Pyramid Center-symmetric Local Binary/Trinary Patterns for Pedestrian Detection

Yongbin Zheng, Chunhua Shen, Richard Hartley and Xinsheng Huang

Australian National University and NICTA, Canberra

Abstract. Detecting pedestrians in images plays a very important role in many computer vision applications such as video surveillance, smart cars and robotics. Feature extraction is the key for this task. Promising features should be discriminative, robust and easy to compute. This paper presents a novel and efficient feature, termed pyramid center-symmetric local binary\ternary patterns (pyramid CS-LBP\LTP), for pedestrian detection. The CS-LBP feature combines the desirable properties of the standard LBP, which can be viewed as texture-based features, and the gradient based feature. Moreover, the pyramid CS-LBP\LTP is easy-to-implement and computationally efficient. Experiments on the INRIA pedestrian dataset show that the proposed feature outperforms Histograms of Oriented Gradients (HOG) feature and comparable with the start-of-the-art pyramid HOG(PHOG) features, when using the Intersection Kernel SVM classifier. Our experiments also show that the combination of our pyramid LBP feature and the PHOG feature could improve the detection performance significantly.

1 Introduction

The ability to detect pedestrians in images has deep impact to applications such as video surveillance, smart vehicles, robotics and so on. Large variations in human body, pose and clothing, combined with varying cluttered backgrounds and environmental conditions, make this problem far from being solved. In the past few years, there has been a surge of interest in pedestrian detection [1–9]. One of the leading approaches in pedestrian detection is based on sequentially applying a classifier at all the possible subwindows, which are obtained by entirely scanning the input image in different scales and positions. For each sliding window, certain feature sets are extracted and fed to the classifier, which is trained beforehand using a set of labeled training data of the same type of features. The classifier then determines whether the sliding window contains a pedestrian [1].

Inspired by the development of object detection and classification, higher and higher performance of pedestrian detection has been achieved by: (1) using highly discriminative and robust image features, such as Haar wavelets [1], region covariance [5, 7] ,HOG [3, 4] and PHOG [10] (2) using the combination of multiple complementary features [9] (3) including spatial information [10] (4) the choices

of classifiers, such as SVM [3, 10], AdaBoost [11]. Feature extraction plays a critical role here. The features should be robust, discriminative, compact and efficient. The HOG feature is one of the best and most popular features used for pedestrian detection [3]. One of its drawbacks is the heavy computation. Maji et al. [10] introduced the PHOG feature into pedestrian detection, and their experiment showed that the new features can lead better classification accuracies than the HOG and much computational simpler and have smaller dimensions. However, these HOG like features, which capture the edge or the local shape, could perform poorly when the background is cluttered with noisy edges [9].

Our goal here is to develop a feature extraction method for pedestrian detection that, in comparison to the state-of-the-art, is comparable in performance and faster to compute. A conjecture is that, if both the shape and texture information are extracted and used in the feature for pedestrian detection, the detection accuracy is likely to increase. The Center-symmetric local binary patterns feature(CS-LBP) [12], which is a modified version of the famous texture feature LBP, inherit the desirable properties of both the texture feature and the gradients feature, and is computationally simple. In this paper, we propose the pyramid center-symmetric local binary\ternary patterns(Pyramid CS-LBP \LTP) features for pedestrian detection. The experiments on INRIA dataset show that our new features outperform HOG and comparable with the state of art PHOG, when using the Intersection Kernel SVM(IKSVM) classifier [10]. We also found that the detection performance can be improved significantly by combining our feature with the PHOG feature.

The rest of the paper is organized as follows. In Section 2, we briefly describe the LBP\LTP operator, the CS-LBP\LTP features and the pyramid CS-LBP\LTP features. In Section 3, we give the details of our approach. The experimental evaluation is carried out in section 4. Section 5 concludes the paper.

