` uses fewer channels than sp-Cov, for a fair comparison, we increase `ACF`'s discriminative power by combining `ACF` features with LBP[3] (M+O+LUV+LBP). The results are reported in Table 3. We observe that sp-Cov yields competitive results to M+O+LUV+LBP. From the table, sp-Cov performs better on the INRIA test set, worse on the ETH test set and on par with M+O+LUV+LBP on

[2] Note here that we extract covariance features at 3 different scales.

[3] In our implementation, we use an extension of LBP, known as the uniform LBP, which can better filter out noises [7]. Each LBP bin corresponds to each channel.

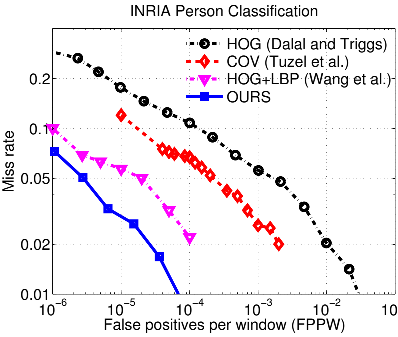

Fig. 3. ROC curves of our sp-Cov features and the conventional covariance detector [5] on the INRIA test image

Table 3. Log-average miss rates of various feature combinations

	# channels	INRIA	ETH	TUD-Br.
M+O+LUV+LBP	68	14.5%	39.9%	47.0%
sp-Cov+LUV	136	12.8%	42.0%	47.8%
sp-Cov+M+O+LUV	143	**11.2%**	39.4%	46.7%
sp-Cov+sp-LBP+M+O+LUV	259	**11.2%**	**38.0%**	**42.5%**

the TUD-Brussels test set. We observe that the best performance is achieved by combining sp-Cov and sp-LBP with M+O+LUV.

3.2 Improving Average Miss Rate with pAUC$^{\text{struct}}$

In this experiment, we evaluate the effect of re-calibrating the final confidence score with pAUC$^{\text{struct}}$. Instead of using the weighted responses from AdaBoost, we re-rank the confidence score of predicted pedestrian patches using a scoring function of pAUC$^{\text{struct}}$. The performance is calculated by varying the threshold value in the false positive range of $[0.01, 1]$ FPPI. Since the partial area under the ROC curve is determine on a logarithmic scale [1], it is non-trivial to determine the best pAUC$^{\text{struct}}$ parameters α and β which maximize the detection rate between 0.01 and 1 false positive per image. In our experiment, we heuristically set α to be 0 and perform a cross-validation to find the best pAUC$^{\text{struct}}$ regularization parameter C (see Supplementary) and the false positive rate β. In this section, we first train the baseline pedestrian detector as discussed in Section 2.3. The baseline detector achieves the log-average miss rate of 21.3%. Next we perform the post-learning step by re-ranking the confidence score of positive and negative samples based on the pAUC criterion. Using cross-validation on the INRIA training set, the post-learning step improves the log-average miss rate by 0.6%. Fig. 4 plots the log average miss rate with respect to the pAUC$^{\text{struct}}$ regularization parameter C and the false positive rate β. From the figure, the following parameters ($C = 2^4$ and $\beta = 0.7$) perform best with a miss rate of 20.7%.

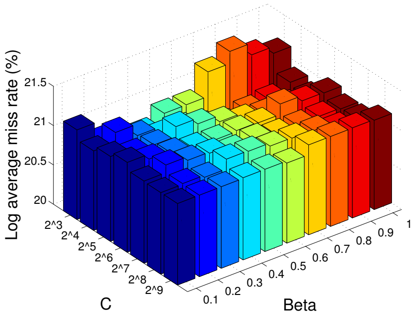

Fig. 4. Cross-validation results (log-average miss rate) as the pAUC$^{\text{struct}}$ regularization parameter C and the false positive rate β change. Without pAUC$^{\text{struct}}$, the detector achieves the miss rate of 21.3%. The detector with post-tuning ($C = 2^4$ and $\beta = 0.7$) performs best with a miss rate of 20.7% (an improvement of 0.6%).

