A Sparse-Dense HOG Window Sampling Technique for Fast Pedestrian Detection in Aerial Images

Ranjeet Kumar(✉) and Alok Kanti Deb

Indian Institute of Technology, Kharagpur, India
ranjeetkumar16722@gmail.com, alokkanti@ee.iitkgp.ac.in

Abstract. Pedestrian detection from Unmanned Aerial Vehicle (UAV) has been an important part of surveillance systems. A Two-stage (Sparse-Dense) sliding window technique has been proposed to increase the speed of pedestrian detection using HOG-SVM classifier. Standard techniques follow a sliding window approach with a fixed sliding strides over a multi-resolution image pyramid for detection. The presented technique breaks down the detection task into sparse sampling and dense sampling stages where the first one is region proposal step and second stage scans only the proposed regions for objects. Sparse sampling stage is working as weak classifier whereas the dense sampling stage works as strong classifier for an image patch. Average pedestrian detection speed using the proposed technique gave improvement from 1.95 fps to 15.36 fps for input images of dimension [640, 360] on a system with 3.2 GHz CPU. UAV123 [1] dataset has been chosen to train the classifier. For detection, Average Center Prediction Error has been taken to quantify detection performance with increased speed.

Keywords: Sparse-dense sampling detector · UAV123 · Region proposal · Sparse sampling · Dense sampling · Pedestrian detection · HOG · SVM

1 Introduction

Fast pedestrian detection on aerial images has been a challenge due to dynamic nature of the images and hardware constraints. Integral Image for fast feature calculation [2] and Histogram of Oriented Gradients (HOG) [3], classification by Support Vector Machine (SVM) based tree-type neural network [4] are some initial work. Some improved methods for feature extraction are Integral channel feature [5], Boosted HOG [6], Channel Feature Extrapolation [7] and Search Region Proposal based on Saliency Map [8]. Pedestrian detection in infrared images has been shown by Zhang et al. [9]. Some techniques exploiting input image properties are Image Orientation Adjustment by Xu et al. [10] and Locally constraint linear coding based detection by Yang et al. [11]. Some techniques trying to enhance detection speed by breaking the task into multiple stages include local binary pattern with HOG-SVM classifier [12], simplified HOG [13], Center Symmetric - Local Binary Pattern (XCS-LBP) [14], Bin-Interleaved HOG [15] and two-stage linear with non-linear SVM [16] but they need improvement for real-time application. Some hardware solutions for fast HOG-SVM based classification have been presented in [17, 18] that proposed hardware design suitable for HOG-SVM classification. This work proposes a Two-stage (Sparse-Dense) sliding window technique for pedestrian detection task and is an improvement over standard single stage sliding window techniques used with HOG+SVM [3, 5] based classifiers. Re-searchers have proposed using features other than HOG too for better detection but the proposed work shows how detection process can be modified to speed it up using existing classification method HOG-SVM and achieve real-time or near real-time performance. Section 2 discusses about standard sliding window based pedestrian detection. Section 3 presents the proposed two-stage (sparse-dense sliding) window based detection technique. Section 4 discusses about experimental setup, results and analysis. Section 5 concludes the work presented.

2 HOG-SVM Classification Based Pedestrian Detection

Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) was given by Dalal et al. [3] to extract visual information from an image patch using pixel gradients. The technique has been used widely for classification/detection [5–9]. One can refer to [3] for HOG feature vector calculation for an image. Parameters to calculate HOG feature descriptor for an image patch of dimension $[M, N]$ has been shown in Table 1 and its length F_l [3] using (1) is 3780.

Table 1. HOG feature descriptor parameters

Parameter	Value
Window size $[W_h, W_v]$	[64, 128] pixels
Cell size $[C_h, C_v]$	[8, 8] pixels
Block size $[B_h, B_v]$	[2, 2] cells
Gamma correction (γ)	0.5
Bin size (b)	9

$$F_l = \left(\frac{M}{C_h} - 1\right) * \left(\frac{N}{C_v} - 1\right) * b * B_h * B_v \tag{1}$$

HOG feature plots with corresponding RGB images has been shown in Fig. 1.

