International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)
Web Site: www.ijettcs.org Email: editor@ijettcs.org, editorijettcs@gmail.com
Volume 9, Issue 5, September - October 2020 ISSN 2278-6856

Dual Superpixel HOG Pedestrian Detector

Harvey Barnett Mitchell[1] and Daniel Asher Mitchell[2]

[1]Rehov Brosh, Mazkeret Batya, Israel

[2]Rehov Brosh, Mazkeret Batya, Israel

Abstract: ***The Histogram-of-Oriented-Gradient (HOG) is a widely used feature used in many pattern recognition applications involving pedestrian detection. The basic idea of HOG is that the local pedestrian appearance can be characterized by the distribution of local intensity gradients and edge directions. In this letter we describe a dual superpixel HOG algorithm in which we fuse together two HOG feature vectors. The first vector is the traditional HOG feature vector calculated on the input image. The second vector is a HOG feature vector which is calculated on the input image after superpixel segmentation. By fusing the two HOG vectors together we obtain a fused HOG with an enhanced performance while at the same time being fully compatible with the traditional HOG. Experimental results on standard pedestrian detection databases show that for noisy input the dual HOG significantly outperforms the traditional HOG detector.***

Keywords: Histogram-of-oriented-gradient, Pedestrian detection, superpixel, HOG, image processing

1. INTRODUCTION

Pedestrian detection is an important branch of pattern recognition used in many applications such as video surveillance [1, 2, 3], biometrics [4, 5], driving assistance systems [6], re-identification [7], car safety [8] and robotics [9, 10]. Detecting pedestrians in a static image is challenging because of the wide variability in pedestrian appearance, illumination and background. Nevertheless, during the last decade, pedestrian detection has attracted world-wide research efforts and great progress has been made [11, 12, 13, 14]. An important step in all pedestrian detection algorithms is feature extraction, and many different features have been employed for this purpose. The image features can be broadly grouped into hand-crafted features [15, 16, 17] and deep convolutional neural network (CNN) features [18, 19, 20, 21].

In general the CNN features have the best performance. However, the CNN has several drawbacks: they require a very large amount of training data and have a long training time. In addition, the CNN's are usually complex with a very high computational load. On the other hand, the traditional methods require much less training data, are much simpler to train and have a much lower computational load. They often employ a sliding window paradigm with hand-crafted features and a traditional classifier. Among the hand-crafted features, the histogram of oriented gradient (HOG) [22] descriptor is the most well-known and may be used with any convenient classifier, e.g. the K-nearest neighbor or linear Support Vector Machine (SVM) [23]. Since its introduction by [22], HOG has been intensively researched [24, 25] and widely used for real-time, or near real-time, applications requiring pedestrian detection with limited computational resources [24, 26, 27, 28].

In this letter we show how we may improve the performance of the HOG descriptor, and in particular make it more robust against image noise and blur. The robustness of the new descriptor is built into the algorithm by fusing together [29] two complementary image gradient feature vectors. The two feature vectors are:

1. **Traditional**. The first feature vector calculates the image gradients on the input image as in the traditional HOG detector.
2. **Superpixel**. The second feature vector calculates the image gradients on the input image after it has been segmented into superpixels using any standard superpixel algorithm.

We fuse together the two feature vectors to give us a new HOG vector. The new HOG has the same size as the traditional HOG and may be used without modification, wherever the traditional HOG detector is used. By fusing together two complementary sets of image gradients the new HOG has an improved performance combined with robustness against additive Gaussian noise and Gaussian blur. This is verified in a series of pedestrian detection experiments.