2 Preliminaries

2.1 LBP and LTP features

LBP is a texture descriptor which codifies local primitives (such as curved edges, spots, flat areas) into a feature histogram. LBP and its extensions outperform existing texture descriptors both with respect to performance and to computational efficiency [13].

The basic version of LBP feature of a pixel is assigned by thresholding the 3×3-neighborhood of each pixel with the center pixel's value . Let g_c be the center pixel graylevel and $g_i\,(i = 0, 1, \cdots, 7)$ be the graylevel of each surrounding pixel. If g_i is smaller than g_c, the binary result of the pixel is set to 0, otherwise to 1. All the results are combined to a 8-bit binary value. The decimal value of the binary is the LBP feature. See Fig. 1 for an illustration of computing the basic LBP feature.

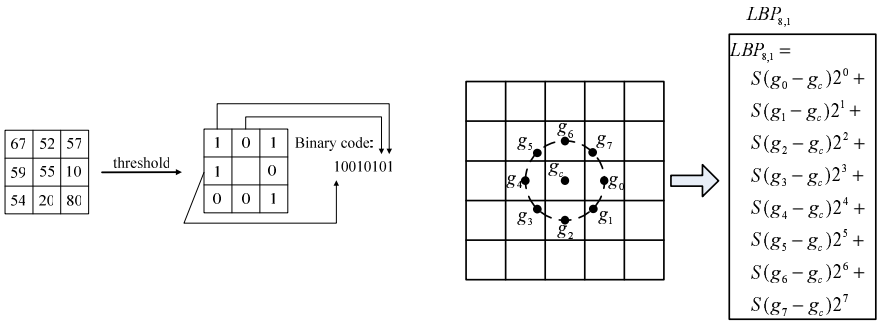
In order to be able to deal with textures at different scales, the original LBP has been extended to arbitrary circular neighborhoods [14] by defining the

neighborhood as a set of sampling points evenly spaced on a circle centered at a pixel to be labeled. It allows any radius and number of sampling points. Bilinear interpolation is used when a sampling point does not fall in the center of a pixel. Let $\mathrm{LBP}_{p,r}$ denote the LBP feature of a pixel's circular neighborhoods, where r is the radius of the circle and p is the number of sampling points on the circle. The $\mathrm{LBP}_{p,r}$ can be computed as Eq. (1):

$$\mathrm{LBP}_{p,r} = \sum_{i=0}^{p-1} S(g_i - g_c)2^i,\ S(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

Here g_c is the center pixel's graylevel and $g_i\,(i = 0, 1, \cdots, 7)$ is the graylevel of each sampling pixel on the circle. See Fig. 1 for an illustration of computing the LBP feature of a pixel's circular neighborhoods with $r = 1$ and $p = 8$.

Ojala *et al.* [14] proposed "uniform pattern" to reduce the original LBP pattern numbers while keeping its discrimination power. An LBP pattern is called uniform if the binary pattern contains at most two bitwise transitions from 0 to 1 or *vice versa* when the bit pattern is considered circular. For example, the bit pattern 11111111 (no transition), 00001100 (two transitions) are uniform whereas the pattern 01010000 (four transitions) is not. Uniform pattern reduces the LBP pattern number from 256 to 58 and is successfully applied to face detection in [15].



(a) Illustration of the basic LBP operator.

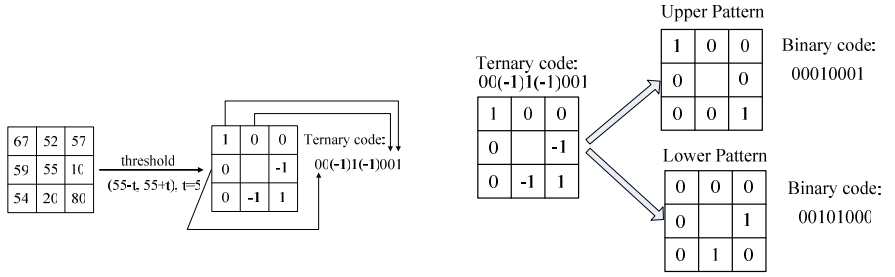
(b) The LBP operator of a pixel's circular neighborhoods with $r = 1$, $p = 8$.