SVM Classification

Support Vector Machines (SVM) [19, 20] is a supervised learning based classification algorithm that creates an N-dimensional hyper-plane that divides m number of classes. Input feature vector (p dimensional) and output label for a sample i has been denoted by

$h_i \in R^p$ and $y_i \in \{-1, 1\}$, where $i = 1, 2, 3, ..., n$, Person and Background classes have been represented by label 1 and -1 respectively.

$$f(h_i) = w^T * \phi(h_i) + b \tag{2}$$

Here, $f(h_i)$ is distance of a sample from decision boundary and its sign indicate predicted class, w is weight vector, $\phi(h_i)$ is a function (kernel) of h_i and b is the offset from decision boundary.

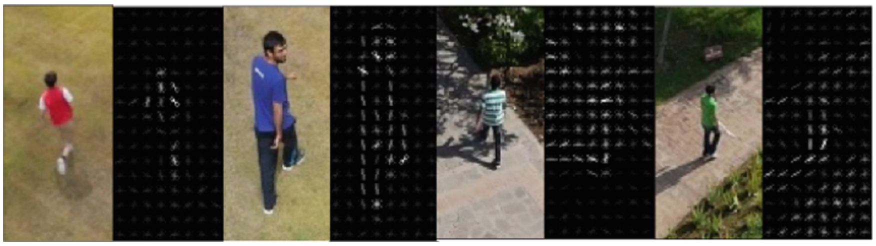

Fig. 1. Example images from UAV123 [1] dataset with their HOG feature

Detection in an Image

While HOG-SVM classifier classify an image patch of dimension $[W_h, W_v]$, pedestrian detection on full image is done by extracting image patches of dimension $[W_h, W_v]$ in sliding window manner from multi-scale image pyramid.. Five level image pyramid formed by original and downscaled versions of an image has been shown in Fig. 2.

An important factor that determines the speed of detection is frequency of classification step for an image that has been denoted by HOG^I_{count} and can be obtained using (3), where W_h and W_v are dimensions of HOG window, DS_h and DS_v are dense sliding strides for HOG window $[W_h, W_v]$ in horizontal and vertical direction respectively. $[M, N]$ is input image width and height in pixels respectively.

$$HOG^I_{count} = \frac{M - W_h}{DS_h} * \frac{N - W_v}{DS_v} \tag{3}$$

Down-scaled image shape can be given by $shape_{dn} = \left[\frac{M}{\alpha^l}, \frac{N}{\alpha^l}\right]$ where $\alpha = 1.5$ and $l \in \{0, 1, ..., (L-1)\}$ are downscaling factor and image pyramid level respectively. $L = 5$ is the number of levels in image pyramid. Classification step frequency for an image pyramid (HOG^P_{count}) can be obtained using (4).

$$HOG^P_{count} = \sum\nolimits_{l=0}^{L-1} \left(\frac{M - W_h}{\alpha^l * DS_h} * \frac{N - W_v}{\alpha^l * DS_v} \right) \tag{4}$$

where, HOG^P_{count} denoted number of HOG-SVM classification step for an image pyramid with L levels. Value of HOG^P_{count} for the parameters given in Table 2 using (4) is 3692. This work focuses on reducing the required number of classification steps for an image pyramid by introducing a two-stage sliding (sparse-dense sampling) window technique.

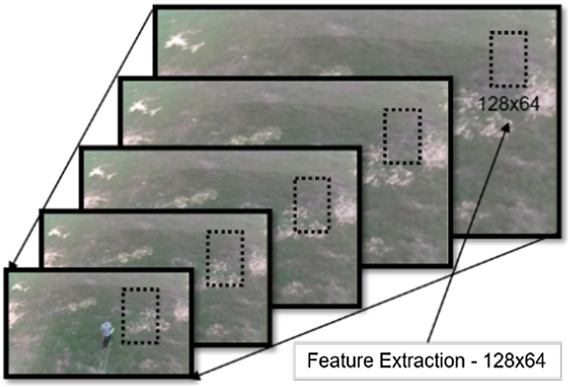

Fig. 2. Multi-resolution image pyramid formed from downscaled versions of original image

Table 2. Input image shape, HOG window shape, sliding stride for standard detection technique [3] and image pyramid parameters

Parameter	Value
Input image shape ($[M, N]$)	[640, 360]
HOG window shape ($[W\_h, W\_v]$)	[64, 128]
Image pyramid levels (L)	5
Downscaling factor (α)	1.5
Dense sliding stride ($[DS\_h, DS\_v]$)	[8, 8]

3 A Two-Stage Sliding Window

Conventional classifier based detection techniques follow a dense sampling approach to classify patches from image pyramid into number of classes. [3, 5, 9]. Dense sampling window stride has been denoted by $[DS_h, DS_v]$ and is [8, 8]. Here, DS_h and DS_v are strides in horizontal and vertical direction respectively. The proposed two-stage sliding window technique divides the detection task into sparse and dense sampling stages. Block diagram has been shown in Fig. 3.