2.HISTOGRAM OF ORIENTED GRADIENTS (HOG)

The basic idea of a HOG feature is that the local object appearance and shape can be characterized by the distribution of local intensity gradients or edge directions. Suppose $I(x,y)$ denotes the intensity of a pixel (x,y) in the image I. Then the main steps in extracting the HOG descriptor [22, 26] are:

1. **Gradient Calculation**. Compute first-order gradients at each pixel:

$$G_x(x,y) = I(x+1,y) - I(x-1,y), \quad (1)$$

$$G_y(x,y) = I(x,y+1) - I(x,y-1), \quad (2)$$

where $G_x(x,y)$ and $G_y(x,y)$ represent, respectively, the horizontal gradient and vertical gradient at the pixel (x,y). Alternative equations for G_x and G_y have been investigated in the literature [22] who found the simple $(1,0,-1)$ gradient formula used in Eqs. (1) and (2) to be optimal. Using Eqs. (1) and (2), the intensity gradient and edge direction at (x,y) are given by:

$$G(x,y)=\sqrt{G_x^2(x,y)+G_y^2(x,y)}\,, \tag{3}$$

$$\theta(x,y)=\arctan\frac{G_y(x,y)}{G_x(x,y)}\,, \tag{4}$$

where $\theta(x,y)\in[0,\pi]$.

4. **Histogram**. The sliding window is divided into large (partially overlapping) spatial regions (called "blocks"). Each block is then divided into 2×2 small square regions (called "cells"). For each cell $C_m, m\in\{1,2,3,4\}$, we divide the $\theta(x,y)$ into 9 equal-width bins $\Delta\theta_h, h\in\{1,2,\ldots,9\}$. Then the histogram of the m th cell is computed as follows:

$$H_m(\theta_h)=\sum_{(x,y)\in C_m} v_h(x,y) \tag{5}$$

where

$$v_h(x,y)=\begin{cases} G(x,y) & \text{if } \theta(x,y)\in\Delta\Theta_h, \\ 0 & \text{otherwise}. \end{cases} \tag{6}$$

The four histograms $H_m, m\in\{1,2,3,4\}$, in a block are concatenated to produce a $36-D$ feature vector $[f(i,1)\,f(i,2)\ldots f(i,36)]$, where $[f(i,1)\,f(i,2)\ldots f(i,36)]$ denotes the feature vector for the i th block.

5. **L2-Hys Normalization**. In each block the $36-D$ feature vector is normalized using the L2-Hys norm. This is defined as L2-normalization followed by clipping maximum values to 0.2 and then re-normalization. Mathematically, the L2-Hys normalization is:

$$f_n(i,j)=\frac{g_n(i,j)}{\sum_{j=1}^{36} g_n^2(i,j))}\,, \tag{7}$$

where

$$g(i,j)=\frac{f(i,j)}{\sum_{j=1}^{36} f^2(i,j)}\,, \tag{8}$$

and

$$g_n(I,j)=min(g(i,j),0.2)\,. \tag{9}$$

6. **Concatenation**. The normalized $36-D$ feature vectors of all blocks in the sliding window are concatenated. This is the *normalized* HOG vector F_n :

$$F_n=[\,f_n(1,1)\ldots f_n(1,36)\,f_n(2,1)\ldots]. \tag{10}$$

3.SUPERPIXEL SEGMENTATION

The concept of superpixels was first described by [30]. Given an input image I , a superpixel algorithm groups the pixels into perceptually meaningful quasi-uniform regions. These regions, or "superpixels" are used to replace the characteristic rigid pixel-grid structure of I . In recent years, different superpixel methods have been proposed to improve the three most important superpixel characteristics: boundary adherence, uniform intensity and compactness. In our experiments, we use the Simple Linear Iterative Clustering (SLIC) algorithm [31]. This is a popular superpixel algorithm which is both simple to implement and computationally efficient. The majority of SLIC's superpixels have regular sizes and shapes, fairly uniform intensity and they adhere well to the image boundaries.

Let $S_k, k\in\{1,2,\ldots,K\}$, denote the set of superpixels in I . Suppose the superpixel S_k contains N_k pixels and has an average gray-level intensity $\overline{I}_k$. Then we may use the gray-level intensities $\overline{I}_k, k\in\{1,2,\ldots,K\}$, to define a superpixel segmented image $I'(x,y)$:

$$I'(x,y)=\overline{I}_k \text{ if } (x,y)\in S_k\,. \tag{11}$$

However, experimentally we found the iterative BTC algorithm [32,33,34] gave the best results. The steps in the algorithm are given in Algorithm 1.