Fig. 1. The LBP operator

LBP tends to be sensitive to noise, particularly in near-uniform image regions, and to smooth weak illumination gradients, Tan and Triggs [16] extended LBP to 3-valued codes, called local trinary patterns(LTP). If each surrounding graylevel g_i is in a zone of width $\pm t$ around the center graylevel g_c, the result value is quantized to 0. The value is quantized to $+1$ if g_i is above this and is quantized to -1 if g_i is below this. The $LTP_{p,r}$ can be computed as :

$$\mathrm{LTP}_{p,r} = \sum_{i=0}^{p-1} S(g_i - g_c)2^i,\ S(x) = \begin{cases} 1 & \text{if } x \geq t, \\ 0 & \text{if } |x| < t, \\ -1 & \text{if } x \leq t, \end{cases} \tag{2}$$

Here t is a user-specified threshold. Fig. 2(a) illustrates the encoding procedure of LTP. For simplicity, Tan and Triggs [16] used a coding scheme that splits each ternary pattern into its positive and negative halves as illustrated in Fig. 2(b), treating these as two separate channels of LBP codings for which separate histograms are computed, combining the results only at the end of the computation.



(a) Illumination of the basic LTP operator.

(b) Splitting the LTP code into positive and negative LBP codes

Fig. 2. The LTP operator

2.2 The CS-LBP/LTP patterns

The CS-LBP is another modified version of LBP. It is originally proposed to alleviate some drawbacks of the standard LBP. For example, the original LBP histogram could be very long and the original LBP feature is not robust on flat images. As illustrated in Fig. 3, instead of comparing graylevel of each pixel with the center pixel, the center-symmetric pairs of pixels are compared. The CS-LBP features can be computed by:

$$\begin{aligned} CS\text{-}LBP_{p,r,t} &= \sum_{i=0}^{N/2-1} S(g_i - g_{i+(N/2)})2^i, \\ S(x) &= \begin{cases} 1 & \text{if } x \geq t, \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{3}$$

Here g_i and $g_{i+N/2}$ correspond to the graylevel of center-symmetric pairs of pixels of N equally spaced on a circle of radius r. t is a small value and is used to

threshold the graylevel difference to increase the robustness of CS-LBP feature on flat image regions. From the computation of CS-LBP, we can see that the CS-LBP is closely related to the gradient operator, because, like some graident operators, it considers graylevel differences between pairs of opposite pixels in a neighborhood. This way the CS-LBP feature take advantage of both the properties of the LBP and the gradient based features. In [12], the authors used the CS-LBP descriptor to describe the region around an interest point and their experiments show that the performance is almost equally promising as the popular SIFT descriptor. The authors also compared the computational complexity of the CS-LBP descriptor with the SIFT descriptor and it has been shown that the CS-LBP descriptor is on average 2 to 3 times faster than the SIFT. That is because the CS-LBP feature needs only simple arithmetic operations while the SIFT requires time consuming inverse tangent computation when computing the gradient orientation.

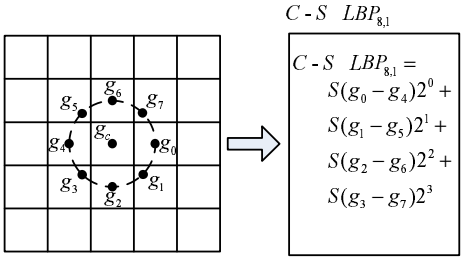


Fig. 3. The CS-LBP features for a neighborhood of 8 pixels.

