Exploring Prior Knowledge for Pedestrian Detection

Yi Yang[1]
yangyi@nlpr.ia.ac.cn
Zhenhua Wang[2]
wzh@ntu.edu.sg
Fuchao Wu[1]
fcwu@nlpr.ia.ac.cn

[1] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, China

[2] Rapid-Rich Object Search Lab
School of EEE
Nanyang Technological University
Singapore

Abstract

In this paper, we aim to explore the role of prior knowledge for pedestrian detection. The main idea is to integrate human body priors into the design of features. To this end, we propose the symmetric features and cross-channel features so as to capture the specific information of human body. Experimental results demonstrate that our detector achieves state-of-the-art performance. What's more, the evaluation results on "scale" subsets of Caltech-USA show that our detector performs best at medium scale and therefore has great potential to be integrated into real-world applications.

1 Introduction

Pedestrian detection is a classical and hot issue in object detection. Well established benchmark data sets [5, 8, 14] make it a playground to explore good ideas for object detection. Although many approaches have been proposed in this area, it remains a challenging problem due to the variances in lighting conditions, scene structures, clothes, view angles, postures, scales, occlusions, *etc*.

As summarized in the recent survey [3], using better features plays an important role in improving detection quality. Similar analysis can be found in [9] that carefully combining multiple features can significantly boost detection performance. Selecting better features from huge feature pools [7, 10, 20, 24, 30] is a recent trend. In addition, prior knowledge has shown good success in designing haar-like features for pedestrian detection [30]. In that work, they exploit a shape prior of human body to generate feature templates so as to capture local differences around human body silhouette. On the other hand, performance of the same feature pool can present huge differences when using different channel combinations. Fortunately, some researchers [7, 10, 30] have demonstrated that a 10-channel combination of HOG+LUV always performs well.

Inspired by [30], our work aims to integrate more prior knowledge into the design of features to enhance performance of pedestrian detection. By observing the pedestrian samples, we have discovered several important priors that are always ignored by previous methods, *e.g.*, the symmetry of human body and the differences among different channels. Intuitively,

Pages 176.1-176.12
DOI: https://dx.doi.org/10.5244/C.29.176

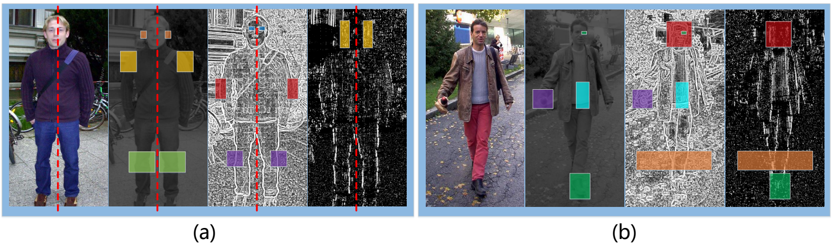


Figure 1: (a) Symmetric features. For clarity, we just show three channels (L channel, gradient magnitude, and one gradient orientation) here. Rectangles with the same color in the same channel represent one symmetric feature. (b) Cross-channel features. Rectangles with the same color in different channels represent one cross-channel feature.

these priors should be helpful. We therefore utilize these priors to design two kinds of features: 1) symmetric features which capture the difference between two local symmetric regions, and 2) cross-channel features which capture the difference between two different channels of the same region. Figure 1 gives some visual examples of these two features. To the best of our knowledge, we are the first to use symmetric and cross-channel priors in designing features for pedestrian detection. Experiments show that our detector achieves state-of-the-art performance, which demonstrates that the prior information helps a lot in designing features.

The remainder of this paper is organized as follows. We summarize our main contributions in next subsection, followed by a review of related works in Sec. 2. In Sec. 3 we introduce our symmetric and cross-channel features and the rules for generating feature pool. Analysis of selecting features is presented in Sec. 4. Subsequently, we report our extensive experiments in Sec. 5 and conclude this paper in Sec. 6.

1.1 Contributions

Our main contribution is to integrate prior knowledge into the design of features for pedestrian detection. We explore two human body priors and experimentally evaluate their effectiveness for pedestrian detection.

Symmetric prior: Human is approximately bilaterally symmetrical about the middle line of the body. Intuitively, symmetric prior will be a helpful cue for pedestrian detection. We therefore design a kind of symmetric features to capture the symmetric information contained by two local symmetric regions.

