Research Article

INTERNATIONAL JOURNAL OF
ADVANCED ROBOTIC SYSTEMS

International Journal of Advanced Robotic Systems
July-August 2017: 1–11
© The Author(s) 2017
DOI: 10.1177/1729881417724679
journals.sagepub.com/home/arx
SAGE

An improved real-time object proposals generation method based on local binary pattern

Yanting Jiang, Jia Yan, Ci'en Fan, Wenxuan Shi and Dexiang Deng

Abstract

Generating a group of category-independent proposals of objects in an image within a very short time is an effective approach to accelerate traditional sliding window search, which has been widely used in preprocessing step of object recognition. In this article, we propose a novel object proposals generation method to produce an order set of candidate windows covering most of object instances. With combination of gradient and local binary pattern, our approach achieves better performance than BING in finding occluded objects and objects in dim lighting conditions. In experiments on the challenging PASCAL VOC 2007 data set, we show that our approach is significantly more accurate than BING. In particular, using 2000 proposals, we achieve 97.6% object detection rate and 69.3% mean average best overlap. Moreover, our proposed method is very efficient and takes only about 0.006 s per image on a laptop central processing unit. The detection speed and high accuracy of proposed method mean that it can be applied to recognizing specific objects in robot visions.

Keywords

Objectness, proposal generation, local binary pattern, computer vision, object detection

Date received: 25 October 2016; accepted: 3 July 2017

Topic: Vision Systems
Topic Editor: Antonio Fernandez-Caballero
Associate Editor: Antonios Gasteratos

Introduction

In recent years, object detection has made great strides and has been widely used in computer vision and robotic vision. Many vision tasks, such as pointing gestures for human–robot interaction, face recognition for robot vision, and route recognition for mobile robot, are closely tied to object recognition algorithm. There are already some effective methods improving the object detection performance through a variety of complex features[1,2] or classifiers.[3] However, most state-of-the-art detectors still determine the most possible object positions over the image by sliding windows,[4–6] which are computationally expensive to evaluate all locations. For a successful detection system, it is quite an important problem to increase the computational efficiency without losing the detection accuracy. In order to accelerate object detection, objectness proposals generation has recently attracted much attention.[7–9]

Objectness proposals generation, which produces some category-independent candidate windows of objects in an image within a very short time, has been widely used in the preprocessing step of object recognition. We hope to generate a small number of bounding boxes, such that each object is well located by at least one box. Thereby, object

Electronic Information School, Wuhan University, Wuhan, China

Corresponding author:
Jia Yan, Electronic Information School, Wuhan University, Luo-Jia-Shan, Wuchang District, Wuhan 430072, China.
Email: yanjia2011@gmail.com

recognition algorithms would be able to evaluate a complex classifier only at a small group of plausible regions rather than at all possible positions and scales in the whole image. Most state-of-the-art object detection frameworks[3,9–12] proposed recently are composed of proposal generation procedure and object recognition procedure. Being different with time-consuming exhausted search, the proposal generation phase remarkably reduces computation cost by generating a group of candidate proposals that may contain objects. Thus, sophisticated classifiers can be used for window assessment in object recognition. Although there exist a variety of proposal generation approaches, they can be put into two categories roughly grouping methods and window scoring methods.[13]

Grouping proposals methods normally employ superpixels grouping strategy to generate multiple (possibly overlapping) segmentations that are likely corresponding to complete objects. A typical approach is a selective search proposed by Uijlings et al.[1] It uses many randomly initialized seeds to start hierarchical superpixels merging based on diversified criteria to generate high-quality proposals.

In contrast, window scoring methods score to generate candidate windows with high ranks, such as objectness—first introduced by Alexe et al.[8] Objectness methods judge how likely it is for an image window to contain an entire object of any class and select windows which are scored based on multiple cues including color, boundary, and superpixel shape. BING[14] is a very fast objectness measure, which selects bounding boxes by training a simple linear classifier with gradient feature (a short review is given in “Overview of BING” section). EdgeBoxes[15] has no learned parameters (similar with selective search) and scores each window according to the number of complete contours in its edge map (obtained via structured forests[16,17]). Zhang et al.[18] proposed a cascade ranking SVMs to generate proposals for object detection. The first stage learns several classifiers for each scale and aspect ratio in a sliding window manner; and the second stage ranks all proposals from the previous stage. Endres and Hoiem[19] coarsely extracted regions following multiple cues and proposed structure learning algorithm to produce object proposals.