Similarly as "uniform LBP pattern", we propose "uniform CS-LBP pattern" to reduce the original CS-LBP pattern numbers. A CS-LBP pattern is called uniform if the binary pattern contains at most one bitwise transition from 0 to 1 or vice versa. For example, the pattern 0000(no transition) and 0111(one transition) are uniform whereas the pattern 0010(two transitions) and 1010(three transitions) are not. We computed the CS-LBP patterns of 741 images in INRIA dataset(288 images containing pedestrains and 453 images without dedestrians) and found that 87.82% of the patterns are uniform, shown in Table 1.

The CS-LTP and the uniform CS-LTP can be developed similarity as the CS-LBP and the uniform CS-LBP.

2.3 Pyramid CS-LBP/LTP features and pyramid uniform CS-LBP/LTP features

Inspired by the image pyramid representation in [17] and the HOG [3], Bosch et al. [18] proposed the PHOG descriptor, which consists of a pyramid of Histograms of orientation gradients, to represent an image by its local shape and the

Table 1. The distribution of the CS-LBP pattern

Uniform	0000	0001	0011	0111	1000	1100	1110	1111	
Percentage(%)	8.93	11.80	8.72	10.22	8.31	9.27	10.99	19.57	total(87.82)
Non-uniform	0010	0100	0101	0110	1001	1010	1011	1101	
Percentage(%)	1.24	1.14	1.52	1.28	1.86	1.31	1.73	2.11	total(12.18)

spatial layout of the shape. Experiments showed that PHOG feature together with the histogram intersection kernel can bring significant performance to object classification and recognition. Maji *et al.* [10] introduced the PHOG feature into pedestrian detection and led to the current state of the art on pedestrian detection.

In this study, we propose the pyramid CS-LBP\LTP features. Because the LTP patterns can be divided into two LBP patterns, we only illustrate the computation of the pyramid CS-LBP features. The details of our features are illustrated as follows:

1) We compute the CS-LBP value and the norm of each pixel of the input image. The LBP value is computed as Eq. 3 and the norm of the pixel located at (x, y) is computed as: $norm(x, y) = \sqrt{G_x^2(x, y) + G_y^2(x, y)}$, where $G_x(x, y)$ and $G_y(x, y)$ are the horizontal gradient and vertical gradient of the pixel. Then we obtain 16 layers of norm images corresponding to each CS-LBP pattern. We call them edge energy responses of the input image .

2) Each layer of the response image is L_1 normalized in non overlapping cells of fixed size $y_n \times x_n$.

3) At each level $l \in \{1, 2, ...L\}$, the response image is divided into non overlapping cells of size $y_l \times x_l$, and a histogram feature is constructed by summing up normalized response within the cell.

4) The histogram feature of each level is normalized to sum to unity. This normalization ensures that the edge or texture rich images are not weighted more strongly than others.

5)The features at a level l are weighted by a factor w_l, and the features at all the levels are combined to form a vector, which is called pyramid CS-LBP features. Fig. 4 shows the first three steps of computing the features.

The precess of computing pyramid uniform CS-LBP is almost same as pyramid CS-LBP. The only difference lies in the first step. In the first step, the edge energy responses of different uniform patterns are count into different layers and the edge energy response of all the non-uniform patterns are count into one layer. So we obtain 9 layers of edge energy responses of the input image.

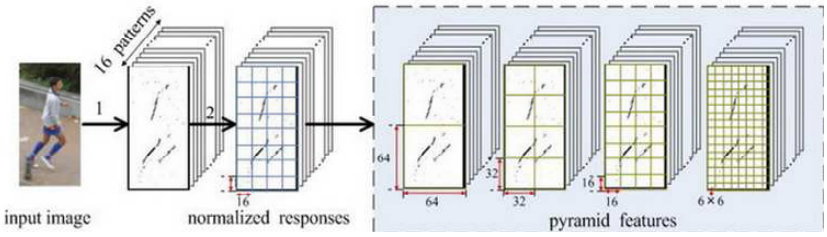


Fig. 4. Pyramid CS-LBP features.