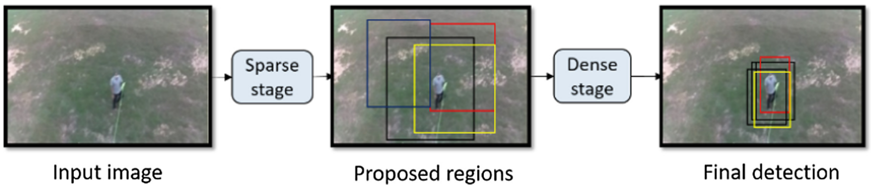

Fig. 3. Block diagram of Two-Stage sliding window pedestrian detection

Stage 1: Sparse Sampling
In first stage, image patches of size $[W_h, W_v]$ are extracted from all the levels in the image pyramid at larger window sliding strides denoted by $[SS_h, SS_v]$ as compared to $[DS_h, DS_v]$ taken by most of the previous techniques [3, 5, 6, 10, 11]. Here, $[SS_h, SS_v] \in \{[32, 64], [43, 90], [51, 102]\}$. HOG-SVM classifier output represents distance from SVM decision boundary and has been taken to determine confidence of classification. Distance threshold for this stage has been denoted by Th_{sparse} and image patches exceeding Th_{sparse} are recorded as regions for proposal to stage 2. The regions proposed in the stage are of dimension $[W_h, W_v]$ in their corresponding downscaled image from the pyramid. The shape of proposed regions has been transformed back to level 0 (to represent same region in original image) denoted by PRS_0 and can be obtained using (5) where $l \in \{0, 1, ..., (L-1)\}$.

$$PRS_0 = [W_h, W_v] * \alpha^l \tag{5}$$

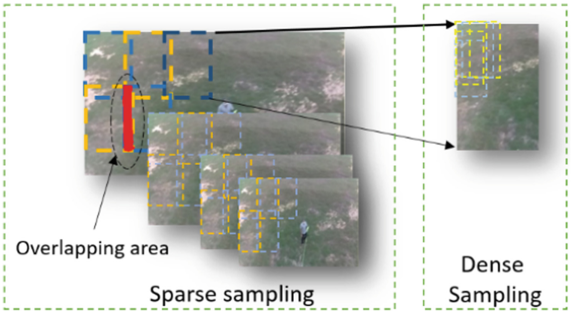

Fig. 4. Sparse sampling for region proposal and dense sampling on proposed region

Window sliding Stride fir sparse sampling stage has been calculated by taking percentage overlap between consecutive sampling windows. Overlap percentage of 50%, 30% and 20% have been taken for experimentation. Window sliding Stride values $[SS_h, SS_v]$ in pixels can be obtained using (6) and are [32, 64], [43, 90] and [51, 102] respectively. HOG-SVM classification step frequency in sparse sampling stage has been demoted by HOG^S_{count} and can be obtained using (7).

$$\left[SS_h, SS_v\right] = overlap\% * \left[W_h, W_v\right] \tag{6}$$

$$HOG^S_{count} = \sum\nolimits_{l=0}^{L-1} \left(\frac{M - W_h}{\alpha^l * SS_h} * \frac{N - W_v}{\alpha^l * SS_v} \right) \tag{7}$$

Stage 2: Dense Sampling
In this stage, image regions proposed from first stage are searched for objects by HOG-SVM classifier with dense sampling window strides $[DS_h, DS_v]$. Image patches crossing a threshold Th_{dense} are final detections. An example has been shown in Fig. 4. Only the