Cross-channel prior: As previous works mainly focus on information contained by the same channel, they ignore the cross-channel characteristics presented in different channels. Some valuable information can be obtained by comparing the cross-channel characteristics. Accordingly, we design a kind of cross-channel features to capture these information.

2 Related Works

Recent survey [3] reviews 40+ methods and divides them into three families: DPM variants [12, 13, 22, 25], deep networks [15, 18, 21, 29], and decision forests [7, 10, 24, 30]. All the three families reach top performance in pedestrian detection. The most relevant method with ours is InformedHaar [30], which is based on decision forest. Therefore, in this section, we focus on methods of decision forest family and pay much attention on channels, feature types, and extra information that have been used in these methods.

Channels: Dollár *et al.* first introduce channels in their paper [7] and build the foundation of decision forest family for pedestrian detection with channels. They explore various kinds of channels and dig out a best combination: HOG+LUV. These channels are regarded as the "core" channels from ChnFtrs [7] to the state-of-the-art detector LDCF [20]. Recently, other than "core" channels, some extra channels or transformations on channels have been considered. Typical extra channels are LBP and covariance feature channels used in SpatialPooling(+) [23, 24], word channels proposed in WordChannels [4] which are based on high level visual words, and self-similarity channels adopted in SketchTokens [17]. Other than using extra channels, LDCF expands the 10 "core" channels to 40 channels by convolving the "core" channels with four learned filters and achieves state-of-the-art performance.

Feature types: The simplest feature type is to directly use pixel values in channels as feature values (SketchTokens). To improve robustness, the aggregated channel feature family (ACF [10], LDCF, and SpatialPooling(+), *etc.*) divides channels into small blocks and sums pixels in each block as feature values. In addition, rectangle feature family (ChnFtrs, SquaresChnFtrs [2], and InformedHaar) considers first-order or higher-order rectangle features. More specifically, a first-order feature is defined as a sum of pixels in a fixed rectangular region in a single channel and higher-order features are defined as any feature that can be computed using multiple first-order features [7]. Rectangle features can be computed efficiently via integral image. Beyond those, SketchTokens employs self-similarity features which capture the portions of an image patch that contain similar textures.

Extra information: It has been shown that leveraging some extra information at training and testing time can improve detection quality [3]. Context information (ground plane constraint: MultiResC [22] and RandForest [19]; 2Ped: JointDeep [21] and MultiResC +2Ped [22]) and optical flow (ACF+SDt [26] and SpatialPooling(+)) are most commonly used. By adding SquaresChnFtrs, LDCF, SDt, and 2Ped together, Katamari [3] reaches superior performance on Caltech-USA test set. In addition, stereo images [16], tracking [11], and lidar data [27] are also considered as extra information.

3 Feature Design

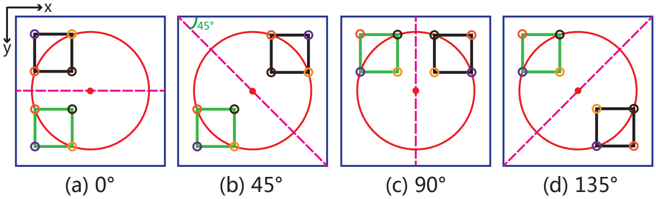
3.1 Symmetric Features

Pedestrians usually appear up-right in image, making pedestrian detection benefit from some favorable constraints and become easier than general human detection. A pedestrian body shape model can be obtained by computing an average edge map based on gradient magnitudes extracted from a large number of training samples [30]. The shape model seems like a fairly standard silhouette of a person standing facing the front, with hands falling naturally on body sides and feet keeping naturally together. In addition to shape information used by [30], we can find more valuable information from the shape model, so as to improve the performance of pedestrian detection. It is worth to mention that, such a fairly standing body silhouette is symmetrical about the vertical middle line of the model, as the red dashed lines shown in Figure 1(a). This symmetry should be a good prior for designing effective features.

Based on this observation, we design a kind of symmetric features to capture the symmetric prior. For convenient implementation and efficient computation, we constrain our feature templates to be rectangles. More specifically, we define the symmetric features to be second-order rectangle features which are composed of two separate local symmetric rectangular regions. The two local symmetric regions share the same size and should be in the same channel. Figure 1(a) gives some examples of these symmetric regions. The feature values can be effectively computed as the difference between the responses of these two symmetric regions by using integral image.

