Real-time Human Detection based on Personness Estimation

Kyuwon Kim[1,2]
q1.kim@samsung.com

Kwanghoon Sohn[1]
khsohn@yonsei.ac.kr

[1] School of Electrical and Electronic Engineering
Yonsei University
Republic of Korea

[2] Mobile Division
Samsung Electronics
Republic of Korea

Abstract

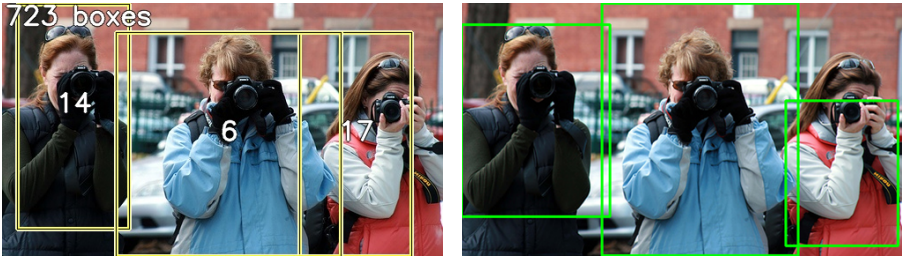
In this work, we study a real-time human detection method for mobile devices using window proposals. We find that the normed gradients, designed for generic objectness estimation, are also able to rapidly generate high quality object windows for a single-category object. We also notice that fusing the normed gradients with additional color feature improves the performance of objectness estimation for the single-category object. Based on these observations, we propose an efficient method, which we call personness estimation, to produce candidate windows that are highly likely to contain a person. The produced candidate windows are used to search over feature maps of an image so that a human detection method can achieve high detection performance within a short period of time. We further present how personness estimation can be efficiently combined into part-based human detection. Our experiments indicate that the proposed method is directly applicable to mobile devices, and allows real-time human detection.

1 Introduction

Human detection has been extensively researched under various names such as pedestrian detection [6], people detection [11, 13], and head-shoulder detection [17] in recent years. Human detection for mobile devices also plays an important role in various applications such as human computer interaction, automatic human focusing, and surveillance. Among many human detection algorithms, deformable part model (DPM) has become one of the most popular human detection methods [5] since Felzenszwalb *et al*. proposed the discriminatively trained part-based model [9]. However, it is difficult to see a practical application of DPM human detection on mobile devices due to its computational overhead.

The DPM method constructs a feature pyramid composed of multiscale feature maps, and takes the sliding window approach for each feature map. The DPM method also needs several mixture models describing various human poses. Each mixture model has one root filter capturing the overall body shape in lower resolution and several part filters capturing detailed body parts. Thanks to these sophisticated techniques and procedures, it becomes possible to detect humans with good detection rate. But a system running the algorithms

Pages 113.1-113.12
DOI: https://dx.doi.org/10.5244/C.29.113



(a) (b)

Figure 1: An example of human detection based on personness estimation: (a) The best matched windows generated by personness estimation: Total 723 candidate windows are generated, and 6th, 14th, and 17th windows are matched. (b) Human bounding boxes (BB) detected with the aid of candidate windows.

needs huge computational power when computing filter scores by performing convolutions between the filters and feature maps with the sliding window approach. Take, for example, the case of DPM human detection on a 375×500 image, OpenCV implementation constructs 33 scale levels of feature pyramid, and performs 1,786,962 convolution operations between 14 filters and the feature pyramid [13]. According to the sample profiling report of Microsoft Visual Studio 2012 [12], the convolution operation dominates the total detection time (which is about 1.4 seconds) with 53.47% for the 375×500 image.

