JCSE Journal of Computing Science and Engineering

Regular Paper

Journal of Computing Science and Engineering,
Vol. 8, No. 4, December 2014, pp. 207-214

An Efficient Pedestrian Detection Approach Using a Novel Split Function of Hough Forests

Trung Dung Do, Thi Ly Vu, Van Huan Nguyen, Hakil Kim*, and Chongho Lee
School of Information and Communication Engineering, Inha University, Incheon, Korea
{dotrungdung, vuthily, conghuan}@vision.inha.ac.kr, {hikim, chlee}@inha.ac.kr

Abstract
In pedestrian detection applications, one of the most popular frameworks that has received extensive attention in recent years is widely known as a 'Hough forest' (HF). To improve the accuracy of detection, this paper proposes a novel split function to exploit the statistical information of the training set stored in each node during the construction of the forest. The proposed split function makes the trees in the forest more robust to noise and illumination changes. Moreover, the errors of each stage in the training forest are minimized using a global loss function to support trees to track harder training samples. After having the forest trained, the standard HF detector follows up to search for and localize instances in the image. Experimental results showed that the detection performance of the proposed framework was improved significantly with respect to the standard HF and alternating decision forest (ADF) in some public datasets.

Category: Human computing

Keywords: Pedestrian detection; Object detection; Random forests; Hough forests; Boosting algorithm; Alternating decision forest

I. INTRODUCTION

Over the last decade, the problem of pedestrian detection has attracted the attention of many researchers in the image processing area and computer vision field, because such systems are in wide demand for a variety of modern applications in the real-world, such as video surveillance [1], autonomous navigation [2], and human-computer interaction [3]. Due to the importance of these applications in daily life and the drawbacks of current detection algorithms, the pedestrian detection problem has been studied intensively to satisfy the high requirements of real-world applications. Although researchers have been dedicating much effort to this task for a long time, there are still unresolved gaps due to the difficulties and challenges of the problem such as:

- Human occlusion: Because humans appear in various and unpredictable backgrounds, occlusion can occur at any time. Thus, to achieve high performance in human detection, the occlusion challenge needs to be effectively handled.
- Human articulation: The appearance of humans can be extremely varied, by changes in pose, distance, or the view point of the camera. Thus, human detection algorithms need to take this aspect into account to make the system more robust and accurate.
- Background noise: Changes in context due to weather conditions, illumination changes, and complex backgrounds are crucial reasons for detection failure or miss-detection.

Open Access http://dx.doi.org/10.5626/JCSE.2014.8.4.207 http://jcse.kiise.org

Received 17 April 2014; **Revised** 5 November 2014; **Accepted** 16 November 2014
*Corresponding Author

 pISSN: 1976-4677 eISSN: 2093-8020

- Processing time: Finally, processing time is a tough requirement for many real-world applications to operate in real time. The state-of-the-art approaches to human detection still need to be improved markedly to satisfy this requirement.

Generally, pedestrian detection algorithms include two phases, offline (training) and online (testing). First, in the offline phase, some parameter models are learned based on a collection of pedestrian and non-pedestrian samples by tuning the parameters in a supervised manner. These parameters are adjusted during the training process to fit the input data in the real world. To achieve this, machine learning algorithms, such as support vector machine (SVM) [4], artificial neural network (ANN) [5], boosting [6, 7], and random forests (RF) [8, 9], can be applied. Second, after models have been learned, they can be used to detect pedestrians and to localize their positions in images.

This paper focuses on the training phase by combining some popular existing machine learning techniques to construct a robust and accurate system for human detection. In fact, there are several approaches in the literature, such as holistic approaches, parts-based approaches, and patches-based approaches.

Holistic (global) approaches: This kind of approach exploits all information encoded in an entire object image rather than using a small set of features. Generally, the local visual descriptors are defined in a fixed order to match with the reference instance (template matching). One weakness of this approach is the demand for a large number of reference instances because of the appearance changes due to, for example, poses and illumination. In [10], the authors used Dominant Orientation Templates (DOT) [11] for fast feature calculation, and a holistic approach to detect the regions of interest in the test image rapidly. These detected regions were then used for the post-processing of pedestrian detection to achieve high performance and low processing time. However, approaches using the DOT features for pedestrian detection are sensitive to illumination changes, noise, and pose variations due to the computation of some dominant orientations in DOT rather than consideration of all orientations as in the histogram of oriented gradients (HOG) [12].