To enrich the symmetric information, we actually consider 3 more symmetry axes with 0°, 45°and 135°symmetry angles (see Figure 2(a), (b), and (d) respectively) in our implementation. With a specific symmetry axis, the symmetry angle is defined as the angle between the positive direction of x-axis and this symmetric axis, such as the 45°angle shown in Figure 2(b). In addition, as the whole human body is not symmetrical about these three additional axes, we actually divide the model region into 4 subregions and generate symmetric rectangles in these subregions to capture the local symmetric information. Figure 3(c) gives an illustration of the model region division.

With the definition above, symmetric rectangles should have the same size, locate in the same channel, and be symmetrical about a specific symmetry axis. Therefore, to generate symmetric rectangles, we first randomly generate one rectangle in the model region. Then, we obtain the 4 vertexes of another rectangle by mapping the vertexes of the first rectangle about the symmetry axis. For a better understanding, we show some simple examples in Figure 2. It can be found that the two corresponding rectangles can capture the symmetric information of the red circle.



(a) 0°
(b) 45°
(c) 90°
(d) 135°

Figure 2: Some simple examples of symmetric rectangles. The red dashed lines are symmetry axes. The big blue, small black, and small green rectangles represent model region, first rectangle and second rectangle respectively. The small circles with the same color indicate the mapping of corresponding vertexes about the symmetry axis.

3.2 Cross-channel Features

In recent pedestrian detectors with multiple channels, *e.g.*, HOG+LUV channels, different channels contain different kinds of information. Previous methods mainly use features that capture the information in a single channel, *e.g.*, `ChnFtrs` only takes responses of rectangles in one channel at a time. Such features inevitably lose information cross different channels. To deal with this problem, we propose a kind of cross-channel features to capture such valuable information by comparing the responses of the same rectangle in different channels. Note that the cross-channel features are also second-order features as they also contain two rectangular regions. Differently, the two rectangles share the same size and position but locate in different channels. The difference of responses between these two rectangles is recorded as feature value. Note that the features of different channels should be normalized to make them comparable. To generate the cross-channel rectangles, we first randomly generate a rectangle in the model region, and then assign it with two different channel indexes. Figure 1(b) gives some examples of the cross-channel features.

3.3 Generating Feature Pool

Due to the limitation of computational power and memory, a largely over-complete feature pool is not suitable. Fortunately, a randomly generated small feature pool can achieve state-of-the-art performance as well [2]. In our implementation, we randomly generate 25,000 features, which consists of the same amount of symmetric features, cross-channel features and the traditional haar features.

The details for generating the proposed symmetric and cross-channel features are described as follows. Firstly, we denote a rectangle as a 4 dimensional vector $\mathbf{r} = (x, y, w, h)$ in which x, y are the coordinates of the top-left vertex and w, h are the width and height respectively. Then, the valid rectangle set $\mathcal{R}$ is defined as the set of all possible rectangles that are inside of the model region and larger than a predefined area threshold S. In an usual image coordinate system, *e.g.* with the origin being the top-left vertex of the model region, the positive x-axis pointing right and the positive y-axis pointing down, such a set can be represented as:

$$\mathcal{R} = \{\mathbf{r_i} : x_i \leq W - w_i, y_i \leq H - h_i, w_i \leq W, h_i \leq H, w_i \times h_i \geq S, x_i, y_i, w_i, h_i \in \mathbb{N}\} \quad (1)$$

where W and H are the width and height of the model region respectively. With the valid rectangle set, we can further generate a pool of rectangular templates. For symmetric features, we select two symmetric rectangles $\mathbf{r_i}$ and $\mathbf{r_j}$ from $\mathcal{R}$ and randomly generate their channel index. For cross-channel features, we select one rectangle $\mathbf{r_i} \in \mathcal{R}$ and randomly generate two different channel indexes.