Generally speaking, objectness tends to be faster than grouping proposal methods because objectness only returns bounding boxes. Therefore, objectness has been recently applied to various computer vision tasks for improving accuracy or speed, such as pedestrian detection,[20] visual object detection,[21,22] salient region segmentation[23] robot vision, and so on.

Keeping the computational cost feasible is very important[24,25] for efficient object detection. In this article, we propose a new approach to locate objects by producing a small bag of objectness proposals which cover almost all object instances. According to experiment results, our method reaches surprising detection performance using standard metrics while being very fast to compute.

The article’s main contributions are as follows: (a) Firstly, we study how BING method works to obtain suggested windows and analyze its problem in locating occluded instances. (b) Secondly, we improve BING by adding local binary pattern (LBP) as a new feature into our model. On VOC 2007[26] data set, the detection rate (DR) of our method is increased from 96% to 97.5%, and the mean average best overlap (MABO) is increased from 65% to 69.3% for 2000 proposals. Moreover, our method would achieve over 99% DR when using 4000 proposals. By this way, we preserve the speed advantage of BING while reaching more accurate detection results, so that we could be able to provide higher quality windows for various detection tasks.

Overview of BING

BING method, an acceleration framework of generic objectness measure, has made significant breakthroughs on calculation efficiency (300 fps on VOC 2007) compared with the current state of the art. The outputs of BING is a small set of proposals covering most of objects rather than their precise locations. Based on the fact that objects are stand-alone things with well-defined closed boundaries and centers[27,28] (different from amorphous background stuff like grass, sky), Cheng et al.[14] observed that when resizing their corresponding windows to a small fixed size, their norm of gradients becomes a discriminative feature (named NG feature), regardless of objects with different shapes and colors. It is because that little variation of closed boundaries could be presented in such an abstract view.[14] In order to realize the acceleration of proposals generation, BING firstly defines 36 different sizes for windows quantification and employs simple norm of gradients to train a two-stage cascaded model with linear SVM. The advantage of gradient maps is that they preserve boundaries information completely with efficient data representation. In the test stage, each window is scored with a linear model $w \in R^{64}$. Window scoring formula is represented as

$$s_l = \langle w, g_l \rangle \tag{1}$$

where s_l and g_l are the filter score and NG feature, respectively. In order to avoid heavy computing when scoring windows, BING realizes speeding up by translating equation (1) into fast bitwise and POPCNT SSE operators.

Such a two-stage cascaded model provides a framework for fast proposals generation. However, we observe that the behavior of BING is not satisfactory in some cases. An apparent drawback of BING is that boundaries are not always closed for occluded or truncated objects. Actually, results of BING show that considerable undetected objects are partially or totally occluded by obstacles around them. Besides, we observe that BING is also poor at finding objects in dim lighting conditions because it’s hard to captain complete contour in such illumination conditions. Figure 1 gives several instances to testify drawbacks of