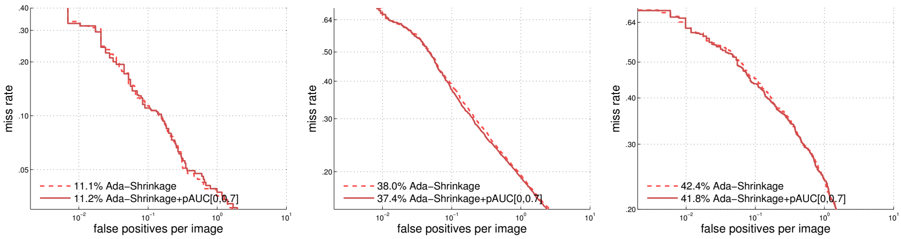

Fig. 5. Detection performance of our detectors with pAUC$^{\text{struct}}$ post-tuning on INRIA (*left*), ETH (*middle*) and TUD-Brussels (*right*) benchmark data sets

In the next experiment, we evaluate the detector with the post-learning step on INRIA, ETH and TUD-Brussels benchmark data sets. ROC curves along with their log-average miss rates between $[0.01, 1]$ FPPI are shown in Fig. 5. Based on our results, applying pAUC$^{\text{struct}}$ improves the log-average miss rate of the original detector on both ETH and TUD-Brussels benchmark data sets by 0.6%. However we do not observe an improvement on the INRIA test set. Our conjecture is that the INRIA test set consists of high-resolution human in a standing position which might be easier to detect than those appeared in ETH and TUD-Brussels data sets. No improvement in detection performance is observed on the INRIA test set as compared to the detection results on ETH and TUD-Brussels data sets.

3.3 Comparison with State-of-the-Art Results

In the next experiment, we compare our combined features with state-of-the-art detectors. Recently Lim et al. [17] propose sketch tokens (ST) feature which achieves the state-of-the-art result on the INRIA test set (a miss rate of 13.3%). Our new detector outperforms ST by achieving a miss rate of 11.2%. Our best performance is achieved when we apply pAUC$^{\text{struct}}$ to the combined features (a

Table 4. Log-average miss rates of various algorithms on INRIA, ETH, TUD-Brussels and Caltech-USA test sets. The best detector is shown in boldface. We train two detectors: one using INRIA training set (evaluated on INRIA, ETH and TUD-Brussels test sets) and another one using Caltech-USA training set (evaluated on Caltech-USA test set). The log-average miss rate of our detection results are calculated using the Caltech pedestrian detection benchmark version 3.2.0. † Results reported here are taken from `http://www.vision.caltech.edu/Image_Datasets/CaltechPedestrians/` and are slightly different from the one reported in the original paper.

Approach	INRIA	ETH	TUD-Brussels	Caltech-USA
Sketch tokens [17] (Prev. best on INRIA†)	13.3%	N/A	N/A	N/A
DBN-Mut [14] (Prev. best on ETH†)	N/A	41.1%	N/A	48.2%
MultiFtr+Motion+2Ped [35] (Prev. best on TUD-Brussels)	N/A	N/A	50.5%	N/A
SDtSVM [8] (Prev. best on Caltech-USA)	N/A	N/A	N/A	36.0%
Roerei [16] (2-nd best on INRIA† & ETH†)	13.5%	43.5%	64.0%	48.4%
Ours (sp-Cov+sp-LBP+M+O+LUV)	**11.1%**	38.0%	42.4%	29.4%
Ours (sp-Cov+sp-LBP+M+O+LUV + pAUC$^{\text{struct}}$)	11.2%	**37.4%**	**41.8%**	**29.2%**