4 Selecting Features

Training details: AdaBoost is usually employed to select features from a large feature pool. In this work, we apply a fast version of AdaBoost [1] for learning. Our final strong classifier is composed of 4096 depth-5 decision trees. We apply a multi-round training strategy to build our strong classifier. Specifically, all ground truth regions and their reflections are extracted as positive samples, and are kept the same in all rounds. In the first round, negative samples are randomly extracted from background area of training images, while in the following rounds, we apply the classifier trained in the last round to all training images and add the false positives as negative samples to retrain a new classifier. The training procedure is stopped after 4 rounds or when the loss function reaches a pre-defined error.

Visualization of selected features: We give a visualization of the selected top features in this section. First, we plot the heat maps of the top 100 selected features for Caltech-USA and INRIA data sets in Figure 3(a). As shown in this figure, the most discriminative features are concentrated on lower body region and upper body region for Caltech-USA and INRIA data sets, respectively. This may due to the different characteristics of these two data sets. Pedestrians in Caltech-USA data set are relatively small and lack of details in upper body region, especially head-shoulder region, while they are more clearer in lower body region. Instead, pedestrians in INRIA data set seem to be much bigger and contain more

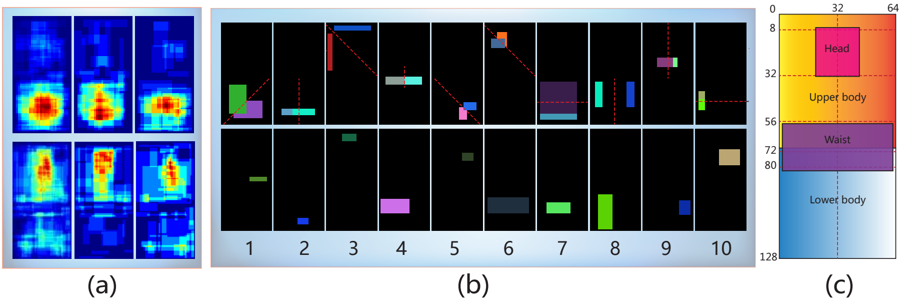


Figure 3: (a) Heat maps of the top 100 selected features for Caltech-USA (first row) and INRIA (second row) data sets. Each row from left to right: all, symmetric and cross-channel features in the top 100 selected features. (b) Top 10 selected symmetric (first row) and cross-channel (second row) features for Caltech-USA data set. (c) Illustration of the 4 subregions of a model region. This figure is better viewed in color mode.

details, especially in the face region. Next, we show the top 10 selected symmetric and cross-channel features for Caltech-USA data set in Figure 3(b). For symmetric features, we use two different colors to distinguish the two symmetric rectangles (may have overlap). Note that we generate symmetric rectangles in subregions (Sec. 3.1) and therefore the symmetry axes should exist in each subregion, *e.g.*, the red dashed lines in Figure 3(b). For cross-channel features, we only show one rectangle as the two rectangles share the same location and size.

The selected most discriminative features are then used for pedestrian detection in still images. We consider multiple scales and slide a window over the whole image of each scale. In our implementation, we set the spatial step to be 4 and the number of scales in each octave to be 8. As there are many repeated detections, we then simply use a non-maximal suppression (NMS) algorithm [7] to suppress nearby repeated detections.

5 Experiments

Our implementation is based on Dollár's open source toolbox [6]. We conduct our experiments on two public benchmark data sets: the INRIA [5] and Caltech-USA [8] pedestrian data sets. For Caltech-USA data set, we conduct a dense sampling of the training data (every 4 frames) following the configuration of `LDCF`. As a result, we obatin a training set with 32,077 images.

5.1 Parameter Analysis

In this section, we explore the influence of different parameter settings on Caltech-USA validation set (some both on Caltech-USA validation set and INRIA test set). We set up the validation set the same as [9], which splits the six training videos into two parts: the first five for training and the last one for testing.

Feature effectiveness: To evaluate the effectiveness of the proposed features, we generate 4 feature pools, each of which contains 30,000 (10,000 for INRIA) symmetric (sym), cross-channel (cross), haar, and sym+cross+haar features respectively. For sym+cross+haar feature pool, we keep the same amount of the three features. The results are shown in Figure 4(a). Not surprisingly, symmetric features and cross-channel features perform a little worse than haar features as they are not designed to be individually used as a general feature, *e.g.*, 10,000 for each one. These two features aim to capture some specific characteristics of pedestrian and supply complementary information for other features. As can be observed, by combining these two features with haar feature, the performance could be improved.