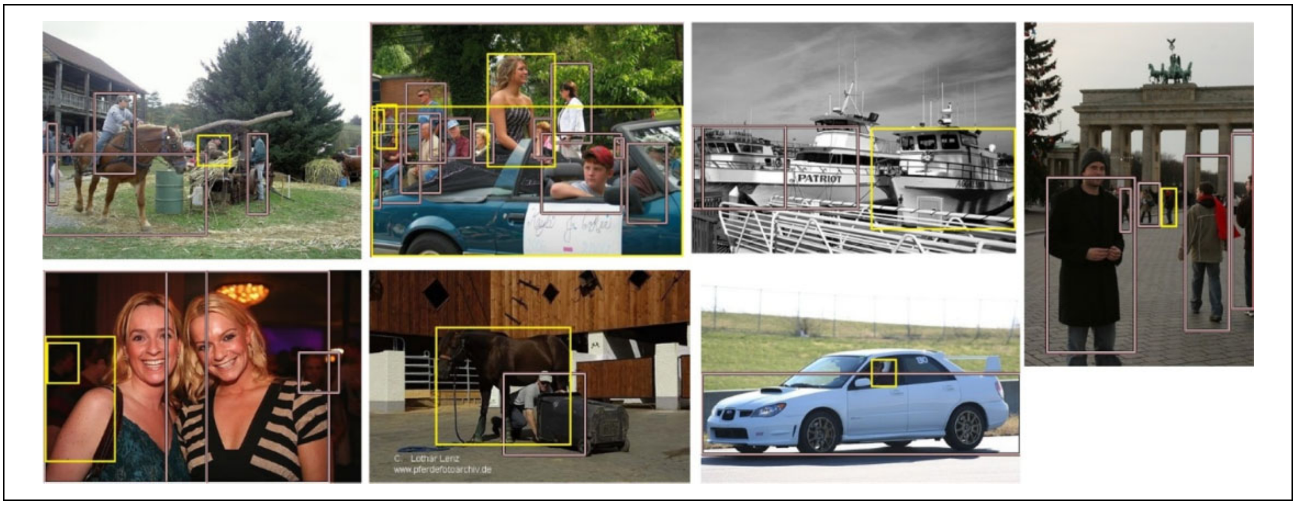

Figure 1. Instances of BING's detection result (pink proposals) on VOC 2007 test images. Proposals in yellow represent objects undetected with BING. Most of them are occluded or in dark lighting.

BING in detecting incomplete objects and objects in poor lighting conditions. The main reason is that the success of BING depends largely on simple gradient feature while boundaries are not always closed for all kinds of object instances. It's difficult to captain complete contour when objects are partially or totally occluded by obstacles around them. Similarly, the gradient feature of objects under dark situation usually cannot be distinguished from amorphous backgrounds stuff.

Motivated by this work, we could be able to improve detection performance by incorporating different kinds of features and classifiers into model training. We choose adding texture feature to increase robustness of new model for hard instances because that objects with incomplete boundaries would usually have distinguished texture from their backgrounds. Considering the balance of detection quality and computation efficiency, we employ LBP to describe image local texture.

Local binary patterns

Local binary pattern, a powerful description for image local texture, was first proposed by Harwoodet al.[29] The original LBP operator works with a 3 × 3 neighborhood by thresholding each pixel with the center value to obtain eight thresholded binary values (such as 00100011), which are saved as a BYTE value (0–255) to express the LBP code of center pixel. An instance of the LBP operator is shown in Figure 2. According to Harwood et al.,[29] the LBP code for a center pixel with coordinate (x, y) can be computed by

$$\mathrm{LBP}(x,\ y) = \sum_{p=0}^{7} s(g_c - g_p)2^p \tag{2}$$

where $s(z)$ is the threshold function: $s(z) = \begin{cases} 1, & z \geq 0 \\ 0, & z < 0 \end{cases}$ and g_c and g_p denote the value of center pixel and pixel in its eight-neighborhood, respectively.

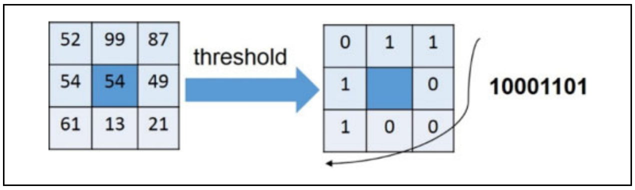

Figure 2. An instance of the LBP operator. The value of center pixel (x, y) is 54. LBP: local binary pattern.