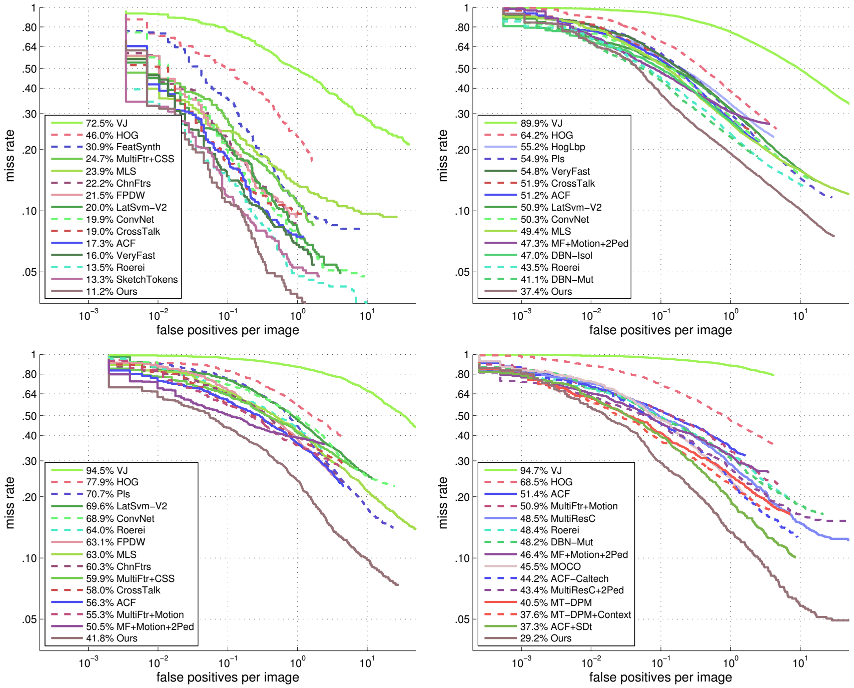

Fig. 6. ROC curves of our proposed approach on INRIA, ETH, TUD-Brussels and Caltech-USA pedestrian detection benchmarks

miss rate of 11.1%). As shown in Table 4, the combined features + pAUC$^{\text{struct}}$ outperform all previous best results on four major pedestrian detection benchmarks. Fig. 6 compares our best results (the last row in Table 4) with other state-of-the-art methods. Fig. 7 shows the spatial distribution of regions selected

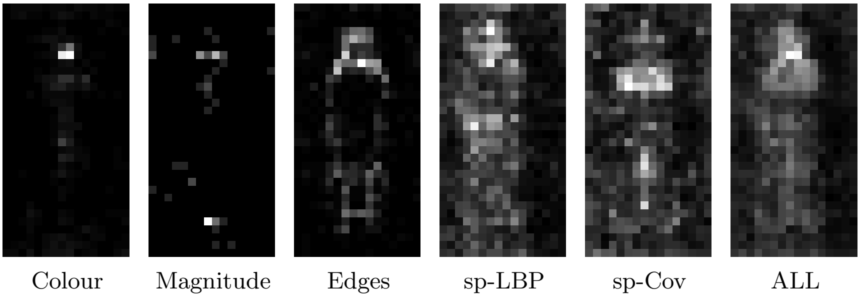

Fig. 7. Spatial distribution of selected regions based on their feature types. White pixels indicate that a large number of features are selected in that area. Often selected regions correspond to human contour and human body.

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.

It is important to point out that our significant improvement comes at the cost of increased computational complexity. We briefly list these additional computational costs compared to [12] here. (i) Additional CPU time to extract two additional features: sp-Cov and sp-LBP. (ii) The time taken to re-compute the confidence score of positive patches. To be more specific, we additionally calculate the dot product of the weak learners' output and pAUC$^{\text{struct}}$ variables (new coefficients for weak learners), *i.e.*, $\boldsymbol{w}^{\top}\boldsymbol{h}$ where $\boldsymbol{w}$ is pAUC$^{\text{struct}}$ variables, $\boldsymbol{h} = [h_1(\cdot), \cdots, h_t(\cdot)]$ and $h_k(\cdot)$ is the k-th weak learner. (iii) Additional CPU time to perform the global normalization (ACE). In our experiment, applying the colour normalization on a 640 × 480 pixels image takes approximately 0.3 seconds. This fast result is already based on an approximation of ACE [36], which estimates a slope function with an odd polynomial approximation and uses the DCT transform to speed up the convolutions. Using a single core Intel Xeon CPU 2.70GHz processor, our detector currently operates at approximately 0.126 frames per second (without global normalization) and 0.119 frames per second (with global normalization) on the Caltech data sets (detecting pedestrians larger than 50 pixels).

4 Conclusion

In this paper we propose a simple yet effective feature extraction method based on spatially pooled low-level visual features. To achieve optimal log-average miss rate performance measure, we learn another set of weak learners' coefficients whose aim is to improve the detection rate at the range of most practical importance. The combination of our approaches contributes to a pedestrian detector which outperforms all competitors on all of the standard benchmark datasets. Based on our experiments, we observe that the choice of discriminative features and implementation details are crucial to achieve the best detection performance. Future work includes incorporating motion information through the use of spatial and temporal pooling to further improve the detection performance.

Acknowledgements. This work was in part supported by Australian Research Council grant FT120100969.