Number of features: Intuitively, more features will lead to better performance as they contain more information. Nevertheless, due the limit of computing power and memories, we can not exhaustively generate all possible features in the feature pool. Fortunately, a randomly generated small feature pool can achieve state-of-the-art performance as well. We therefore restrict the maximal number of candidate features to be 30,000 and evaluate performances of different sizes of feature pool. As shown in Figure 4(b), the best performance is achieved with 30,000 features, while performance with 25,000 features is competitive as well.

Channels: The "core" channels (HOG+LUV) are used as the baseline of channel combination. We further add LBP channel and use LDCF [20] to expand the channels by 4 times. As shown in Figure 4(c), the best performance is obtained by using "HOG+LUV+LBP+LDCF" and the use of "core" channels also performs competitively.

Classifier: We evaluate the performance for different number of weak classifiers and different tree depths. As can be observed in Figure 4(d), the best performance is achieved by using 3072 weak classifiers. A small number of weak classifiers may not be distinctive enough due to the large variances of pedestrians, while too many number of weak classifiers may lead to overfitting. Figure 4(e) shows that a depth-5 tree performs the best as it can sufficiently exploit the information in the rich channels.

Smoothing: Figure 4(f) and Figure 4(g) show the evaluations for pre-smoothing and post-smoothing with binomial filters, respectively. Specifically, without pre-smoothing on colors achieves best performance and using larger radius results in worse results. Post-smoothing on channels seems to have a slightly effect on performance.

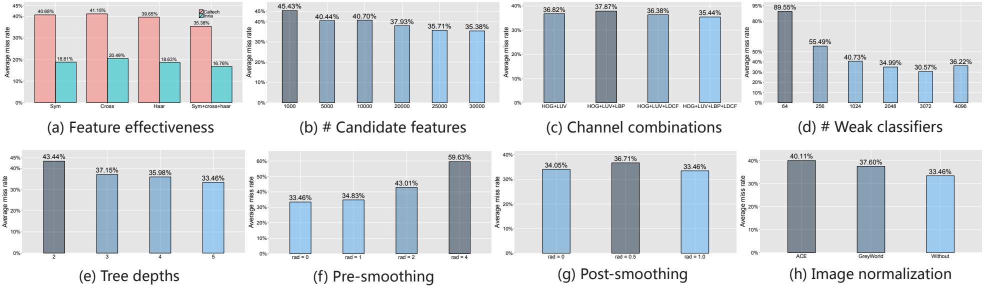
Image normalization: The influences of different image normalization algorithms are shown in Figure 4(h). We only evaluate global image normalization as local normalization seems to be ineffective [2, 30]. The same as `Roerei` [2], automatic color equalization (ACE) and GreyWorld [28] are considered. Different from `Roerei`, we obtain best performance without any normalization.

With the above analysis, we set the parameters for testing on Caltech-USA test set as follows: 25,000 sym+cross+haar features (10,000 for INRIA); HOG+LUV+LBP+LDCF channels (no LDCF for INRIA); 4096 weak classifiers with depth-5 (2048 depth-2 trees for INRIA); no pre-smoothing; post-smoothing of radius 1; no image normalization. Note that we use all the six training videos to train the detector for testing on Caltech-USA test set. Due to the increase of training data, we actually use 4096 weak classifiers.

5.2 Comparison with State-of-the-art Detectors

In this section, we compare our results with some state-of-the-art results on Caltech-USA and INRIA test sets. For a fair comparison, we use the public evaluation code of [9]. The evaluation criterion is the ROC curve and the overall performance is summarized by average miss rate.