Many object/human detection papers try to improve detection speed by optimizing each procedure of DPM with better algorithms, strong CPUs (plus SIMD instructions), and many GPUs on a desktop PC [2, 16, 19]. But, when software is developed and submitted to some application stores, one cannot assume what kinds of device our algorithm runs on. To make things worse, processors in mobile devices are much slower than desktop CPUs, and do not have enough GPU cores to boost the algorithm speed. Therefore, optimizing each procedure of DPM and making it several times faster are not enough for mobile devices. DPM should be able to search for humans from the most promising windows in mobile devices. Detection algorithms for mobile devices work under time constraints. Heavy detection algorithms that are not responding for a long time can affect system performance or cause inconvenience to users. Moreover, DPM detection for mobile devices needs several mixture models, since various poses and partially occluded shapes are frequently captured. Considering hardware limitations and captured high resolution images of mobile devices, an exhaustive search for DPM human detection is not a practical approach.

On the other hand, *objectness* (a.k.a. *detection proposals*) measure methods have gained popularity recently [10]. An objectness measure generates object windows that are likely to contain generic objects, and allows avoiding an exhaustive search. Its intention to improve detection speed seems to be perfectly matched with real-time detection in mobile devices. However, when our DPM implementation based on [13] spends about 200 milliseconds searching over all multiscale feature maps, existing objectness measure methods reviewed and evaluated in [10] spend more than 250 milliseconds. Since spending more than 250 milliseconds only for generating candidate windows is not well suited for real-time detection, it is difficult to adopt current measure methods except for BING which takes about 15 milliseconds on our platform. An objectness measure method needs to be reasonably faster than an exhaustive search in real-time detection.

For this reason, we propose a more efficient and accurate method for estimating human windows in an image in order to enhance detection rate within a short period of time. In this study, we are interested in humans, not generic objects. So we name objectness estimation [1] for humans simply as *personness* estimation.

Our contributions are as follows:

1. **We show that our DPM human detection method based on personness estimation is efficient** as demonstrated in section 3. In our experiments, personness estimation makes the DPM human detector obtain more than 70 percent of its original performance with only 10ms window search that includes generating windows and performing convolutions. When a detector looks for a single-category object in real time, too many candidate windows for all possible objects make the detector rather slow. We show how to avoid generating too many windows based on the linear support vector machine (SVM). We also show how to use additional color feature for the single-category object using the SVM framework.

2. **We show how a DPM detector can make better use of objectness estimation.** Our DPM design efficiently computes filter responses using provided candidate windows under time constraints. The DPM implementation also considers two important factors (*aspect-ratio threshold* and *patch size* for *pinpoint* explained in subsection 2.3) to achieve better detection performance using window proposals.

3. **We introduce the *recall-time* metric to evaluate objectness measure methods for real-time DPM detection.** Speed is very important in comparing objectness measure methods for real-time detection, and should be involved in the metric. To our best knowledge, there has been no previous metric that consider both the speed and the quality of objectness measure method. Our metric imposes importance on speed of an objectness measure, and evaluates the measure based on the detector performance it supports. We make all code related to our experiments publicly available to facilitate the study of real-time object estimation[1].

2 The proposed method

We choose the following two features to take the discriminative approach on PASCAL VOC datasets [7].

Edge As one can see from the success of HOG [5], there are various strong edges (oriented gradients) in and around humans. To get the edge information, we adopt the normed gradients (NG) feature [4].

Color We use a wide range of skin colors for color feature. But we take into account of only small region around head, since skin colors can be also shown in background and clothes.

To rapidly determine the priority order of each feature vector with a supervised approach, we make use of the fast NG feature [4]. BING is the approximated version of the NG feature [4]. For the convenience, we call the objectness measure method using BING or NG feature *Bing*. In this section, we describe the NG feature first, and then explain our proposed method in detail.

[1] http://q1kim.github.io/personness/

(a) (b) (c) (d)

Figure 2: NG features and a 8×8 downsampled example: (a) Objectness filter $\mathbf{w}$ generated with the original implementation [4]. (b) Personness filter $\mathbf{w}_p$ for human detection. (c) A 128×256 human window. (d) The 8×8 downsampled image of (c). The yellow box emphasizes the head part in (c).