Parts-based (local) approach: Typical parts-based approaches to object detection consider local pieces of the image, and classify the pieces into background or foreground using machine learning algorithms, then building the object detector based on the pieces. However, such approaches cannot cope with major difficulties in the detection task, when the object of interest is small, or imaging conditions are unfavorable. To enhance the performance of the classical parts-based approach, the geometric information of the object is taken into account by modeling the position relationship of the parts of the object. In the work of Murphy et al. [13], the local features are used to describe parts extracted from images. The parts near the center object will be considered as a positive example, and the parts outside the object's bounding box will be a negative one. Then, these extracted parts are passed to the boosting classifier for training data. To overcome the limitation of local feature mentioned above, authors of this paper introduced the gist of an image. This gist captured texture and spatial layout of an image that can be learned as global features. Using both the local features and global features, the performance of the detection task can be improved significantly.

Patches-based approaches: This approach was introduced by Leibe et al. [14]; it generates a codebook based on sets of positive patches (presence of object) and negative patches (no object present). In the detection process, extracted patches of the unseen image are used to match against the codebook entries for finding the pedestrian hypothesis. This approach can be used to detect the presence of a pedestrian in the scene and to localize its position by analyzing the obtained object hypothesis. However, when the training data are significantly huge, or the dimension of feature is very high, the problems of creating and processing the codebook become highly expensive. To address this, the RF technique [15, 16] or the HF [17-19] framework can be applied to efficiently and rapidly generate a codebook even when the dimensions of the feature are infinite.

The purpose of this study was to develop a patches-based approach by providing a novel split function for the RF techniques as well as its extension for the HF frameworks. During forest construction, the errors are consecutively optimized, using the global loss function [20, 21] in each stage of the training process. The rest of this paper is organized as follows: Section II briefly describes

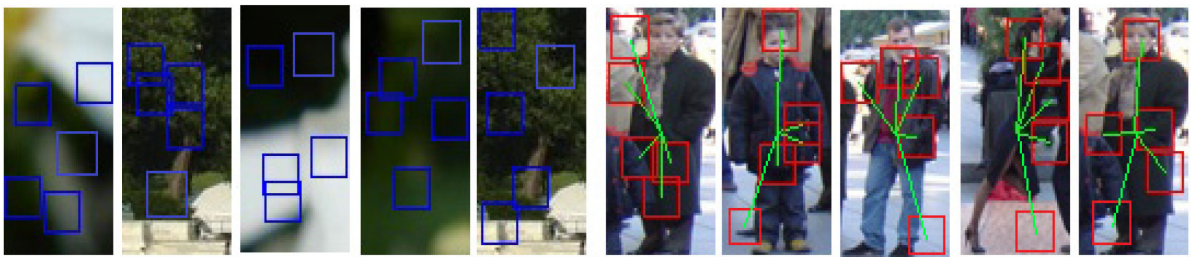

Fig. 1. Pedestrians IRINA training data set (blue rectangles and red rectangles are patches randomly extracted from negative and positive images, respectively. Green lines indicate offsets).

the HF framework and the proposed novel split function, and applied boosting idea to the HF framework. Experimental results and analysis compared with the standard HF framework are shown in Section III. Finally, Section IV provides the conclusions and discusses future work.

II. NOVEL SPLIT FUNCTION AND ERROR OPTIMIZATION FOR HF

A. Novel Split Function for HF Framework

The HF framework proposed by Gall and Lempitsky [17] can be considered as a derivation of the RF [8], which consists of a collection of trained decision trees. Due to the interesting characteristics of using an ensemble model, such as fast training and testing, flexibility for high-dimensional data, robustness to noise, and expandability for parallel processing, the HF framework has received much attention in recent years. Generally, the HF uses an ensemble model of decision trees to generate a discriminative codebook using a patches-based method. The essential key idea for splitting data in internal nodes of the HF is the combination of using both information about appearance and geometry regarding patches. In more detail, trees in the HF are constructed, based on a set of a large number of patches ($P_i = \{I_i, c_i, d_i\}$) as shown in Fig. 1. Where I_i is the feature set (appearance) of a patch, c_i is the class label, which indicates whether this patch belongs to positive or negative training image, and d_i is the offset of the patch (a vector between the center of patch and the referent point). To make a decision about which patch is being directed to the left or right child node, the split function needs to be evaluated as follows:

$$l(P_i,\Theta) = \begin{cases} 0, & if\ P_i^k(\theta_1) - P_i^k(\theta_2) < \theta_3 \\ 1, & otherwise \end{cases} \quad (1)$$

where $l(,)$ is split function (test function) as shown in Fig. 2, Θ denotes the set of parameters including θ_1, θ_2, θ_3, k which are represented by two random points, random threshold, and index of channels in feature space, respectively. In order to decide which parameter set of the split function in the current node is most suitable for separating training data, the information gain needs to be maximized, defined as:

$$I_j = H(S_i) - \sum_{i \in \{L,R\}} \frac{|S_j^i|}{|S_j|} H(S_j^i) \quad (2)$$

where S_j is the set of training patches reaching to the j node, $|.|$ represents the number of patches in the current set, S_j^L and S_j^R are the sets of training data directed to left and right, respectively, and $H(S) = -\Sigma_{c \in C} p(c|S) log(p(c|S))$ is the Shannon entropy of set S. Here, $p(c|S)$ is a normalized empirical histogram of labels corresponding to the training point in S. The split function in the standard HF is very simple, in that it only considers the different intensities between two random points θ_1, θ_2 of the particular channel k in the feature space. Thus, it is sensitive to noise and illumination changes. Therefore, to overcome this limitation, the statistical information in feature space is used by randomly selecting five points θ_j with $j=1...5$. The intensity of these points $\overline{P}^k(\theta_j)$ is replaced by the mean of the neighborhoods rather than intensity of the points themselves in the standard HF. Then, the maximal intensity value of these points is chosen for comparing to the random threshold θ_6 to make the decision about directing this patch to the left or right child node. The entire process is illustrated in Fig. 2, and the split function in Eq. (1) can be rewritten as:

$$l(P_i,\Theta) = \begin{cases} 0, & if\ \arg\max_{j=1..5} (\overline{P}_i^k(\theta_j)) < \theta_6 \\ 1, & otherwise \end{cases} \quad (3)$$

where the set of parameters Θ in the current node consists of five points θ_j with $j=1...5$, the threshold value θ_6, and the index of channel k in feature space.

This process of choosing the novel split function mentioned above can generate a more discriminative codebook which shows robustness for noise as well as adaptability to illumination changes.

B. Error Optimization

In the standard HF, the decision tree is constructed in a

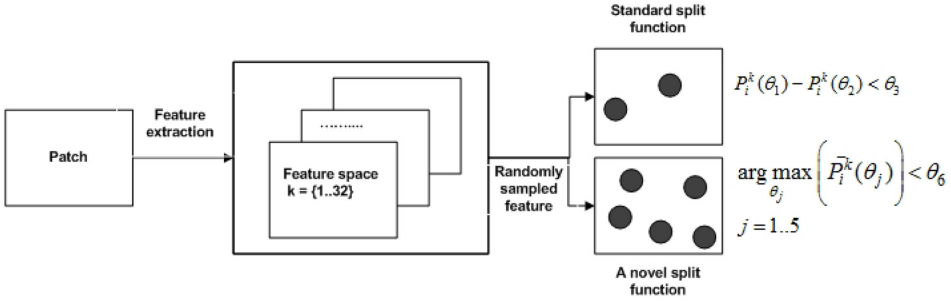

Fig. 2. Process for calculating standard split function and novel split function for each patch.

depth-first manner, meaning that the training data is first split in the left child node until the stopping criteria are reached, as illustrated in Fig. 3(a). Using this kind of construction, the error of the tree is calculated only after the entire tree is constructed. If the training data set is big, or the dimensions of the data are very high, this method is not an efficient way to compute errors. In contrast to the depth-first manner, building a tree in a breadth-first manner can efficiently handle error of tree by consecutively calculating the loss in each stage (see Fig. 3(b) for more details).

Adopting ideas from boosting algorithms [7] and alternating decision forests (ADF) [18], the errors of the training forest can be minimized using weights for all patches from both positive and negative images. These weights will be updated in each stage of the tree construction process by minimizing the global loss. Thus, the weights become higher for classifying harder patches and smaller with easier ones. This process is able to support trees to focus attention on harder samples rather than treating all samples equally as in standard HF. To do that we need to 1) perform tree construction in a breadth-first manner instead of depth-first, 2) update weights in each stage, and 3) change the estimation class probability to exploit the updated weights.