Apparently, the result is a LBP map after calculating LBP code during the whole image. In applications of LBP such as texture classification[30] and face recognition,[31] people usually use the statistical histogram of LBP code rather than LBP map as feature vectors. However, we are going to use LBP map as a 64-D feature in our proposed model because of the usage of INT64 similar to BING. Since the original LBP was introduced, several improved LBP operators are proposed, such as an extension of LBP using circular neighborhoods of different sizes,[32] rotation-invariant LBP, and uniform patterns LBP. LBP operator and its extensions have been applied in different areas because of their good rotation invariance, robustness to illumination, and calculating efficiency. People usually combine LBP with HOG as features for human recognition,[33,34] which remarkably increases detection rate.

In this article, we use a modified LBP operator, called MLBP, to compose a 64-D feature of our model in consideration of effectiveness. MLBP code of a center pixel is calculated by comparing the value of each pixel in its neighborhood with the mean value of them represented by

$$\mathrm{MLBP}_{P,R} = \sum_{p=0}^{P-1} s(g_p - g_m)2^p \tag{3}$$

where $s(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$ and $g_m = \frac{1}{P+1}\sum_{p=0}^{P} g_p$. In equation (3), P is the number of pixels in neighborhood, g_p

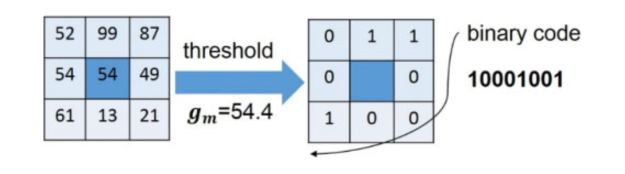

Figure 3. An instance of MLBP operator. The value of center pixel is 54. MLBP: modified local binary pattern.

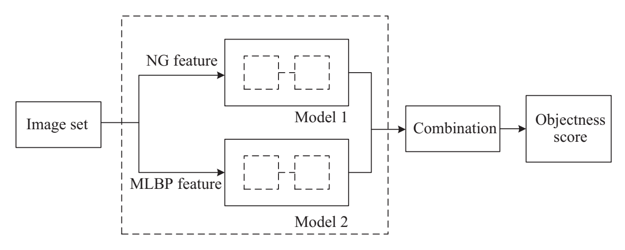

Figure 4. The framework of our method. It consists of two submodels training with NG feature and MLBP feature, respectively. The input of it is image data set and the output is a set of proposals with objectness score. MLBP: modified LBP.

is the value of a pixel in neighborhood, and g_m is the mean value of center pixel and all pixel in neighborhood. Figure 3 gives an instance of MLBP operator. We choose MLBP because it is powerful in texture description and is easy to compute.

Proposed approach

In this article, we propose an improved approach based on BING for efficient objectness estimation, which outputs an order set of windows containing object instances. A real-time object proposals generation method is very helpful in preprocessing step of object recognition, which can be widely applied in pattern recognition, artificial intelligence, and robot vision. Motivated by the observation that most undetected instances of BING method are occluded or in dim lighting, we learn a new model by combining NG feature and LBP feature, which enables to improve the robustness of model. By following clues of texture and contour, we can improve the detection rate as well as proposals quality. As shown in Figure 4, the framework of our approach consists of two analogous submodels: (1) the original model as in BING and (2) a model training with MLBP feature. Both of the two submodels are two-stage cascaded models sharing similar structure, and they are special for different types of object features. In the testing stage, each image window is scored with two learned models, respectively, and after a size-weighted step, our system would outputs a set of windows with top-ranking scores.

Since the training process of submodel 1 (same as model of original BING) is already introduced in “Overview of BING” section, next we will flesh out the training paradigm of submodel 2. As illustrated in Figure 5, four components are needed for submodel 2, including window quantization, feature extraction, learning a linear filter w, and learning a learnt coefficient vi and a bias term ti for each quantized size i. Besides, speeding up step is also indispensable for our model.