2.1 Normed gradients and objectness estimation

An NG feature is a 64 dimensional vector describing the magnitude (a.k.a. *norm*) of the gradients of an 8×8 downsampled image. The authors of [4] found that one can efficiently estimate whether an image window contains an object or not by examining the NG feature of the window. As shown in Fig. 2(a), the learned 64 dimensional linear model $\mathbf{w} \in \mathbb{R}^{8\times 8}$ obtained by using the first stage linear SVM [8] on VOC 2007 dataset [7] shows a distinctive gradients roughly similar to a circle. This center-surrounded pattern implies that generic objects have well-defined closed boundaries [4]. To examine every image window for the objectness measure, a magnitude map of gradients is resized to 36 predefined *quantized shapes* (or *quantized sizes* in [4]). The results are called *NG maps*. By calculating the correlation values between the NG maps and $\mathbf{w} \in \mathbb{R}^{8\times 8}$, we can obtain a filter score for each window. Because each quantized shape has information about the original scale and aspect ratio, we can bring back the original window shape for the greatest filter score. It is also important to note that, in the public code [2] of [4], a quantized shape is ignored in the predicting stage if the quantized shape has less than or equal to 50 positive samples in the training stage. The final objectness score is calculated as follows:

$$o_i = v_i \cdot s_{(i,x,y)} + t_i \tag{1}$$

where $v_i, t_i \in \mathbb{R}$ and $s_{(i,x,y)}$ are learned coefficient, a bias term and the filter score at position (x,y) of each quantized shape i. $s_{(i,x,y)}$ is obtained using the first stage linear SVM, and v_i and t_i are obtained using the second stage linear SVM. We refer the reader to [4] for more details.

The simple framework allows high computational efficiency. However, *Bing* is not considered to be well designed for detection tasks by Hosang *et al*. and Qiyang *et al*. [10, 14]. It is because *Bing* is optimized for an intersection over union (IoU) of 0.5 [10, 14]. Using *Bing*-based personness estimation, we show that this weakness can be compensated for by modifying a DPM detector to calculate additional filter responses around the target location.

[2] http://mmcheng.net/bing/

2.2 From objectness to personness

Even though SVM training on various object categories makes the original linear model $\mathbf{w} \in \mathbb{R}^{8\times8}$ work for generic objects, it degrades the performance of the SVM classifier for human detection, since, for example, a person and a chair have huge difference in shape. And in many images of VOC 2007 dataset, head is not shown for the ground truth region of a person. These difficult person images in the VOC dataset also impair the classifier's performance. Therefore, we generate the new linear model $\mathbf{w}_p \in \mathbb{R}^{8\times8}$ shown in Fig. 2(b) by training the linear SVM only on humans that our DPM detector can detect. This restricted training technique also reduces the number of quantized shapes to consider. Here, our DPM implementation has two mixture models, one is for full-body and the other is for upper body. Fig. 2(b) illustrates that $\mathbf{w}_p \in \mathbb{R}^{8\times8}$ surprisingly places more confidence in the shoulder and head regions. Therefore, our personness estimation can be understood in the same context of *edgelet* [18] and *edge pattern* [17] that are trying to capture the strong edges of humans to enhance the detection performance.

Skin color is another important feature to measure the personness. After training the linear SVM on VOC 2007 dataset, we choose four points $(3,1),(3,2),(4,1),(4,2)$ to extract skin color information. Head and neck are usually located in these four points as shown in the Fig. 2(d). Even though the personness score of a silhouette or a back of a person where skin is not shown is decreased, it is obvious that people with skin exposed are more likely to be captured. Because we want to search for humans from the most promising windows, making use of skin color of the face part generates better candidate windows.

When function $\mathrm{skin}(x,y)$ returns a binary value indicating whether the point (x,y) is skin color according to the skin color model in [15], the skin score at position (x,y) is

$$c_{(i,x,y)} = \frac{1}{4}\sum_{j=1}^{2}\sum_{k=3}^{4}\mathrm{skin}(x+k,y+j) \tag{2}$$

And the new personness score is as follows:

$$p_i = v_i \cdot s_{(i,x,y)} + u_i \cdot c_{(i,x,y)} + t_i \tag{3}$$

where $u_i \in \mathbb{R}$ is learned coefficient for skin score using the second stage linear SVM. Even though we use additional skin information, the personness measure is faster than the objectness measure with the NG feature because of the reduced number of quantized shapes.