Constructing trees in a breadth-first manner has the advantages of being able to specify the stage of the forest corresponding to the depth of tree to estimate the global loss function [18] defined as:

$$\arg\min_{\theta^t} \sum_{i=1}^{N} l(c_i; F_{1:t-1}(I_i;\overline{\theta}) + f_t(I_i;\theta^t)) \tag{4}$$

where $l(.,.)$ the loss function; I_i, c_i the feature set and the label of patch. N is the number of training patches; $F_{1:t-1}(I_i;\overline{\theta}) = \Sigma_{j=1}^{t-1} v.f_j(I_j;\theta^j)$ is described as trained classifier, and $f_t(I;\theta^t)$ the classifier in the current iteration t; finally $\overline{\theta}$, θ^t are the set of parameters of the trained classifier and the parameters to be trained in the current iteration t, respectively.

After defining it, the global loss function can be used to calculate the boosting weights in the current stage. This weight ξ_i^t of the patch P_i in the iteration t can be defined as:

$$\xi_i^t = \left| \frac{\partial l(c_i, F_{1:t-1}(I_i;\overline{\theta}))}{\partial F(I)} \right| \tag{5}$$

There are several different boosting algorithms that can be differentiated by the loss function they use. In this work, the tangent loss function [20] was used to minimize the errors within stage during the training forest. Its formulation is defined as:

$$L_t(I) = (2arctan(F(I)) - 1)^2 \tag{6}$$

Using the tangent loss function, the weight update in Eq. (5) can be rewritten as:

$$\xi_i^t = \left| -\frac{4(2arctan(F(I_i, c_i)) - 1)}{1 + (F(I_i, c_i))^2} \right| \tag{7}$$

To take the weights into account in the split function, the estimation class probability needs to be changed as:

$$p(c|S) = \frac{\sum_{i=1}^{|S|} \mathbb{1}[y_i = c].\xi_i^t}{\sum_{i=1}^{|S|} \xi_i^t} \tag{8}$$

where $\mathbb{1}[y_i = c]$ is the indicator function which returns 1 if the label y_i of patch $P_i \in S$ is equal to c, and 0 otherwise.

By training the decision tree in the breadth-first manner, one can calculate errors gradually in each stage of tree construction, and adjust weights in a suitable way to minimize errors. This process is iterated until the depth of tree is reached or the number of patches in each node is smaller than some threshold value.

III. EXPERIMENTAL RESULTS

A. Data Preparation and Training Forest

In this work, the data used for training were taken from the INRIA database [12], which consists of 1,237 positive samples (human presence) and 3,891 negative samples (no human presence). Only 600 positive examples and 600 negative examples were selected to train the forest. Those examples were resized to a fixed size, so that the larger bounding box size including width or height was approximately 100 pixels, on average, over the data set.

In each example, 50 patches with fixed size (16 × 16) were sampled randomly to create the training data set. Thus, the number of positive patches was 30,000, and the number of negative patches was 30,000. Features used to describe patches are combinations of simple local fea-

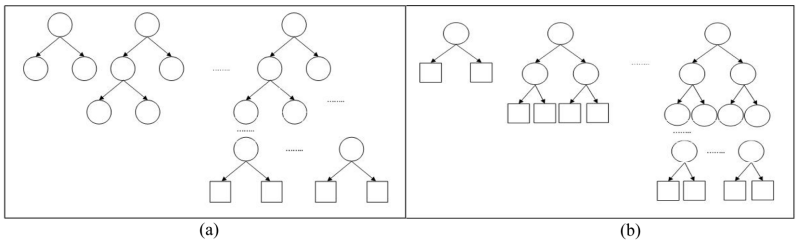

Fig. 3. Tree construction in (a) depth-first manner and (b) breadth-first manner.

tures, such as LAB color space, the first- and the second-order derivatives in the x and y directions of the gray image, and HOG features. For constructing the forest, 15 decision trees were used and the maximum depth for each tree was 15. To obtain the best separate data for each internal node, 2,000 split functions were generated randomly to find the maximal information gain (Section II). After finishing tree construction with the proposed method, 13 parameters were stored in internal nodes (versus 7 parameters in the standard HF), and the probability of foreground and a set of offsets were stored in the leaf nodes. All of this information is needed for detecting pedestrians in the scenes, and localizing the position by analyzing the object hypothesis [12]. The constructed forest is then used for comparing the performance of the proposed method with respect to the standard HF on some public data sets, including TUD-pedestrian, TUD-crossing, and TUD-campus.