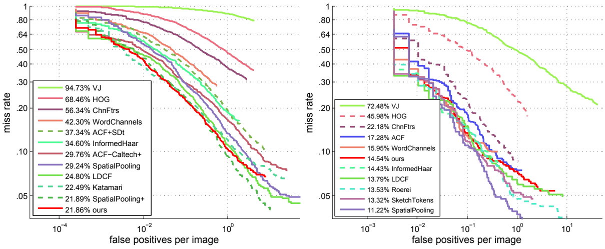
Figure 5(a) shows the results on the "reasonable" [9] subset of Caltech-USA test set. Our detector `ours` not only outperforms baseline detectors `ChnFtrs` and `LDCF` by about 34%



(a) Feature effectiveness (b) # Candidate features (c) Channel combinations (d) # Weak classifiers

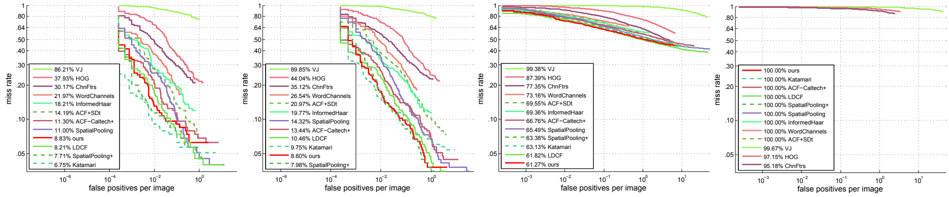
(e) Tree depths (f) Pre-smoothing (g) Post-smoothing (h) Image normalization

Figure 4: Evaluation of different parameters on Caltech-USA validation set.



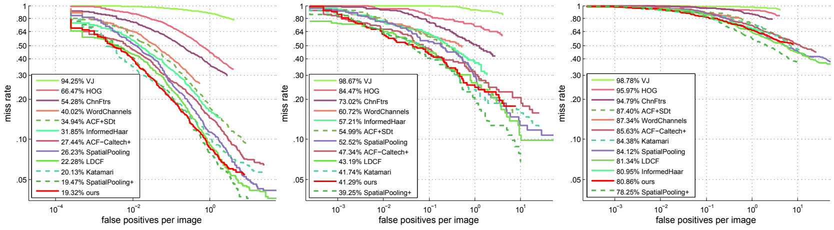
(a) Caltech-USA test set (b) INRIA test set

Figure 5: Results of different detectors on Caltech-USA and INRIA test sets. Dashed lines in (a) represent detectors using motion features.



(a) Scale = large (b) Scale = near (c) Scale = medium (d) Scale = far

Figure 6: Results under different "scale" subsets of Caltech-USA test set. Detectors using motion features are dashed.



(a) Occ = none (b) Occ = partial (c) Occ = heavy

Figure 7: Results under different "occlusion" subsets of Caltech-USA test set. Detectors using motion features are dashed.

Detector	INRIA	Caltech-USA							
		Reas -onable	Scale				Occlusion		
			Large	Near	Medium	Far	Non	Partial	Heavy
VJ	72.48%	94.73%	86.21%	89.85%	99.38%	99.67%	94.25%	98.67%	98.78%
HOG	45.98%	68.46%	37.93%	44.04%	87.39%	97.15%	66.47%	84.47%	95.97%
ChnFtrs	22.18%	56.34%	30.17%	35.12%	77.35%	**95.18%**	54.28%	73.02%	94.79%
ACF	17.28%	51.36%	23.48%	28.71%	76.44%	96.81%	48.86%	71.55%	94.72%
ACF-Caltech+	–	29.76%	11.30%	13.44%	66.76%	100%	27.44%	47.34%	85.63%
LDCF	13.79%	24.80%	**8.21%**	10.46%	61.82%	100%	22.28%	43.19%	81.34%
WordChannels	15.95%	42.30%	21.97%	26.54%	73.16%	100%	40.02%	60.72%	87.34%
InformedHaar	14.43%	34.60%	18.21%	19.77%	69.36%	100%	31.85%	57.21%	80.95%
Roerei	13.53%	48.35%	16.07%	21.79%	74.16%	97.40%	45.82%	68.49%	90.38%
SketchTokens	13.32%	–	–	–	–	–	–	–	–
SpatialPooling	**11.22%**	29.24%	11.00%	14.32%	65.49%	100%	26.23%	52.52%	84.12%
*ACF+SDt	–	37.34%	14.19%	20.97%	69.55%	100%	34.94%	54.99%	87.40%
*SpatialPooling+	–	21.89%	7.71%	7.98%	63.38%	100%	19.47%	39.25%	78.25%
*Katamari	–	22.49%	6.75%	9.75%	63.13%	100%	20.13%	41.74%	84.38%
ours	14.54%	**21.86%**	8.83%	**8.60%**	**61.27%**	100%	**19.32%**	**41.29%**	**80.86%**

Table 1: Performance comparisons for state-of-the-art detectors under various conditions. The average miss rates for different datasets or their subsets are summarized in corresponding columns. * indicates detectors using motion features, the bold ones indicate best performance among detectors without using motion features, and the underlined ones indicate best performance among all the tested detectors. Our detector achieves three best performances among all the tested detectors and six best performances among the detectors without using motion features.

and 3% respectively, but also outperforms all state-of-the-art detectors even including detectors which consider additional motion features (`Katamari` and `SpatialPooling+`) and an exhaustive over-complete feature pool (`Katamari`).