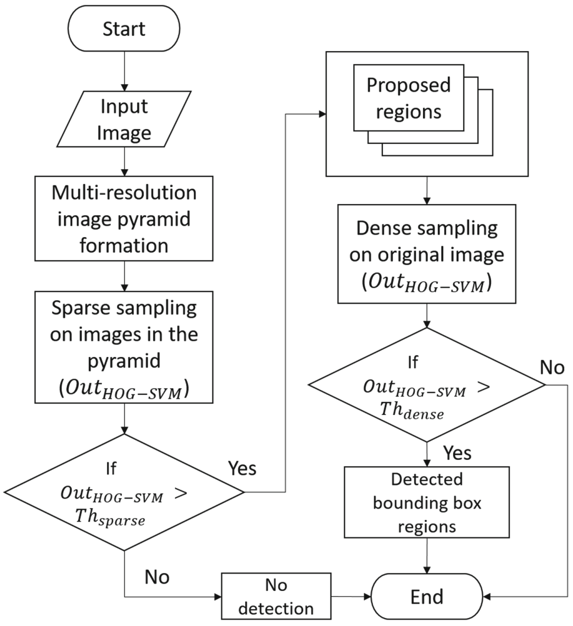
Start
Input Image
Multi-resolution image pyramid formation
Sparse sampling on images in the pyramid ($Out_{HOG-SVM}$)
If $Out_{HOG-SVM} > Th_{sparse}$
Yes
No
No detection
Proposed regions
Dense sampling on original image ($Out_{HOG-SVM}$)
If $Out_{HOG-SVM} > Th_{dense}$
No
Yes
Detected bounding box regions
End

Fig. 5. Two-stage sliding window pedestrian detector process flow

proposed regions from stage 1 are processed in this stage and not the whole image pyramid. This saves significant processing time. Flow-chart for the two-stage process has been shown in Fig. 5.

Classification step frequency for dense sampling stage has been denoted by HOG^I_{count} (3) and depends upon region proposal. Here, I in HOG^I_{count} represent an image region proposed by sparse sampling stage. $\left(HOG^S_{count} + HOG^I_{count}\right)$ is combined classification step frequency and has been determined by average value for 1000 pedestrian images from UAV123 [1] dataset.

Suppressing Duplicate Detections

Dense sampling stage yields multiple detections around the object as classifier output crosses Th_{dense}. These duplicate detections have been suppressed by computing Intersection-over-Union (IoU) among the detection boxes using (8) where R_{B_1} and R_{B_2} denote area of box B_1 and B_2 respectively. If $IoU(B_1, B_2)$ crosses a threshold IoU_{th}, then the box with lower confidence ($f(h_i)$(2)) is discarded. An example can be seen in Fig. 6.

$$IoU(B_1, B_2) = \frac{R_{B_1} \cap R_{B_2}}{R_{B_1} \cup R_{B_2}} \tag{8}$$

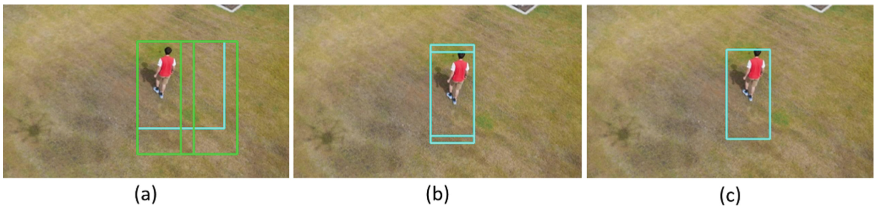

Fig. 6. (a) Sparse sampling stage output, (b) Dense sampling stage output, (c) *IoU* thresholded output

4 Experimental Results and Analysis

Dataset

Dataset used for training and testing of HOG-SVM classifier is UAV123 [1]. It has 11575 images in its person class, out of which 2620 images with pedestrians are randomly selected and a window of [64, 128] size has been cropped to form person and 2450 windows cropped for background class. Positive and negative class formation for training has been shown in Fig. 7.