2.3 Combining DPM and personness

We divide DPM into three stages: setup, detection and evaluation. In the setup stage, DPM builds a feature pyramid, and initializes three pyramids: filter response pyramid, *probe* flag pyramid and convolution flag pyramid. Because our DPM does not perform convolutions from left-top to right-bottom, we make sure that duplicated computations do not arise by adopting the two flag pyramids (*probe* and convolution). *probe* implies a trial of performing selective convolutions triggered by a given window. In the detection stage, filter responses are computed between root/part filters and some selected feature vectors. Given a candidate window, how can we effectively select feature vectors in the feature pyramid?

In the estimation stage, personness estimation just produces many candidate windows, sorted in descending order according to their personness scores. In detection stage, on the other hand, DPM needs to know not only scale level l of the filter response pyramid to identify a filter response map, but also location $(\bar{x},\bar{y})$ of the filter response map to select feature

Algorithm 1 Converting a window to pinpoints

Input: *threshold* - aspect-ratio threshold
window - a candidate window
rootFilter - a root filter
probePyramid - the probe flag pyramid
Output: *pinPoints* - $(l,\bar{x},\bar{y})$ vector

1: $ratio \leftarrow$ compare-aspect-ratio(*window*, *rootFilter*)
2: **if** $ratio \geq threshold$ **then**
3: Throw away *window*
4: **else**
5: level $\hat{l} :=$ level-search(*window*, *rootFilter*)
6: **for** $l \leftarrow \hat{l}-1$ **to** $\hat{l}+1$ **do**
7: $(\bar{x},\bar{y}) \leftarrow$ center(*window*)/scale(l)
8: **if** $probePyramid(l,\bar{x},\bar{y}) \neq$ true **then**
9: Add $(l,\bar{x},\bar{y})$ to *pinPoints*
10: $probePyramid(l,\bar{x},\bar{y}) \leftarrow$ true
11: **end if**
12: **end for**
13: **end if**

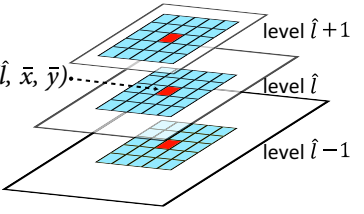


Figure 3: Three 5×5 patches around the pinpoints (in red) in a response pyramid of a root filter.

vectors to be convolved. With width w, height h, and center position $(x+w/2, y+h/2)$ of a window, we can pick the corresponding $(l,\bar{x},\bar{y})$. We call the exact response-map coordinates $(l,\bar{x},\bar{y})$, which correspond to a given window, *pinpoints*. The more detailed procedure of the window-pinpoints conversion is described in Algorithm 1. At this conversion from windows to pinpoints, one should keep in mind that a window for personness was quantized, so it might not be completely fitted to a real human window. Therefore, two important factors are additionally considered for the window-pinpoints conversion.

Aspect-ratio threshold Aspect ratios of candidate windows should be similar to the ones of root filters. The detection performance significantly increases by ignoring largely different windows in shape. Conversely, the detection performance decreases if too many candidate windows are abandoned. Therefore, the parameter *threshold* shown on line 2 in Algorithm 1 needs to be carefully decided. From our experiments on VOC datasets, we found that setting *threshold* to 1.52–1.55 achieves the best performance for detection[3].

Patch size for pinpoint For each computation of the root filter responses, our DPM method performs additional convolutions around the pinpoint in the filter response pyramid. We call the area that we additionally consider *patch*. In our implementation, additional two layers of patches are also used for each candidate window as shown in Fig. 3 and Algorithm 1. The probe flag pyramid prevents duplicated convolutions caused by the same pinpoint as shown on line 8 in Algorithm 1. The convolution flag pyramid prevents duplicated convolutions caused by a near pinpoint located in the same patch. Increasing the patch size allows DPM to detect objects even with slightly overlapping

[3]The experimental results about the aspect-ratio threshold and the patch size are available in the supplementary material.