Results on INRIA test set are shown in Figure 5(b). Our detector achieves comparable results with state-of-the-art detectors. Due to the difference of performances on Caltech-USA and INRIA data sets, we conclude that our features are better at detecting relatively small pedestrians, *i.e.*, pedestrians which are far from the camera, like the smaller ones in Caltech-USA data set rather than the bigger ones in INRIA data set. We further evaluate this conjecture by the following experiments about "scale".

Figure 6 shows the experimental results respect to "scale". We consider four "scale" levels: scale = large (100 pixels or taller), scale = near (80 pixels or taller), scale = medium (30 to 80 pixels), and scale = far (20 to 30 pixels) follow [9]. The performances of all detectors drop significantly as scale reduces. Fortunately, our detector seems to be better at dealing with relatively small scales and achieves overall best performance at medium scale. It is worth to mention that, detection at medium scale is critical for automotive applications [9]. With the common vehicle speed of 55 km/h, the person which is 1.5 s to 4 s away is about 30 to 80 pixels in a normal 720p camera. Therefore, detecting too large (near) pedestrians seems to leave insufficient time to alert the driver, while too small (far) pedestrians seem to be less relevant.

We also evaluate our detector respect to "occlusion" and plot the results in Figure 7. We consider three "occlusion" levels: occ = none (no occlusion), occ = partial (1-35% occluded), and occ = heavy (35-80% occluded) as in [9]. Although the performances of all detectors drop significantly as occlusion increases, our detector `ours` always keeps a top 2 ranking. Actually, it achieves best performance among all detectors without using motion features in all levels of occlusion and is a slightly worse than `SpatialPooling+`, which considers optical flow as motion features, for the cases of partial and heavy occlusion. We conclude that our proposed features are relatively robust against occlusions.

For a more clearer comparison, we summarize all the results from Figure 5 to Figure 7 in Table 1. As summarized in this table, our detector achieves three best performances among all the tested detectors and six best performances among the detectors without using motion features.

5.3 Runtime

We implement our detector in Matlab, on an Intel Core-i5 CPU (3.1GHz). It takes 22 hours for training 4 rounds on Caltech-USA data set (every 4 frames sampling). An average time for testing on a 640 × 480 image is 3.88 seconds. We further evaluate the time costs of different components of our detector and show the results in Table 2. As can be observed, half of the time are spent on computing integral images as there are a total of 44 channels and 27 scales (one integral image for each channel in each scale). Fortunately, we can parallel these computations to further speed up.

	Channel pyramid	LDCF	Integral image	Sliding window detection	All
Time (seconds)	0.23	0.70	1.58	0.75	3.88

Table 2: Detailed (average) time statistics for each computation.

6 Conclusion

In this paper, we have explored two human body priors for pedestrian detection by integrating symmetric and cross-channel information into feature design. With these useful prior knowledge, our detector achieves superior performance and even outperforms detectors which take consideration of motion features. Furthermore, evaluation results on "scale" subsets of Caltech-USA test set demonstrate that our detector has great potential to be integrated into real-world applications.

References

[1] Ron Appel, Thomas Fuchs, Piotr Dollár, and Pietro Perona. Quickly boosting decision trees-pruning underachieving features early. In *JMLR Workshop and Conference Proceedings*, volume 28, pages 594–602. JMLR, 2013.

[2] Rodrigo Benenson, Markus Mathias, Tinne Tuytelaars, and Luc Van Gool. Seeking the strongest rigid detector. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3666–3673. IEEE, 2013.

[3] Rodrigo Benenson, Mohamed Omran, Jan Hosang, and Bernt Schiele. Ten years of pedestrian detection, what have we learned? *In ECCV, CVRSUAD workshop*, 2014.