References

[1] Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: An evaluation of the state of the art. IEEE Trans. Pattern Anal. Mach. Intell. 34, 743–761 (2012)
[2] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., vol. 1 (2005)
[3] Shen, C., Wang, P., Paisitkriangkrai, S., van den Hengel, A.: Training effective node classifiers for cascade classification. Int. J. Comp. Vis. 103 (2013)
[4] Paisitkriangkrai, S., Shen, C., Zhang, J.: Fast pedestrian detection using a cascade of boosted covariance features. IEEE Trans. Circuits Syst. Video Technol. 18, 1140–1151 (2008)
[5] Tuzel, O., Porikli, F., Meer, P.: Pedestrian detection via classification on Riemannian manifolds. IEEE Trans. Pattern Anal. Mach. Intell. 30, 1713–1727 (2008)
[6] Viola, P., Jones, M.J.: Robust real-time face detection. Int. J. Comp. Vis. 57, 137–154 (2004)
[7] Wang, X., Han, T.X., Yan, S.: An HOG-LBP human detector with partial occlusion handling. In: Proc. IEEE Int. Conf. Comp. (2009)
[8] Park, D., Zitnick, C.L., Ramanan, D., Dollár, P.: Exploring weak stabilization for motion feature extraction. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2013)
[9] Sermanet, P., Kavukcuoglu, K., Chintala, S., LeCun, Y.: Pedestrian detection with unsupervised multi-stage feature learning. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2013)
[10] Wang, X., Yang, M., Zhu, S., Lin, Y.: Regionlets for generic object detection. In: Proc. IEEE Int. Conf. Comp. Vis. (2013)
[11] Yang, J., Yu, K., Gong, Y., Huang, T.: Linear spatial pyramid matching using sparse coding for image classification. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2009)
[12] Dollár, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Trans. Pattern Anal. Mach. Intell. 1 (2014)
[13] Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part based models. IEEE Trans. Pattern Anal. Mach. Intell. 32, 1627–1645 (2010)
[14] Ouyang, W., Zeng, X., Wang, X.: Modeling mutual visibility relationship with a deep model in pedestrian detection. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2013)
[15] Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2009)

[16] Benenson, R., Mathias, M., Tuytelaars, T., Gool, L.V.: Seeking the strongest rigid detector. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2013)

[17] Lim, J.J., Zitnick, C.L., Dollár, P.: Sketch Tokens: A learned mid-level representation for contour and object detection. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2013)

[18] Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: Proc. of British Mach. Vis. Conf. (2009)

[19] Walk, S., Majer, N., Schindler, K., Schiele, B.: New features and insights for pedestrian detection. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn., San Francisco, US (2010)

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

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

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

[23] Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. Pattern Anal. Mach. Intell. 24, 971–987 (2002)

[24] Tuzel, O., Porikli, F., Meer, P.: Region covariance: A fast descriptor for detection and classification. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. Part II. LNCS, vol. 3952, pp. 589–600. Springer, Heidelberg (2006)

[25] Jia, Y., Huang, C., Darrell, T.: Beyond spatial pyramids: Receptive field learning for pooled image features. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2012)

[26] Bo, L., Ren, X., Fox, D.: Multipath sparse coding using hierarchical matching pursuit. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2013)

[27] Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms. In: Int. Conf. on Multimedia (2010)

[28] Paisitkriangkrai, S., Shen, C., van den Hengel, A.: Efficient pedestrian detection by directly optimizing the partial area under the roc curve. In: Proc. IEEE Int. Conf. Comp. Vis. (2013)

[29] Wu, J., Brubaker, S.C., Mullin, M.D., Rehg, J.M.: Fast asymmetric learning for cascade face detection. IEEE Trans. Pattern Anal. Mach. Intell. 30, 369–382 (2008)

[30] Narasimhan, H., Agarwal, S.: $\mathrm{SVM}_{\mathrm{pAUC}}^{\mathrm{tight}}$: 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)

[31] Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Prediction, Inference and Data Mining. Springer (2009)

[32] Friedman, J., Hastie, T., Tibshirani, R.: Additive logistic regression: A statistical view of boosting. Ann. Stat. 28, 337–407 (2000)

[33] Appel, R., Fuchs, T., Dollár, P., Perona, P.: Quickly boosting decision a trees-pruning underachieving features early. In: Proc. Int. Conf. Mach. Learn. (2013)

[34] Rizzi, A., Gatta, C., Marini, D.: A new algorithm for unsupervised global and local color correction. Patt. Recogn. 24, 1663–1677 (2003)

[35] Ouyang, W., Wang, X.: Single-pedestrian detection aided by multi-pedestrian detection. In: Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (2013)

[36] Getreuer, P.: Automatic color enhancement (ACE) and its fast implementation. Image Proc. On Line 2012 (2012)