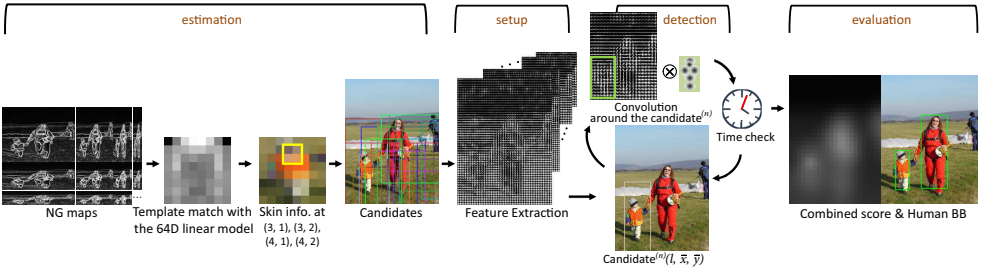


Figure 4: The flowchart of the proposed method: Our method has four stages (estimation, setup, detection, evaluation). In the estimation stage, candidate windows are generated by personness estimation. The setup, detection, and evaluation are performed by the DPM human detector. The generated windows in the estimation stage are used to perform selective convolutions in the detection stage. Detection threads check the remaining time at every 3 windows.

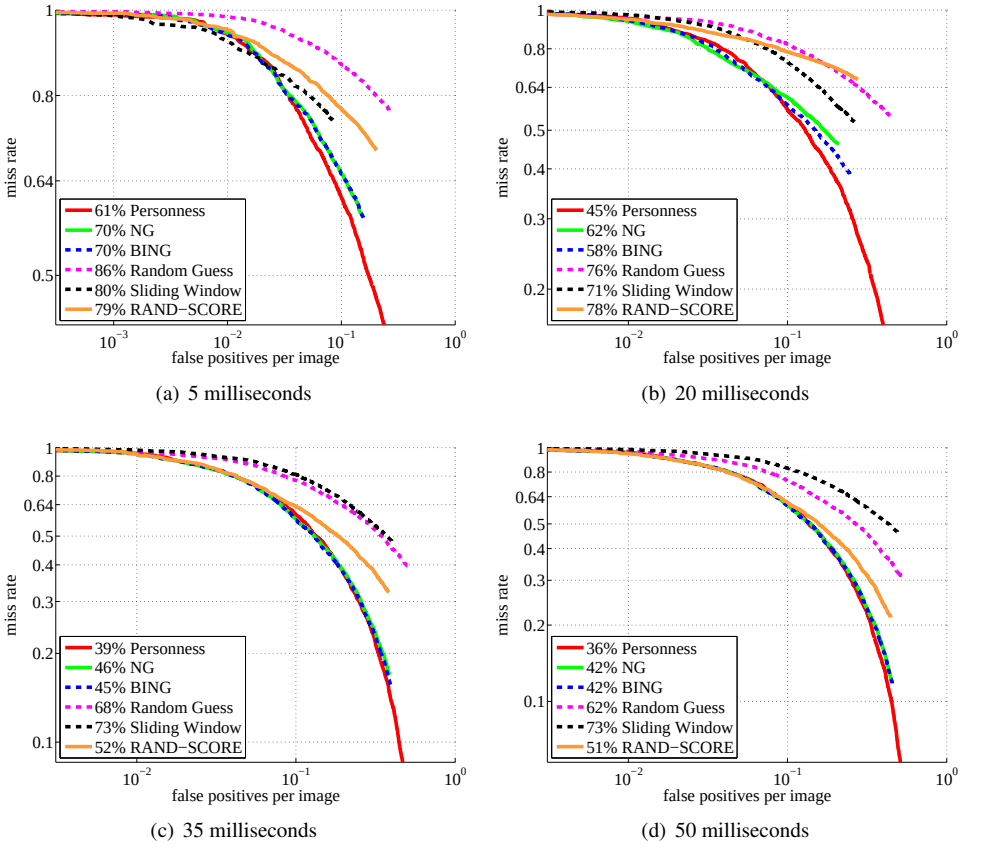
candidate windows. Doing so, however, causes a problem of performing unnecessary convolutions. Another problem of enlarging patch is that detection can exceed its allowed time budget. Our experiments demonstrate that size 5×5 or size 7×7 yields good performance[3].