B. Evaluation

The performance of the proposed approach was evaluated on three challenging public data sets: TUD-pedestrian, TUD-crossing, and TUD-campus [22]. The TUD-pedestrian data set contains 250 images with 311 side-view fully visible pedestrians with significant variation in clothing and articulation. The TUD-crossing sequence consists of 201 images with 1,008 annotated pedestrians, and the TUD-campus sequence contains 71 images of 303 annotated pedestrians.

For comparison with standard HF, precision-recall curves were generated. Precision and recall can be defined as:

$$precision = \frac{TP}{TP+FP} \tag{9}$$

$$recall = \frac{TP}{TP+FN} \tag{10}$$

where true positive (TP) is the correct detection, false positive (FP) is a false alarm, false negative (FN) is a miss-detection. In this paper, the PASCAL measure [23] was used to define whether the detected object is correct, false, or missed:

$$a_0 = \frac{area(BB_{dt} \cap BB_{gt})}{area(BB_{dt} \cup BB_{gt})} > 0.5 \tag{11}$$

where BB_{dt}, BB_{gt} are the detected bounding box and the ground truth box, respectively. According to Eq. (11), the bounding box is considered as a true positive detection if the overlap area of detection and the ground-truth bounding box exceeds 50%. Unmatched BB_{dt} is counted as a false positive and unmatched BB_{gt} as a false negative. The detection results of the proposed method are shown in Fig. 4(a) with TUD-pedestrian, Fig. 4(b) with TUD-crossing, and Fig. 4(c) with TUD-campus. Using a novel

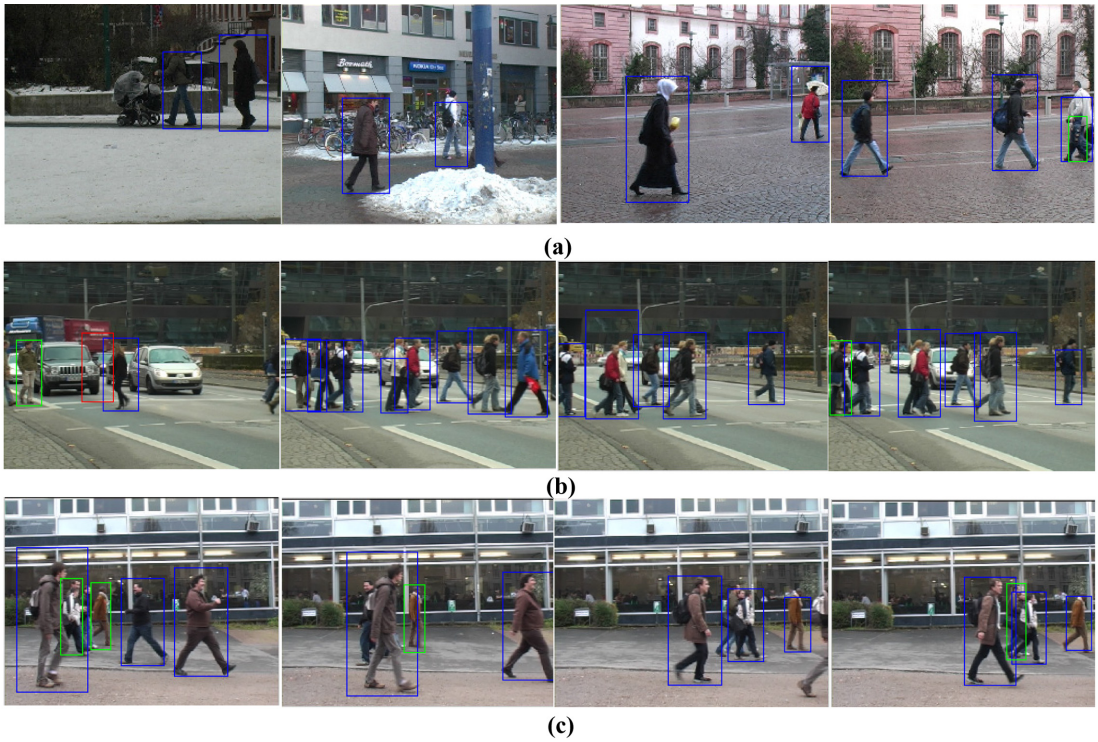

Fig. 4. The detection results evaluated on (a) TUD-pedestrian, (b) TUD-crossing-sequences, and (c) TUD-campus-sequences datasets (blue: true detection, green: missed detection, red: false alarm).