Algorithm 1: Iterative BTC Algorithm
Input: Input image I and corresponding superpixels $S_k, k\in\{1,2,\ldots,K\}$
Output: Segmented image I'
$t_k=\overline{I}_k$
For *until convergence* **do**

Use threshold t_k to divide the pixels $(x,y) \in S_k$ into low and high value pixels:

$$T_k(x,y) = \begin{cases} 1 & \text{if } I(x,y) \geq t_k \\ 0 & \text{otherwise} \end{cases}$$

Calculate mean intensity of low and high value pixels by summing over all pixels $(x,y) \in S_k$:

$$a_k = \sum (1 - T_k(x,y)) I(x,y) / \sum (1 - T_k(x,y))$$

$$b_k = \sum T_k(x,y) I(x,y) / \sum T_k(x,y)$$

Calculate a new threshold: $t_k = (a_k + b_k)/2$

end do

Calculate the segmented image:

$$I'(x,y) = (a_k + b_k)/2 \text{ if } (x,y) \in S_k$$

In practice we found the superpixels converged within one or two iterations. In the experiments described in this article we limited the number of iterations to two.

4. DUAL SUPERPIXEL HOG

In the new dual superpixel HOG detector we calculate two *non-normalized* HOG vectors. The first vector is a non-normalized version of the traditional HOG vector:

$$F = \left[f(1,1)\ f(1,2) \ldots f(1,36)\ f(2,1) \ldots \right]. \qquad (12)$$

It is calculated by applying the traditional HOG algorithm to the input image I, but *without* the L2-Hys normalization. The second vector is a non-normalized superpixel HOG vector:

$$F' = \left[f'(1,1) \ldots f'(1,36)\ f'(2,1) \ldots \right]. \qquad (13)$$

It is calculated by applying the HOG algorithm to the segmented image I', but *without* the L2-Hys normalization.

We fuse F and F' together by multiplying them together element-by-element:

$$\hat{f}(i,j) = f(i,j) \times f'(i,j). \qquad (14)$$

The final dual HOG vector is obtained by L2-Hys normalizing $\hat{f}(i,j)$. We denote the normalized dual HOG vector as:

$$\hat{F}_n = \left[\hat{f}_n(1,1) \ldots \hat{f}_n(1,36)\ \hat{f}_n(2,1) \ldots \right]. \qquad (15)$$

5. EXPERIMENTS

We tested the dual HOG detector on the standard INRIA pedestrian database. This contains gray-scale images (size 64×128) of humans cropped from a varied set of personal photographs. We randomly selected 1218 of the images as positive training examples, together with their left-right reflections (2436 images in all). A fixed set of $12180 = 1218 \times 10$ patches sampled randomly from 1218 person-free training photos provided the negative set. A separate set of 352 positive images and 3520 negative images were selected from the INRIA database to be used as test samples. A simple linear SVM classifier was then used to classify the test images.

To evaluate the robustness with respect to distortions and noise, we considered additive Gaussian noise (standard deviation σ) and Gaussian image blurring (standard deviation σ). We use the original noise-free INRIA images for training while testing on the noisy images. The results shown are the average of 10 cross-validated independent runs.

Operating parameters for the HOG and the superpixel HOG algorithms are given in Table 1.

Table 1. HOG and SLIC operating parameters

HOG	
Cell size	8×8 pixels
Block size	2×2 cells
Block overlap	50%
SLIC	
Superpixel size	25 pixels
Superpixel compactness	10

We give the experimental results obtained with the traditional HOG and the new dual HOG as a 2×2 confusion matrix C and the corresponding precision P and recall R values:

$$C = \begin{pmatrix} TN & FP \\ FN & TP \end{pmatrix}, \qquad (16)$$

$$P = \frac{TP}{TP + FP} \times 100\%, \qquad (17)$$

$$R = \frac{TP}{TP + FN} \times 100\%, \qquad (18)$$

where TN (true negative) and TP (true positive) are, respectively, the number of non-pedestrians and pedestrians correctly identified as non-pedestrians and pedestrians; FP (false positive) is the number of non-pedestrians incorrectly identified as pedestrians and FN (false negative) is the number of pedestrians incorrectly identified as non-pedestrians.