[4] Arthur Daniel Costea and Sergiu Nedevschi. Word channel based multiscale pedestrian detection without image resizing and using only one classifier. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 2393–2400. IEEE, 2014.

[5] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[6] Piotr Dollár. Piotr's Computer Vision Matlab Toolbox (PMT). http://vision.ucsd.edu/~pdollar/toolbox/doc/index.html.

[7] Piotr Dollár, Zhuowen Tu, Pietro Perona, and Serge Belongie. Integral channel features. In *BMVC*, volume 2, page 5, 2009.

[8] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: A benchmark. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 304–311. IEEE, 2009.

[9] Piotr Dollár, Christian Wojek, Bernt Schiele, and Pietro Perona. Pedestrian detection: An evaluation of the state of the art. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):743–761, 2012.

[10] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36 (8):1532–1545, 2014.

[11] Andreas Ess, Bastian Leibe, Konrad Schindler, and Luc Van Gool. Robust multiperson tracking from a mobile platform. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(10):1831–1846, 2009.

[12] Pedro Felzenszwalb, David McAllester, and Deva Ramanan. A discriminatively trained, multiscale, deformable part model. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[13] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(9):1627–1645, 2010.

[14] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012.

[15] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 580–587. IEEE, 2014.

[16] Christoph G Keller, Markus Enzweiler, Marcus Rohrbach, David Fernandez Llorca, Christoph Schnorr, and Dariu M Gavrila. The benefits of dense stereo for pedestrian detection. *Intelligent Transportation Systems, IEEE Transactions on*, 12(4):1096–1106, 2011.

[17] Joseph J Lim, C Lawrence Zitnick, and Piotr Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3158–3165. IEEE, 2013.

[18] Ping Luo, Yonglong Tian, Xiaogang Wang, and Xiaoou Tang. Switchable deep network for pedestrian detection. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 899–906. IEEE, 2014.

[19] Javier Marin, David Vázquez, Antonio M López, Jaume Amores, and Bastian Leibe. Random forests of local experts for pedestrian detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2592–2599. IEEE, 2013.

[20] Woonhyun Nam, Piotr Dollár, and Joon Hee Han. Local decorrelation for improved pedestrian detection. In *Advances in Neural Information Processing Systems*, pages 424–432, 2014.

[21] Wanli Ouyang and Xiaogang Wang. Joint deep learning for pedestrian detection. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 2056–2063. IEEE, 2013.

[22] Wanli Ouyang and Xiaogang Wang. Single-pedestrian detection aided by multi-pedestrian detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3198–3205. IEEE, 2013.

[23] Sakrapee Paisitkriangkrai, Chunhua Shen, and Anton van den Hengel. Pedestrian detection with spatially pooled features and structured ensemble learning. *arXiv preprint arXiv:1409.5209*, 2014.

[24] Sakrapee Paisitkriangkrai, Chunhua Shen, and Anton van den Hengel. Strengthening the effectiveness of pedestrian detection with spatially pooled features. In *Computer Vision–ECCV 2014*, pages 546–561. Springer, 2014.

[25] Dennis Park, Deva Ramanan, and Charless Fowlkes. Multiresolution models for object detection. In *Computer Vision–ECCV 2010*, pages 241–254. Springer, 2010.

[26] Dennis Park, C Lawrence Zitnick, Deva Ramanan, and Piotr Dollár. Exploring weak stabilization for motion feature extraction. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2882–2889. IEEE, 2013.

[27] Cristiano Premebida, Joao Carreira, Jorge Batista, and Urbano Nunes. Pedestrian detection combining rgb and dense lidar data. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 4112–4117. IEEE, 2014.

[28] Alessandro Rizzi, Carlo Gatta, and Daniele Marini. A new algorithm for unsupervised global and local color correction. *Pattern Recognition Letters*, 24(11):1663–1677, 2003.

[29] Pierre Sermanet, Koray Kavukcuoglu, Soumith Chintala, and Yann LeCun. Pedestrian detection with unsupervised multi-stage feature learning. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3626–3633. IEEE, 2013.

[30] Shanshan Zhang, Armin B. Cremers, and Christian Bauckhage. Informed haar-like features improve pedestrian detection. *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 947 – 954, 2014.