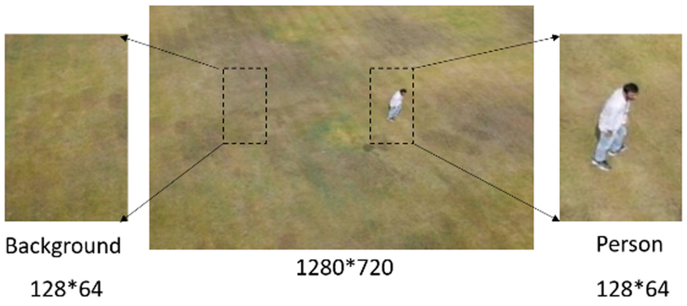

Fig. 7. Person and background dataset creation form UAV123 [1] for training and testing of SVM classifier

SVM Training

Dataset size is 5070 images (2620 person class and 2450 background class) with train-test split ratio 80:20. 5-fold cross validation scheme has been adopted to split dataset into 5 mutually exclusive parts. Training has been done on 4 parts combined and testing on the remaining part. 5 trials of training/testing has been done. Scikit-learn [21] python library has been used to train Support Vector Classifier (SVC) for binary classification of Person and Background classes. Classifier parameters have been shown in Table 3.

Classification Performance

Precision, Recall, F1-score and Accuracy have been taken as classification performance metrics [22]. Metrics for 5 trials for test data with mean values has been presented in Table 4.

Table 3. SVM parameters

Parameter	Value
Kernel	Linear
Train/Test Image window size	128×64
Person class data size	2620
Background class data size	2450
Train: Test split	80:20
Iteration	5000

Table 4. Classification performance (5-fold cross validation, SD - Standard Deviation)

Dataset	Class	Precision	Recall	F1-score	Accuracy
Set1	Person	0.9943	0.9943	0.9943	0.9941
	Background	0.9939	0.9939	0.9939	0.9941
Set2	Person	1.0000	0.9356	0.9667	0.9666
	Background	0.9351	1.0000	0.9665	0.9666
Set3	Person	1.0000	0.9261	0.9616	0.9617
	Background	0.9263	1.0000	0.9617	0.9617
Set4	Person	1.0000	0.8220	0.9023	0.9077
	Background	0.8390	1.0000	0.9125	0.9077
Set5	Person	0.9715	0.8386	0.9002	0.9314
	Background	0.9127	0.9856	0.9478	0.9314
Mean ± SD	**Person**	0.9892 ± 0.019	0.9033 ± 0.064	0.9450 ± 0.038	0.9523 ± 0.03
	Background	0.9214 ± 0.05	0.9959 ± 0.006	0.9565 ± 0.027	0.9523 ± 0.03

Detection Performance

UAV123 [1] provides ground-truth bounding boxes for evaluation of detection result. The prediction bounding box dimension of HOG-SVM classifier is fixed to [64, 128] as the classifier is not designed for bounding box regression task. Centre Prediction Error (CPE) [23] has been taken as performance metric that measures difference between predicted object centre and ground-truth box centre. $CE_{avg}^{S_i}$ denote average CPE for an image sequence S_i and can be obtained using (9) where N_{S_i} denote number of frames in a sequence S_i, $\left(x^G, y^G\right)$ and $\left(x^P, y^P\right)$ represent ground-truth box centre and predicted box centre respectively. Chosen dataset has 23 person class image sequences named from **person1** to **person23** captured from an UAV platform. Average of $CPE_{avg}^{S_i}$ for 23 sequences and shown in (10) where $i = \{1, 2, 3, ..., K\}$ and $K = 23$.

$$CE_{avg}^{S_i} = \frac{1}{N_{S_i}} \sum\nolimits_{t=1}^{N_{S_i}} \sqrt{\left(x_t^G - x_t^P\right)^2 + \left(y_t^G - y_t^P\right)} \tag{9}$$

$$CE_{avg} = \frac{1}{K} \sum\nolimits_{i=1}^{K} CE_{avg}^{N_{S_i}} \tag{10}$$

$$CE_{avg}(\%) = \frac{CE_{avg} \times 100}{\sqrt{M^2 + N^2}} \tag{11}$$

CE_{avg} as percentage of input image diagonal length can be given by (11) to present a metric invariant to input image dimension. $CE_{avg}(\%)$ for person sequences in UAV123 [1] has been found out by running detector for all the 23 sequences and comes out to be 1.47%. Centre coordinate plot for X and Y direction for person1 has been shown in Fig. 8. It can clearly be seen from X and Y coordinate detection graphs that detector is following the ground-truth coordinates almost all the time. An example image with ground-truth box, detection box and a line joining their centers has been shown in Fig. 9.