In Tables 2 and 3 we give the confusion matrix (C) and precision (P)/recall (R) values as measured on the test data with additive Gaussian noise and Gaussian blur.

We see that for all additive noise levels, the average recall and the average precision of the dual HOG always exceeds that of the traditional HOG. The difference in the average recall increasing significantly as the noise level increases.

Table 2. Traditional vs Dual HOG: Additive Gaussian Noise

σ	Traditional HOG				Dual HOG			
	C		R	P	C		R	P
0	3568	32	91	91	3567	33	91	91
	32	320			32	320		
1	3560	40	91	90	3565	35	91	90

	32	320			31	321		
2	3567	33	90	91	3570	30	91	91
	36	316			34	319		
3	3569	31	86	91	3572	28	90	92
	48	304			37	315		
5	3575	25	79	92	3575	25	85	92
	75	277			51	301		
10	3586	14	59	94	3582	18	72	94
	146	206			98	254		

Table 3. Traditional vs Dual HOG: Gaussian Blur

σ	Traditional HOG				Dual HOG			
	C		R	P	C		R	P
1	3531	69	91	82	3501	99	93	77
	30	322			25	327		
2	3476	124	86	71	3407	193	90	62
	49	303			35	317		
3	3438	162	77	63	3327	273	85	52
	82	270			53	299		
4	3406	195	66	55	3259	341	77	45
	121	231			80	272		
5	3378	222	55	47	3206	394	72	40
	159	193			98	254		

For Gaussian blur, the average recall of the dual HOG always exceeds that of the traditional HOG albeit with a slight reduction in precision.

5.CONCLUSION

We have described an enhanced dual superpixel HOG pedestrian detector. The new detector uses information derived from a preliminary superpixel segmentation of the input image. The new HOG detector has the same input and output as the original HOG and may thus be used as a plug-in replacement for the original HOG. Detection results obtained on the standard INRIA pedestrian database show the the new HOG detector is very successful in the case of additive noise: the average recall and average precision always exceed that of the traditional HOG. For Gaussian blur, average recall of the dual HOG exceeds that of the traditional HOG, while the average precision of the dual HOG is less than that of the traditional HOG.

References

[1] V. Gajjar, Y. Khandhediya, and A. Gurnani, "Human detection and tracking for video surveillance: a cognitive science approach," IEEE Int. Conf. Comp. Vis. Workshops (ICCVW), pp. 2805-2809, 2017.

[2] J.-L. Chua, Y. C. Chang, and W. K. Lim, "A simple vision-based fall detection technique for indoor video surveillance," Sig. Imag. Video Proc., 9, pp. 623-633, 2015.

[3] M. Bilal, A. Khan, M. U. K. Khan, and C.-M. Chong-Min Kyung, "A low complexity pedestrian detection framework for smart video surveillance systems," IEEE Trans. Circ. Syst. Video Tech., 27, pp. 2260-2273, 2017.

[4] J. Neves, S Narducci, F. Narducci, S. Barra, and H. Proenca, "Biometric recognition in surveillance scenarios: a survey," Art. Intell. Rev., 46, pp. 515-541, 2016.

[5] I. Bouchrika, "A survey of using biometrics for smart visual surveillance: Gait recognition," In Surveillance in Action. Advanced Science and Technologies for Security Applications, P. Karampelas and T. Bourlai (eds), Springer, 2018.

[6] A. Shashua, Y. Gdalyahu, and G. Hayun, "Pedestrian detection for driving assistance systems: single-frame classification and system level performance," IEEE Intell. Vehicle Symp., pp. 1-6, 2004.