After DPM performs convolutions between the chosen feature vectors and a root filter, it calculates part filter responses of feature vectors corresponding to the root positions. When time runs out, the detector finishes computing filter responses, and moves to the evaluation stage. In the evaluation stage, the detector computes the overall root scores as discussed in [9]. Fig. 4 illustrates the overall procedure of our method.

3 Experimental results

We evaluate the performance of our personness measure on VOC 2007/2012 datasets [7]. The images of VOC datasets capture realistic scenes that our method exactly aims at. For all the experiments, we use the same the 1.54 aspect-ratio threshold and 7×7 patch size. The personness measure is compared with *Bing*, random guess, sliding windows, and RAND-SCORE [14]. Qiyang *et al*. proposed RAND-SCORE and, according to them, it also generates promising windows whose IoU is above 0.5 [14]. We could not evaluate the objectness measure methods reviewed in [10] except for *Bing*, since their speeds are slower than the sliding window approach of our DPM implementation. To fairly compare our method with other methods, we remove all the ground truth boxes that the DPM with exhaustive search cannot detect. We evaluate each measure method based on how much it contributes to the DPM human detector in a given time. We consider a person is detected if IoU between a detector's BB and a ground truth BB is greater than 0.5.

For the robust linear model $\mathbf{w}_p \in \mathbb{R}^{8\times 8}$ and reliable evaluation, we use 2984 human windows in 1959 images of VOC 2007 dataset for training, and 6137 human windows 4806 images of VOC 2012 dataset for testing. We make sure that VOC 2007 images in VOC 2012 dataset are excluded from testing. Our method is implemented in C++ using OpenCV



(a) 5 milliseconds (b) 20 milliseconds

(c) 35 milliseconds (d) 50 milliseconds

Figure 5: Log-average miss rate for different time limits

library, and all experiments are carried out with an Intel i7-4470 3.40GHz processor under Windows 7 64-bit environment without OpenMP.

As a performance measure, we first plot the log-average miss rate (LAMR) against *false positives per image* [6] at 5ms, 20ms, 35ms, and 50ms time limits as shown in Fig. 5. The time limits are imposed on the estimation and detection stages that are related to window search. In the detection stage, our DPM implementation checks the elapsed time at every 3 candidate windows, and if the time limit is exceeded, it finishes computing filter responses. Fig. 5 indicates that our method has the lowest LAMR at each time limit. This means that our personness measure provides the best candidate windows for DPM detection in our experiments. We present the illustrative results for the 25ms time limit in Fig. 7.

We also measure recall with different time limits from 5ms to 55ms for six methods to show the efficiency of our method clearly. Fig. 6 illustrates the recall-time curves. Our method in red line reaches high recall value in a much shorter time than other methods. Considering that it takes about 135ms for the conventional sliding window approach to reach 0.9 recall value, the personness measure shows outstanding estimation ability to help the DPM method achieve about 0.9 recall value within only 30ms. Personness estimation is more effective especially when a given time budget is very limited. The DPM method achieves