Window quantization

Quantization is an important step in our method. In order to collect more positive samples with diversity and get a more robust linear model, we quantize a ground truth window to several (normally two or three) windows of base-2 sizes (For example, a 196 × 174 window will be quantized to a 256 × 128 window and a 256 × 256 window, and a 214 × 113 window will be quantized to a 256 × 64 window, a 128 × 128 window, and a 256 × 128 window.), guaranteeing no less than 50% overlap with the original ground truth. By this way, all training samples are divided into 36 kinds based on their sizes, and samples in size i will be discarded if their amount is less than 50. Two instances of window quantization are illustrated in Figure 6. If the quantized window is partly out of range of image, the outranged part will be cut to adjust to image. These quantized windows will then replace the original ground truth window becoming positive samples of linear SVM after resizing and feature extraction.

Window quantization step not only reduces variation of sizes of samples but also largely simplifies the procedure of estimating windows in the sliding window manner over the image. Our model aims to generate a set of suggested windows in less than 36 kinds of sizes (our model generates suggested windows of $(Wi,\ Hi)$, where $Wi,\ Hi \in \{2^4, 2^5, 2^6, 2^7, 2^8, 2^9\}$). When scoring windows of size i (wi, hi), the original image $(imgW, imgH)$ is first resized to a smaller image of size $(W0,\ H0) = \left(\frac{8\times imgW}{wi}, \frac{8\times imgH}{hi}\right)$, so that a 8 × 8 window in the smaller image corresponds to a $wi \times hi$ window in the original image. Therefore, we can slide the learnt 8 × 8 filter W across the shrunken image to get a filter score of $wi \times hi$ window in each location.

Feature extraction

In training stage, we resize all quantized windows of each size i and randomly sampled windows to a uniform 8 × 8 size and then extract MLBP feature over them to form 64-D features, which are used as inputted samples of linear SVM. MLBP is an extension of original LBP operator introduced in “Local binary patterns” section. Figure 7 shows two examples of extracting 64-D MLBP feature from quantized window.

Learning a linear filter w

We train a linear filter w using linear SVM in the first learning stage. MLBP features of the ground truth object windows (after quantization) and randomly sampled background windows are used as positive and negative training samples, respectively. The learnt filter w is added to the first stage of cascaded model after normalization.

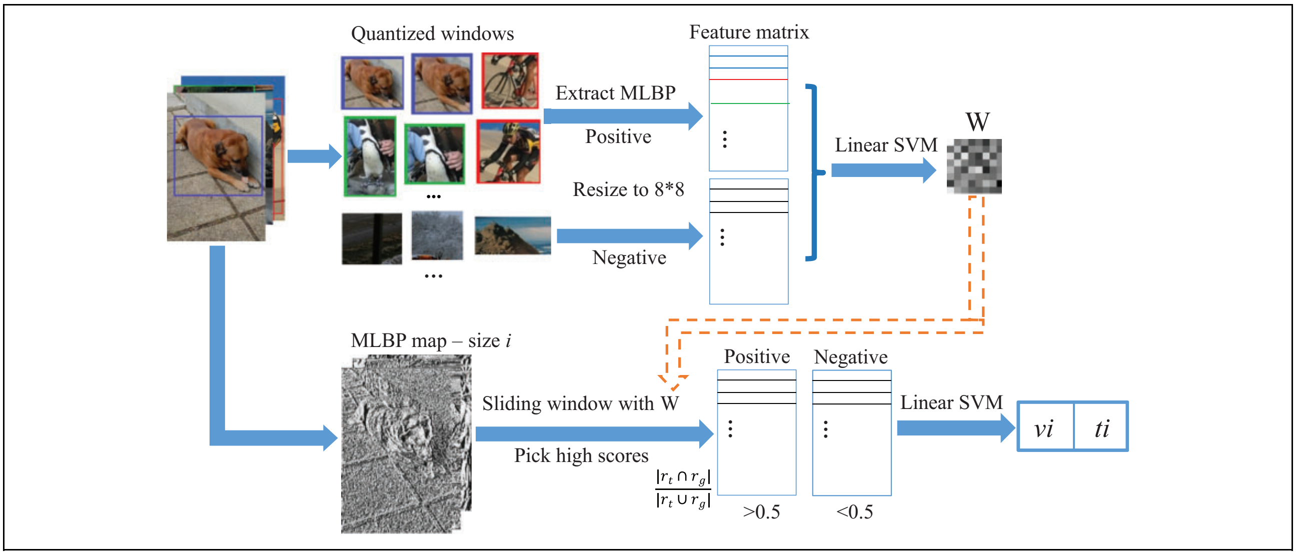

Figure 5. Illustration of submodel 2 in training stage, including window quantization, feature extraction, learning a linear filter *w*, and learning a learnt coefficient *vi* and a bias term *ti* for each quantized size *i*.