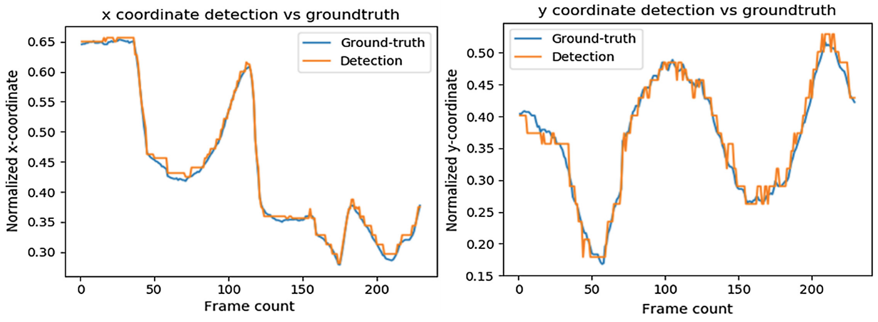

Fig. 8. x and y coordinate detection vs ground-truth

Proposed Technique Evaluation

In the proposed technique, $Th_{sparse} = 0.1$ has been taken for sparse sampling, $Th_{dense} = 1.0$ for dense sampling and $IoU_{th} = 0.5$ for IoU thresholding step. Improvement in processing time has been shown in Table 5 for different percentage of overlap between consecutive sampling windows in sparse sampling stage. Testing has been done on a 3.2 GHz CPU machine. Significant reduction in processing time can be seen as originally HOG-SVM classification step frequency has decreased from 3692 to 469, 344 and 294 (Table 5).

As shown in Fig. 10, the technique gives different region proposals for different sliding strides and detections are concentrated around object after dense sampling stage. It can be observed that maximum detections are there in single stage dense sampling (Fig. 10-b) method but most of them are redundant and will be removed after IoU thresholding. Sparse sampling with [32, 64] (Fig. 10-c)and [43, 90] (Fig. 10-f) strides

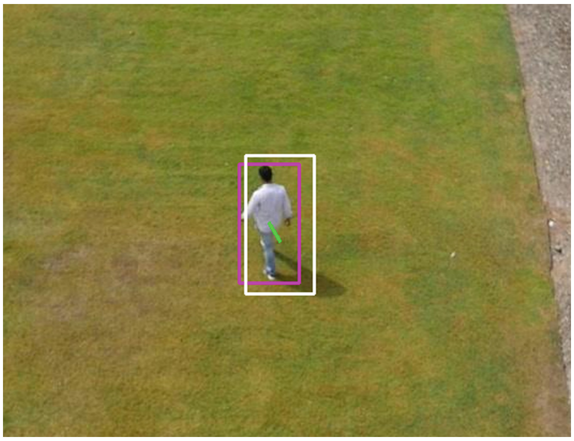

Fig. 9. Ground-truth box (pink), detection (white) and line between centers (green)

Table 5. Percentage overlap vs processing time: 2 stage detector (input size: [640, 360])

% overlap	Slide stride	Classification steps per image	Speed (fps)
[84.4, 92.2] (standard/baseline dense sampling detection)	[8, 8]	3692	1.95
[50.0, 50.0]	[32, 64]	469	15.36
[30.0, 30.0]	[43, 90]	344	20.92
[20.0, 20.0]	[51, 102]	294	24.48

provide significant number of region proposals around object and multiple detections after dense sampling stage (Fig. 10-d and 10-g). IoU thresholding has been applied to remove duplicate detections.

Comparison with single stage dense sampling technique has been shown in Table 5. The improvement in speed for input image of size [640, 360] with $[SS_h, SS_v] = [32, 64]$ is from 1.95 to 15.36 fps (improved by a factor of 7.88) and 1.95 to 24.48 fps for $[SS_h, SS_v] = [51, 102]$ (improved by a factor of 10.50).

Detection speed on full images has been compared with existing techniques and presented in Tables 6, 7, 8, 9 and 10 for different input image dimension.

It is evident from the comparison tables that the proposed technique performs better than similar techniques. Moreover, the detection quality of the technique has been quantified in terms of average center prediction error (CPE). It is a standard metric used to judge the distance of predicted bounding box to that of ground-truth. Average CPE $CE_{avg}(\%)$ for UAV123 dataset (person class) is only 1.47% of image diagonal length. Also, it should be noted that using classification based techniques clubbed with window sliding always introduces a quantization error in detection (evident in Fig. 9) which is equal to 0.5 times HOG window sliding stride $[DS_h, DS_v]$. So, a trade-off has to be maintained between detection speed and quality to choose a particular sliding stride.