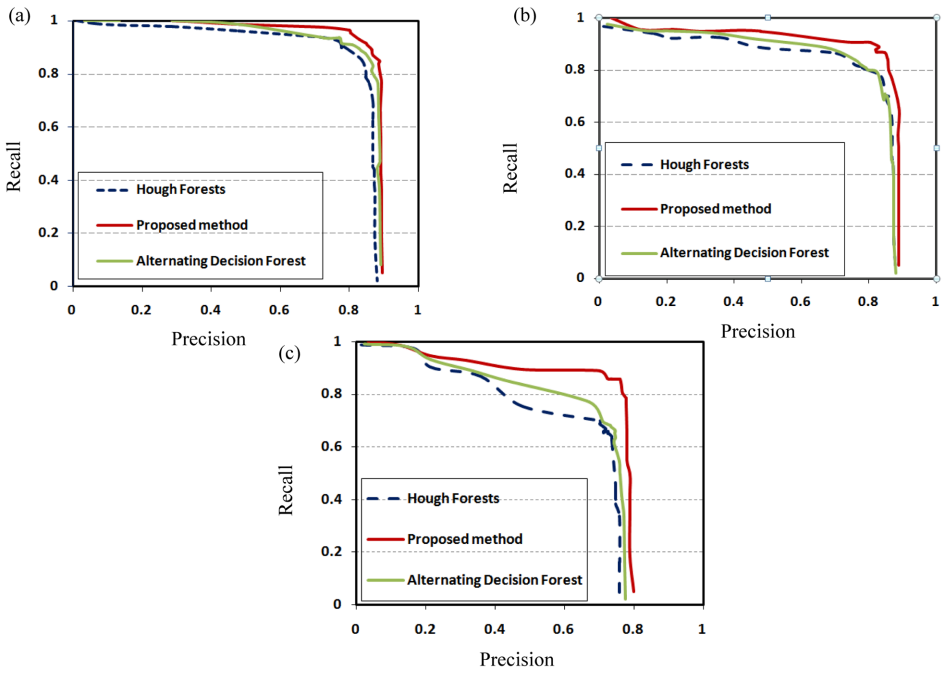

Fig. 5. Precision-recall curves generated for (a) TUD-pedestrian, (b) TUD-crossing-sequences, and (c) TUD-campus-sequences data sets.

Table 1. Comparison of AP and AUC with different methods on some public data sets

Datasets	HF		Proposed method		ADF	
	AP	AUC	AP	AUC	AP	AUC
TUD_1	0.863	0.851	**0.881**	**0.872**	0.869	0.847
TUD_2	0.741	0.784	**0.792**	**0.814**	0.766	0.780
TUD_3	0.586	0.648	**0.684**	**0.698**	0.621	0.654

TUD_1: TUD-pedestrian, TUD_2: TUD-crossing, TUD_3: TUD-campus, AP: average precision, AUC: area under curve, HF: Hough forest, ADF: alternating decision forest.

split function and the boosting idea from boosting algorithms, the individual pedestrians were detected successfully within the different conditions of background, such as weather changes, or illumination variations, even when the pedestrians appeared in a group as seen in Fig. 4(b) and (c). Precision-recall curves were generated for comparing the proposed method with standard HF and ADF (Fig. 5). The average precision (AP) and area under curve (AUC) [24] were calculated. A summary is shown in Table 1.

To evaluate the influence of the randomness point number Ω in a novel split function on the randomness sample number Γ generated in each node, statistical information was calculated for the TUD-pedestrian data set. Experimental values of average precision are shown in Table 2. With 2,000 randomness numbers generated in each node, the proposed method (novel split function [NFS]) gave the best performance (the highest AP = 0.881) when $\Omega = 5$. This value can be compared with result of standard HF (AP = 0.863) and ADF (AP = 0.869). However, with the increased number of randomness points in the novel split function, the system will require more processing time in both the training and testing phases compared with standard HF and ADF.