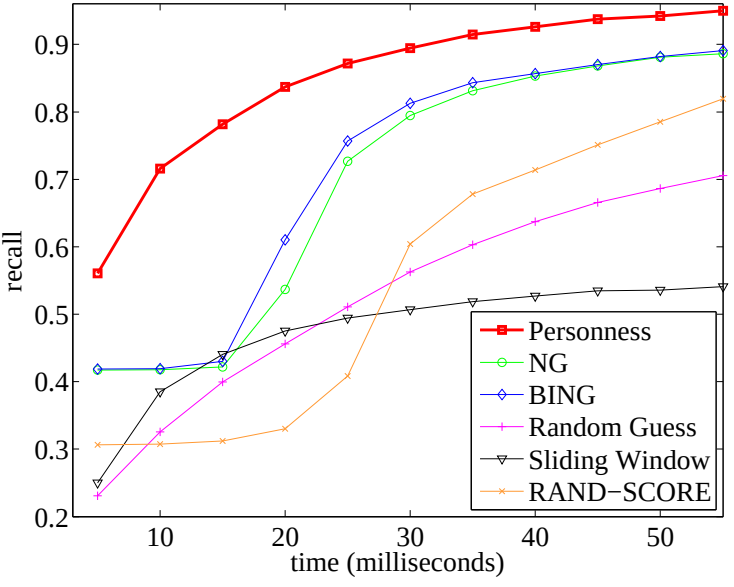


Figure 6: The recall-time graph

more than 0.5 recall value within 5ms with personness estimation. With the 15ms time limit, the recall value of personness estimation is at least 0.44 greater than other methods. It is important to note that these experiments are carried out on a desktop PC. So in mobile devices with less powerful processors, we believe that our method becomes more useful due to larger performance gap between personness estimation and others.

It is also interesting that the distinctive performance difference between *Bing* and RAND-SCORE can be observed with the recall-time graph. The influence of *Bing* jumps after 20ms because their processing time for estimation is about 15ms. However, the influence of RAND-SCORE [14] is still minimal. This is a different result from what Qiyang *et al*. observed in [14]. They state in [14] that the performance of RAND-SCORE is rather good or sometimes even better than *Bing* with the *detection-rate/windows-amount* metric.

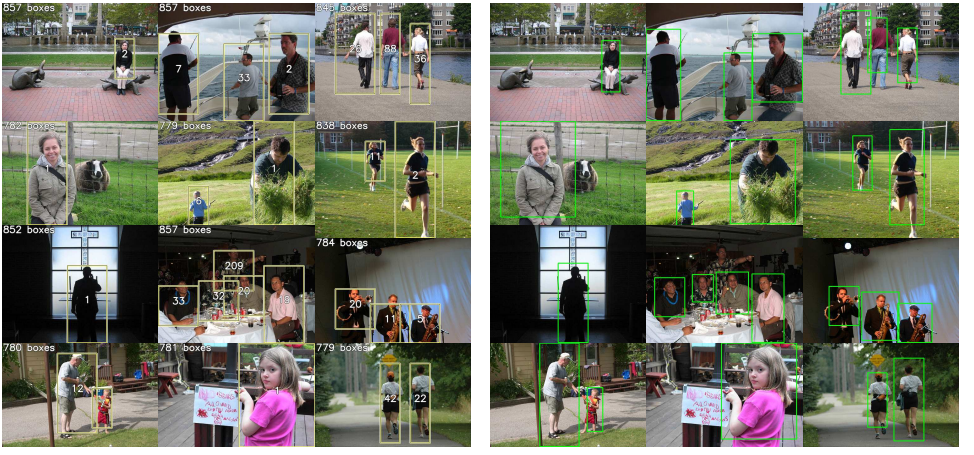
As explained in subsection 2.2, training the linear SVM only on human windows in VOC 2007 images reduces the number of quantized shapes to 16. This reduced number of quantized shapes makes our personness measure faster than the objectness measure with the *Bing* features as shown in table 1.

Table 1: Average estimation time on VOC 2012 dataset [7]

	NG	RAND-SCORE	BING	Personness
Time (milliseconds)	16.06	15.05	14.66	**3.69**

3.1 Discussion

In [10], Hosang *et al*. state that the objectness measure methods specialized in IoU of 0.5 are not effective for object detection. Indeed, if an objectness measure method can generate more



(a) (b)

Figure 7: Personness based human detection with the 25ms time limit on VOC 2012 dataset [7]: (a) The best matched windows produced by personness estimation. Window indices are written at the center of each window. The window index starts at 1. As can be seen, low-index windows are matched well among about 800 candidate windows. (b) Obtained human BBs with the help of personness estimation. It is worthwhile to note that these results are obtained with only 25ms window search. That is, the time budget for personness estimation and convolution computation is 25ms. Because of the limited time budget, a false negative case is also shown.

accurate windows, the DPM detector does not require larger patches, and therefore inspects more windows in a given time. However, we believe that IoU of 0.5 is still meaningful to real-time detection, especially for small number of categories. The problem of lower overlap (IoU = 0.5) can be overcome by increasing the patch size in DPM. If low-index windows have more chances to contain objects, good detection rate can be achieved as demonstrated in this section. Under real-time constraints, instead of consuming much time for accurate windows, we think that starting searching early with 0.5-IoU windows can be a reasonable method.