[7] L. Zheng, H. Zhang, S. Sun, M. Chandraker, Y. Yang and Q. Tian, "Person re-identification in the wild," Proc. IEEE Conf. Comp. Vis. Patt. Recogn. (CVPR), pp. 1367-1376, 2017.

[8] E. Coelingh, A. Eidehall, and M. Bengtsson, "Collision warning with full auto brake and pedestrian detection – a practical example of automeatic emergency braking," IEEE Int. Conf. Intell. Transport Syst., pp. 155-160, 2010.

[9] L. Dong, X. Yu, L. Li, and J. K. E. Hoe, "HOG based multi-stage object detection and pose recognition for service robot," Int. Conf. Contr. Automat. Robot Vis., pp. 2495-2500, 2010.

[10] O. H. Jafaril, D. Mitzell, and B. Leibel, "Real-time rgb-d based people detection and tracking for mobile robots and head-worn cameras," IEEE Int. Conf. Robot Automat. (ICRA), pp. 5636-5643, 2014.

[11] P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: an evaluation of the state of the art," IEEE Trans. Patt. Anal. Mach. Intell., 34, pp. 743-761, 2012.

[12] R. Trichet, and F. Bremond, "Dataset optimization for real-time pedestrian detection," IEEE Access, 6, pp. 7719-7727, 2018.

[13] R. Benenson, M. Omran, J. Hosang, and B. Schiele, "Ten years of pedestrian detection, what have we learned," EECV, CVRSUAD Workshop, 2014.

[14] M. Bilal and M. S. Hanif, "Benchmark revision for hog-svm pedestrian detector through reinvigorated training and evaluation methodologies," IEEE Trans. Intell. Transport, 21, pp. 1277-1287, 2020.

[15] C. Q. Lai and S. S. Teoh, "A survey of traditional and deep learning-based histogram of oriented gradient feature," IEEE Res. and Develop. (SCOReD), pp. 1-6, 2014.

[16] T. Georgiou, Y. Liu, W. Chen, and M. Lew, "A survey of traditional and deep learning-based features for high dimensional data in computer vision," Int. Journ. Multimedia Inform. Retrieval, 9, pp. 135-170, 2020.

[17] B. Fan, Z. Wang, and F. Wu, Local Image Descriptor: Modern Approaches, Springer, 2015.

[18] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew, "Deep learning for visual understanding: a review," Neurocomp., 187, pp. 27-48, 2016.

[19] A. Angelova, A. Krizhevsky, V. Vanhoucke, A. S. Ogale, and D. Ferguson, "Real-time pedestrian detection with deep network cascades," Proc. BMVC, pp. 1-12, 2015.
[20] J. Li, X. Liang, S. Shen, T. Xu, J. Feng, and S. Yan, "Scale-aware fast r-cnn for pedestrian detection," IEEE Trans. Multimedia, 20, pp. 985-996, 2018.
[21] W. Lan, J. Dang, Y. Wang, and S. Wang, "Pedestrian detection based on yolo network model," IEEE Int. Conf. Mechantron. and Automat. (ICMA), pp. 1547-1551, 2018.
[22] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," Int. Conf. Comp. Vis. Patt. Recogn. (CVPR '05), pp. 886-893, 2005.
[23] H. Bristow and S. Lucey, "Why do linear svm's trained on hog features perform so well?," ArXiv, 1406.2419, 2014.
[24] V.-D. Hoang, M.-H. Le, K.-H. Jo, "Hybrid cascade boosting machine using variant scale blocks based hog features for pedestrian detection," Neurocomp., 135, pp. 357-366, 2014.
[25] R. Trichet and F. Bremond, "How to train your dragon: Best practices in pedestrian classifier training," IEEE Access, 8, pp. 3527-3538, 2020.
[26] K. Piniarski, P. Pawlowski, and A. Dabrowski, "Tuning of classifiers to speed-up detection of pedestrians in infrared images," Sensors, 18, pp. 4363, 2020.