Table 2. Comparison of AP with respect to different methods for TUD-pedestrian dataset

Method	$\Gamma = 500$	$\Gamma = 1000$	$\Gamma = 2000$	$\Gamma = 4000$
HF	0.842	0.854	**0.863**	0.842
ADF	0.848	0.862	**0.869**	0.857
NSF ($\Omega = 2$)	0.845	0.840	0.871	0.853
NSF ($\Omega = 3$)	0.853	0.858	0.874	0.858
NSF ($\Omega = 4$)	0.865	0.857	0.877	0.841
NSF ($\Omega = 5$)	0.849	0.843	**0.881**	0.849
NSF ($\Omega = 6$)	0.841	0.838	0.873	0.852
NSF ($\Omega = 7$)	0.831	0.833	0.871	0.851

AP: average precision, HF: Hough forest, ADF: alternating decision forest, NSF: novel split function.

IV. CONCLUSIONS AND FUTURE WORK

Using a novel split function to exploit the statistical information of features, the constructed forest is more

robust to noise and illumination changes. Moreover, the idea from boosting is used to minimize errors gradually during the training process; thus, the forest can focus attention on hard samples. However, the proposed method fails to detect a pedestrian when the background consists of many different objects, such as cars, trees, and buildings. This limitation can be solved by combining with motion information such as the background modeling, in video sequences. The proposed method can be applied to train for other object types, such as vehicles and animals as a direction for future work.

AKNOWLEDGMENTS

This research was funded by the Ministry of Science, ICT, & Future Planning (MSIP), Korea, in the ICT R&D Program 2014.

REFERENCES

1. I. Cohen, and G. Medioni, "Detecting and tracking moving objects for video surveillance," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Fort Collins, CO, 1999, pp. 2319-2325.
2. A. Ess, K. Schindler, B. Leibe, and L. Van Gool, "Object detection and tracking for autonomous navigation in dynamic environments," *International Journal of Robotics Research ,* vol. 29, no. 14, pp. 1707-1725, 2010.
3. Y. Kong, Y. Jia, and Y. Fu, "Learning human interaction by interactive phrases," in *Proceedings of the 12th European Conference on Computer Vision*, Florence, Italy, 2012, pp. 300-313.
4. V. N. Vapnik, *Statistical Learning Theory*, New York: Wiley, pp. 493-520, 1998.
5. R. Rojas, *Neural Networks: A Systematic Introduction*, Berlin: Springer, 1996.
6. Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119-139, 1997.
7. R. E. Schapire, "The boosting approach to machine learning: an overview," in *Nonlinear Estimation and Classification*, New York: Springer, pp. 149-171, 2003.
8. L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5-32, 2001.
9. A. Criminisi and J. Shotton, *Decision Forests for Computer Vision and Medical Image Analysis*, New York: Springer, pp. 25-46, 2013.
10. D. Tang, Y. Liu, and T. K. Kim, "Fast pedestrian detection by cascaded random forest with dominant orientation template," in *Proceedings of the British Machine Vision Conference (BMVC2012)*, Surrey, UK, 2012, pp. 1-11.
11. S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab, "Dominant orientation templates for real-time detection of texture-less objects," in *Proceedings of the 23rd IEEE Conference on Computer Vision and Pattern Recognition (CVPR2010)*, San Francisco, CA, 2010, pp. 2257-2264.
12. N. Dalal and B. Triggs, "Histogram of oriented gradients for human detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2005)*, San Diego, CA, 2005, pp. 886-893.
13. K. Murphy, A. Torralba, D. Eaton, and W. Freeman, "Object detection and localization using local and global features," in *Toward Category-Level Object Recognition*, New York: Springer, pp. 382-400, 2006.
14. B. Leibe, E. Seemann, and B. Schiele, "Pedestrian detection in crowded scenes," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2005)*, San Diego, CA, 2005, pp. 878-885.
15. A. Bosch, A. Zisserman, and X. Muoz, "Image classification using random forests and ferns," in *Proceedings of the 11th International Conference on Computer Vision (ICCV2007)*, Rio de Janeiro, Brazil, 2007, pp. 1-8.
16. B. Xu, Y. Ye, and L. Nie, "An improved random forest classifier for image classification," in *Proceedings of the International Conference on Information and Automation (ICIA2012)*, Shenyang, China, 2012, pp. 795-800.
17. J. Gall and V. Lempitsky, "Class-specific Hough forests for object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2009),* Miami, FL, 2009, pp. 1022-1029.
18. S. Schulter, P. Wohlhart, C. Leistner, A. Saffari, P. M. Roth, and H. Bischof, "Alternating decision forests," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2013)*, Portland, OR, 2013, pp. 508-515.
19. P. Wohlhart, S. Schulter, M. Kostinger, P. M. Roth, and H. Bischof, "Discriminative Hough forests for object detection," in *Proceedings of the British Machine Vision Conference (BMVC2012)*, Surrey, UK, 2012, pp. 1-11.
20. H. Masnadi-Shirazi, V. Mahadevan, and N. Vasconcelos, "On the design of robust classifiers for computer vision," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2010)*, San Francisco, CA, 2010, pp. 779-786.
21. M. Kobetski and J. Sullivan, "Improved boosting performance by exclusion of ambiguous positive examples," in *Proceedings of the 2nd International Conference on Pattern Recognition Applications and Methods (ICPRAM2013)*, Barcelona, Spain, 2013, pp. 11-21.
22. M. Andriluka, S. Roth, and B. Schiele, "People-tracking-by-detection and people-detection-by-tracking," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR2008),* Anchorage, AK, 2008, pp. 1-8.
23. P. Dollar, C. Wojek, B. Schiele, and P. Perona, "Pedestrian detection: an evaluation of the state of the art on pattern analysis and machine intelligent," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 743-761, 2012.
24. M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (VOC) challenge," *International Journal of Computer Vision*, vol. 88, no. 2, pp. 303-338, 2010.