3 Pedestrian detection based on pyramid CS-LBP\LTP features

We use the sliding window approach. The first major component of our approach is feature extraction. We perform the graylevel normalization of the input image to deduce the illumination variance. After the normalization is performed, all the input image have the same intensity ranged from 0 to 1. Then,the detection window slides on the input images in all positions and scales, with a fixed scale factor and a fixed step size. we follow the steps in Sec. 2.3 to compute the pyramid CS-LBP\LTP features of each detection window.

The second major component of our approach is the employed classifier. We use the histogram intersection kernel SVM(IKSVM) [10] as classifier. The histogram intersection kernel, $k_{HI}(h_a, h_b) = \sum_{i=1}^{n} min(h_a(i), h_b(i))$ is often used as a measurement of similarity between histogram h_a and h_b and it can be used as a kernel for discriminate classification using SVMs. Compared to linear SVMs, histogram intersection kernel requires great computational expense. Maji *et al.* [10] approximated the histogram intersection kernel for faster execution. Their experiments showed that the approximate IKSVM consistently outperforms linear SVM at a modest increase in running time.

The third major component of our approach is the merging of the multiple overlapping detections using non maximal suppression(NMS). After the merging, detections with bounding boxes and confidence scores are obtained.

Detecting pedestrians on the INRIA human dataset, our approach has the following parameters: 128×64 detection windows, detection window slides with a step size 8×8 and a scale factor 1.0905, block normalization window size of 16×16, 4 pyramid levels, cell size of 64×64,32×32,16×16,6×6 at levels 1,2,3 and 4 respectively, the weights to the features of levels are 1,2,4 and 9 respectively.

4 Experiments

4.1 Experiment setup

Datasets. We perform the experiments on INRIA human dataset [3], which is the most popular publicly available dataset and has helped drive recent ad-

vances in pedestrian detection. The dataset consists of training set and test set. The training set contains 1208 images of size 96×160 pixels (a margin of 16 pixels around each side) of human samples (2416 mirrored samples) and 1218 pedestrian-free images. The test set contains 288 images with human samples and 453 human free images. The human samples are cropped from a varied set of personal photos and vary in pose, clothing, illumination, background and partial occlusions, what make the dataset is very challenge.

Methodology. *Per-window* performance is accepted as the methodology for evaluating pedestrian detectors by most authors. But this evaluating methodology is flawed. As mentioned in [8], *per-window* performance can fail to predicted *per-image* performance. There may be at least two reasons: first, *per-window* evaluation does not measure errors caused by detections at incorrect scales or positions or arising from false detections on body parts, nor does it take into account the effect of NMS. Second, the *per-window* scheme uses cropped positives and uncropped negatives for training and testing: classifiers may exploit window boundary effects as discriminative features leading to good *per-window* but poor *per-image* performance. In this paper, we use *per-image* performance, plotting detection rate versus false positives per-image(FPPI).

We select the 2416 mirrored human samples from the training set as positive training examples. A fixed set of 12180 patches sampled randomly from 1218 pedestrian-free training images as initial negative set. Same as [3], a preliminary IKSVM detector is trained and the 1218 negative training images are searched exhaustively for false positives. The classifier is then retrained using the augmented set combined by the initial training set and the found false positives. The SVM tool we used is the fast intersection kernel SVMs proposed by Maji et al. [10] and it can be download from: `http://www.cs.berkeley.edu/~smaji/projects/fiksvm/`.

We detect pedestrian on each test images (both positive and negative) in all positions and scale. Multiscale and nearby detections are merged using NMS and a list of detected bounding boxes are given out. Evaluation on the list of detected bounding box is done using the PASCAL criterion which counts a detection to be correct if the overlap of the detected bounding box and ground truth bounding box is greater than 0.4.