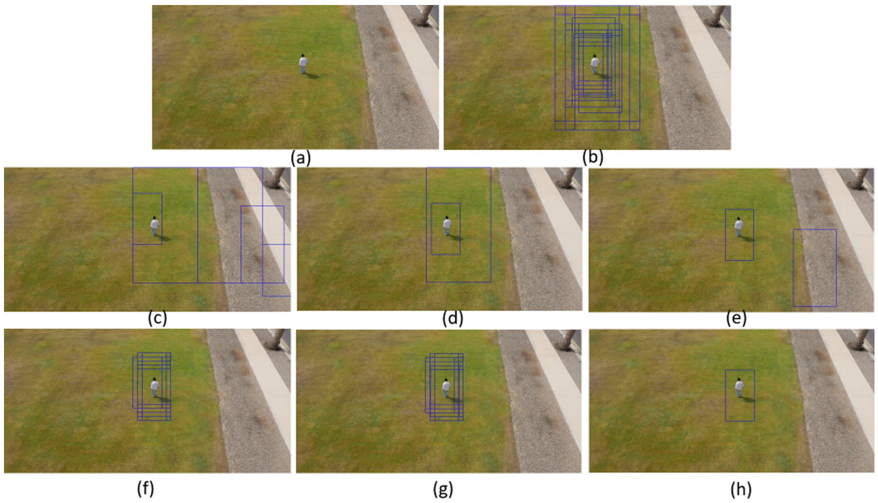

Fig. 10. (a) Original image (b) Standard dense sampling detection output on image pyramid, (c) Sparse stage output - [32, 64], (d) Sparse sampling output - [43, 90], (e) Sparse sampling output - [52, 102], (f) Dense sampling output for **c**, (g) Dense sampling output for **d**, (h) Dense sampling output for **e**

Table 6. Comparison with existing work (input image size: [320, 240])

Author	Technique	Detection speed (fps)
Dalal and Triggs [3] (2005)	Original HOG	1.07
Son et al. [15] (2010)	Bi-HOG	2.12
Sheng et al. [13] (2012)	Simplified HOG	3.33
Proposed	**Sparse-dense sampling**	**24.4**

Table 7. Comparison with existing work (input image size: [640, 480])

Author	Technique	Detection speed (fps)
P Dollar et al. [5] (2009)	Integral channel feature	0.5
Min et al. [16] (2013)	Two-stage linear+non-linear SVM	3.33
Vasuki et al. [14] (2016)	XCS-LBP with HOG-linear-SVM	4.05
Proposed	**Sparse-dense sampling**	**8.61**

Table 8. Comparison with existing work (input image size: [720, 400])

Author	Technique	Detection speed (fps)
Cao et al. [6] (2011)	Boosting HOG	4.76
Proposed	**Sparse-dense sampling**	**10.14**

Table 9. Comparison with existing work (input image size: [1200, 400])

Author	Technique	Detection speed (fps)
Xu et al. [10] (2017)	Orientation adjustment	5.3
Proposed	**Sparse-dense sampling**	**8.65**

Table 10. Comparison with existing work (input image size: [640, 320])

Author	Technique	Detection speed (fps)
Yang et al. [11] (2019)	Locally constraint linear coding	9
Proposed	**Sparse-dense sampling**	**16.8**

This work has used [8, 8] as sliding stride in dense sampling stage and is a standard used by other researchers. The presented average CPE value includes this quantization error inherently along with the actual detection error.

5 Conclusion

The work introduced a Two-stage (Sparse-Dense) sliding window sampling technique for fast pedestrian detector. The first stage was sparse sampling stage to extract relevant regions. In the second stage, the proposed regions were taken to run classifier with smaller strides and larger classification threshold. Thus, a modified version of simple HOG-SVM detector has been presented. Visual information in image was exploited for region proposal and most of the time was spent in processing proposed regions. The proposed technique can be utilized to run real-time detection on low-cost processor on UAV platform and thus eliminate dependency on external system. This eventually opens up scope of more applications using UAV systems.

Acknowledgement. We kindly acknowledge IMPRINT I project, MHRD, Govt. of India for supporting with resources from the project "Decentralized target tracking using swarm of aerial robots".