Trung Dung Do

Trung Dung Do is a graduate student in the doctoral program of the School of Information and Communication Engineering, Inha University, Korea. He received his bachelor's degree in Computer Technology from National Research Irkutsk State Technical University, Russia, in 2010, and a master's degree from the Graduate School of Information and Communication Engineering, Inha University, in 2014. His research interests include computer vision, pattern recognition, image processing, and machine learning.

Thi Ly Vu

Thi Ly Vu completed her B.S. in Department of Information Technology from Le Quy Don Technical University, Vietnam, in 2011. She is currently pursuing the M.S. degree in the Information Intelligent Processing System Lab. of the Graduate School of Information & Communication Engineering at Inha University, Korea. Her studies include pattern recognition, image processing, and intelligent processing system.

Van Huan Nguyen

Van Huan Nguyen completed his B.S. in the Department of Applied Mathematics and Informatics from Hanoi University of Science and Technology, Vietnam, in 2005, and M.S. and Ph.D. degrees in the Computer Vision Lab. of the Graduate School of Information & Communication Engineering at Inha University, in 2008 and 2012, respectively. He is currently working as a postdoctoral researcher in the Super Intelligence Technology Center, Inha University. His research interests include biometrics, pattern recognition, remote sensing, video processing, and video surveillance systems.

Hakil Kim

Hakil Kim is a professor at School of Information and Communication Engineering, Inha University, Korea. He received the B.S. degree in Control and Instrumentation Engineering from Seoul National University, Korea, in 1983, and the M.S. and Ph.D. degrees in Electrical and Computer Engineering from Purdue University, West Lafayette, IN, USA, in 1985 and 1990, respectively. His research areas include computer vision and pattern recognition. He has been actively participating in IOS/IEC JTC1-SC37 and ITU-T/SG17 WP2/Q.8 Telebiometrics as a project editor and a rapporteur.

Chongho Lee

Chongho Lee received his M.S. degree in Electrical Engineering from Seoul National University, Korea and Ph.D. degree from Iowa State University, USA, in the Department of Computer Engineering. His research areas are artificial neural networks and intelligent systems. He is currently a Professor in School of Information & Communication Engineering at Inha University, Incheon. He joined in Dynamic Partial Reconfigurable FIR filter design, LNCS 3839, Mar. 2006, Design of CSVM Processor for Intelligence Expression, Journal of Electrical and Electronic Material, Feb. 2007 and DNA-inspired CVD Diagnostic Hardware Architecture, Journal of KIEE, Feb 2008.