4.2 Performance of the pyramid CS-LBP/LTP feature based detector

In this section, we study the performance of our approach by comparing with the state of art PHOG feature based approach. We obtain the PHOG based detector from its author and the PHOG's level number and cell size in each level are same as our features. The results are shown in Fig. 5. The performance of pyramid CS-LTP based detector performs best, with detection rate over 80% at 0.5 FPPI. Then followed by the pyramid uniform CS-LTP based detector, which is slightly better than the PHOG based detector. The pyramid CS-LBP based detector performs almost as good as the PHOG. Though the pyramid uniform CS-LBP based detector performs slightly worse than PHOG basd detector, it

outperforms the HOG features with linear SVM based detector proposed by Dalal and Triggs [3].

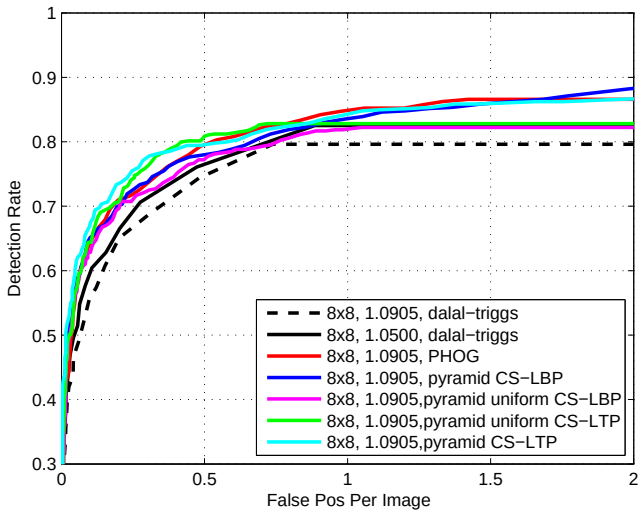


Fig. 5. Detection rate versus false positive per-image curves for detectors based on different features.

4.3 Detection results with features combined by PHOG and pyramid CS-LBP

We explore the performance of augmented features combined by the PHOG and the pyramid CS-LBP (PHOG + pyramid uniform CS-LBP) in Fig. 6. The detection rate versus FPPI curves show that the augmented feature can significantly improve the detection performance, especially when the FPPI is small. The detection rate raises about 6% at 0.25 FPPI and raises about 1.5% at 0.5 to 1 FPPI. Fig. 7 shows pedestrian detection on some example test images. The three rows show the bounding boxes detected by PHOG based detector, pyramid uniform CS-LBP based detector and the PHOG + pyramid uniform CS-LBP based detector, respectively.

5 Conclusions

We have presented pyramid CS-LBP\LTP features for pedestrian detection problems. The experimental results on the INRIA dataset show that the pyramid

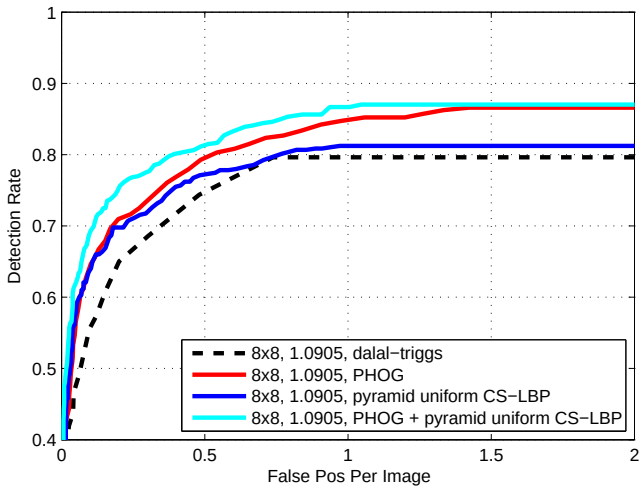


Fig. 6. Performance of detector

Fig. 7. Some examples of detections on test images for the detectors using PHOG, pyramid uniform CS-LBP and augmented features(combined by HOG and pyramid uniform CS-LBP). First row: detected by the PHOG based detector. Second row: detected by the pyramid uniform CS-LBP based detector. Third row: detected by the PHOG + pyramid uniform CS-LBP based detector.