References

1. Mueller, M., Smith, N., Ghanem, B.: A benchmark and simulator for UAV tracking. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9905, pp. 445–461. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46448-0_27
2. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1 (2001)

3. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 1, pp. 886–893 (2005)
4. Jayadeva, Deb, A., Chandra, S.: Binary classification by SVM based tree type neural networks. In: Proceedings of the 2002 International Joint Conference on Neural Networks, vol. 3, pp. 2773–2778 (2002)
5. Dollár, P., Tu, Z., Perona, P., Belongie, S.: Integral channel features. In: The British Machine Vision Conference, pp. 91.1–91.11 (2009)
6. Cao, X., Wu, C., Yan, P., Li, X.: Linear SVM classification using boosting hog features for vehicle detection in low-altitude airborne videos. In: IEEE Conference on Image Processing, pp. 2421–2424 (2011)
7. Dollar, P., Appel, R., Belongie, S., Perona, P.: Fast feature pyramids for object detection. IEEE Trans. Pattern Anal. Mach. Intell. **36**(8), 1532–1545 (2014)
8. Blondel, P., Potelle, A., Pegard, C., Lozano, R.: Human detection in uncluttered environments: from ground to UAV view. In: 13th International Conference on Control Automation Robotics and Vision, pp. 76–81 (2014)
9. Zhang, L., Wu, B., Nevatia, R.: Pedestrian detection in infrared images based on local shape features. In: IEEE Conference on Computer Vision and Pattern Recognition, pp. 1–8 (2007)
10. Xu, Y., Yu, G., Wu, X., Wang, Y., Ma, Y.: An enhanced Viola-Jones vehicle detection method from unmanned aerial vehicles imagery. IEEE Trans. Intell. Transp. Syst. **18**(7), 1845–1856 (2017)
11. Yang, Z., Huang, Z., Yang, Y., Yang, F., Yin, Z.: Accurate specified-pedestrian tracking from unmanned aerial vehicles. In: International Conference on Communication Technology Proceedings, ICCT, October 2019, pp. 1256–1260 (2019)
12. Park, W.-J., Kim, D.-H., Suryanto, Lyuh, C.-G., Roh, T.M., Ko, S.-J.: Fast human detection using selective block-based HOG-LBP. In: 19th IEEE International Conference on Image Processing, pp. 601–604 (2012)
13. Sheng, Y., Jiefa, W., Lingling, Z.: A fast pedestrian detection method based on simplified HOG descriptor. Int. J. Digit. Content Technol. Appl. **6**(4), 14 (2012)
14. Vasuki, P., Veluchamy, S.: Pedestrian detection for driver assistance systems. In: International Conference on Recent Trends in Information Technology, pp. 1–4 (2016)
15. Son, H., Lee, S., Choi, J., Min, K.: Efficient pedestrian detection by Bin-interleaved Histogram of Oriented Gradients. In: IEEE Region 10 Conference-TENCON, pp. 2322–2325 (2010)
16. Min, K., Son, H., Choe, Y., Kim, Y.-G.: Real-time pedestrian detection based on a hierarchical two-stage Support Vector Machine. In: 8th IEEE Conference on Industrial Electronics and Applications, pp. 114–119 (2013)
17. Yuan, X., Cai-nian, L., Xiao-liang, X., Mei, J., Jian-guo, Z.: A two-stage hog feature extraction processor embedded with SVM for pedestrian detection. In: IEEEInternational Conference on Image Processing, pp. 3452–3455 (2015)
18. Wang, M.-S., Zhang, Z.-R.: FPGA implementation of hog based multi-scale pedestrian detection. In: IEEE International Conference on Applied System Invention, pp. 1099–1102 (2018)
19. Hearst, M., Dumais, S., Osuna, E., Platt, J., Scholkopf, B.: Support vector machines. IEEE Intell. Syst. Appl. **13**(4), 740–755 (1998)
20. Vapnik, V., Cortes, C.: Support vector networks. Mach. Learn. **20**, 622–628 (1995)
21. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)

22. Lin, T.Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) Computer Vision – ECCV 2014. ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
23. Čehovin, L., Leonardis, A., Kristan, M.: Visual object tracking performance measures revisited. IEEE Trans. Image Process. **25**(3), 1261–1274 (2016)