4 Conclusions

In this paper, we proposed personness estimation that generates promising human windows within a short period of time for real-time human detection. To show the efficiency and practicality of personness estimation, we designed a real-time DPM human detection that makes effective use of personness estimation. The experiments on VOC 2007/2012 datasets [7] indicate that our proposed method allows to use complex detection algorithms even for real-time human detection. Our personness estimation itself can be orthogonally applied to other human detection algorithms apart from DPM. In our future work, we would like to study human detection with convolutional neural network (CNN) on mobile devices and its personness estimation.

References

[1] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34:2189–2202, November 2012.

[2] R. Benenson, M. Mathias, R. Timofte, and L. Van Gool. Pedestrian detection at 100 frames per second. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2903–2910, 2012.

[3] R. Benenson, M. Omran, J. Hosang, and B. Schiele. Ten years of pedestrian detection, what have we learned? In *Proc. of the European Conference on Computer Vision Workshop*, 2014.

[4] M. Cheng, Z. Zhang, W. Lin, and P. Torr. BING: Binarized normed gradient for objectness estimation at 300fps. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3286–3293, 2014.

[5] N. Dalal and B. Triggls. Histograms of oriented gradients for human detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 886–893, 2005.

[6] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 34: 743–761, Apr. 2012.

[7] M. Everingham, V. Gool, L. Williams, C. K. I., J. Winn, and A. Zisserman. The pascal visual object classes (VOC) challenge. *International Journal of Computer Vision*, 88: 303–338, Jun. 2010.

[8] R. Fan, K. Chang, C. Hsieh, X. Wang, and C. Lin. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[9] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 32:1627–1645, Sep. 2010.

[10] J. Hosang, R. Benenson, P. Dollár, and B. Schiele. What makes for effective detection proposals? *arXiv:1502.05082*, 2015.

[11] A. A. Mekonnen, F. Lerasle, A. Herbulot, and C. Briand. People detection with heterogeneous features and explicit optimization on computation time. In *Proc. of the IEEE International Conference on Pattern Recognition*, pages 4322–4327, 2014.

[12] Microsoft. Understanding profiling methods. http://msdn.microsoft.com/library/dd264994.aspx, 2012. [Online; accessed 6-May-2015].

[13] OpenCV. Latent svm. http://docs.opencv.org/modules/objdetect/doc/latent_svm.html, 2014. [Online; accessed 6-May-2015].

[14] Z. Qiyang, L. Zhibin, and Y. Baolin. Cracking bing and beyond. In *Proc. of the British Machine Vision Conference*. BMVA Press, 2014.

[15] N. A. A. Rahman, K. C. Wei, and J. See. Rgb-h-cbcr skin colour model for human face detection. Faculty of Information Technology, Multimedia University, 2004.

[16] M. A. Sadeghi and D. Forsyth. 30Hz object detection with DPM V5. In *Proc. of the European Conference on Computer Vision*, pages 65–79, 2014.

[17] S. Wang, J. Zhang, and Z. Miao. A new edge feature for head-shoulder detection. In *Proc. of the IEEE International Conference on Image Processing*, pages 2822–2826, 2013.

[18] B. Wu and R. Nevatia. Detection of multiple, partially occuluded humans in a single image by Baysian combination of edgelet part detectors. In *Proc. of the IEEE International Conference on Computer Vision*, volume 1, pages 90–97, 2005.

[19] J. Yan, Z. Lei, L. Wen, and S. Z. Li. The fastest deformable part model for object detection. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2497–2504, 2014.