CS-LTP features using IKSVM classifier outperform the PHOG using IKSVM classifier and the pyramid CS-LBP features perform as good as the HOG. Further experiments show that a pedestrian detector based on the augmented features combined by the PHOG and the pyramid CS-LBP can achieve significantly better performance.

There are many directions for further research. To make the conclusion more convincing, the performance of the pyramid CS-LBP\LTP features based pedestrian detector needs to be further evaluated on other dataset, e.g. the Caltech Pedestrian Dataset [8]. Another further study will be to compare the computational complexity of the pyramid CS-LBP\LTP features with PHOG both theoretically and experimentally. Thirdly, it is worthy to study how to combine our features with PHOG or other features more efficiently. We are also interested in implement the new feature in the boosting framework.

References

1. Viola, P., Jones, M., snow, D.: Detecting pedestrian using patterns of motion and appearance. In: Proc. Int'l Conf. Computer vision. (2003) 734–741
2. Mikolajczyk, K., Schmid, C., Zisserman, A.: Human detection based on a probabilistic assembly of robust part detectors. In: European Conference on Computer Vision. (2004) 69–81
3. Dalal, N., Triggs, B.: Histogram of oriented gradients for human detection. In: Proc. IEEE Conf. on Computer Vision and Pattern recognition. Volume 1., IEEE (2005) 886–893
4. Dalal, N.: Finding people in images and videos. PhD thesis, Institut National Polytechnique de Grenoble (2006)
5. Tuzel, O., Porikli, F., Meer, P.: Human detection via classification on riemannian manifolds. In: Proc. IEEE Conf. Computer vision and Pattern Recognition. (2007) 1–8
6. Munder, S., Gavrila, D.M.: An experimental study on pedestrian classification. IEEE Trans. on Pattern Analysis and Machine Intelligence **28(11)** (2006) 1863–1868
7. Paisitkriangkrai, S., Shen, C., Zhang, J.: Fast pedestrian detection using a cascade of boosted covariance features. IEEE Trans. on Circuits and Systems for Video Technology **18(8)** (2008) 1140C1151
8. Dollár, P., Wojek, C., Schiele, B., Perona, P.: Pedestrian detection: A benchmark. In: Proc. IEEE Conf. Computer vision and Pattern Recognition. (2009)
9. Wang, X., Han, T., Yan, S.: An HoG–LBP human detector with partial occlusion handling. In: Proc. 12th IEEE International Conf. on Computer Vision, IEEE (2009)
10. Maji, S., Berg, A., Malik, J.: Classification using intersection kernel support vector machines is efficient. (2008) 1–8
11. Wojek, C., Walk, S., Schiele, B.: Multi-cue onboard pedestrian detection. In: EEE Conference on Computer Vision and Pattern Recognition. (2009) 794–801
12. Heikkila, M., Schmid, C.: Description of interest regions with local binary patterns. Pattern Recognition **42** (2009) 425–436
13. Ojala, T., Pietikainen, M., Harwood, D.: A comparative study of texture meatures with classification based on featured distribution. Pattern Recognition **29** (1996) 51–59

14. Ojala, T., Pietikainen, M., Maenpaa, T.: Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Trans. on Pattarn Analysis and machine Intelligence **24** (2002) 429–436
15. Ahonen, T., Hadid, A., Pietikainen, M.: Face detection with local binary patterns: application to face recognition. IEEE Trans. on Pattarn Analysis and machine Intelligence **28** (2006) 2037–2041
16. Tan, X., Triggs, B.: Enhanced local texture feature sets for face recognition under difficult lighting conditions. IEEE Trans. on on Image Processing **19(6)** (2010) 1635–1650
17. Bosch, A., Zisserman, A., Munoz, X.: Scene classification via plsa. In: Proceedings of the European Conference on Computer Vision. Volume IV. (2006) 517–530
18. Bosch, A., Zisserman, A., Munoz, X.: Representing shape with a spatial pyramid kernel. In: Proceedings of the 6th ACM international conference on Image and video retrieval. (2007) 401–408