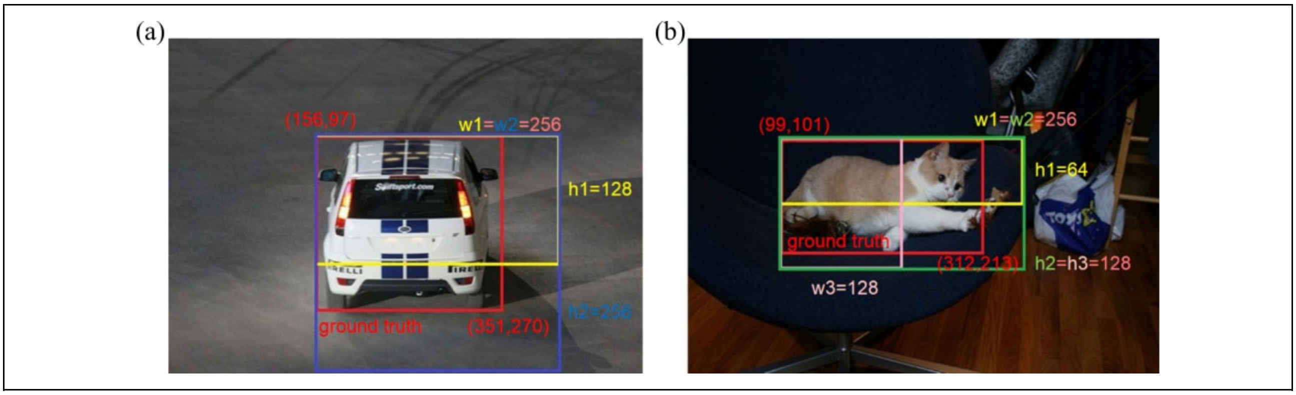

Figure 6. Two instances of window quantization on the PASCAL VOC 2007 training set. The ground truth windows are in red. (a) A toy car with two quantized windows (yellow and blue) and (b) a cat with three quantized windows (yellow, green, and pink).

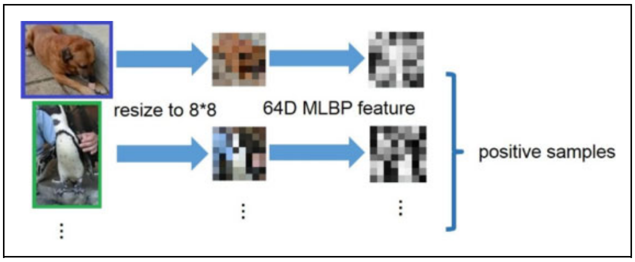

Figure 7. Two examples of extracting 64-D MLBP feature from quantized window. Extracted features are used as positive samples in linear SVM. MLBP: modified LBP.

Learning vi and ti

Since some sizes are more likely to contain an object instance than others, the second stage of model is designed to obtain learnt coefficient *vi* and bias term *ti* for each quantized size *i*. Firstly, we estimate sliding windows (for each quantized size *i*) with learnt filter *w* and then perform non-maximum suppression to select a set of windows with high-filter scores. These selected windows are divided into positive and negative samples according to their overlap scores with the ground truth. Actually, given the selected window bounding box (r_t) and the ground truth bounding box (r_g), the overlap score $s = \frac{\text{area}(r_t \cap r_g)}{\text{area}(r_t \cup r_g)}$ is used to determine sample labels. The filter scores of samples are used as 1-D feature. Therefore, we can obtain learnt coefficient *vi* and bias term *ti* for each quantized size *i*.

Speeding up

As we hope to generate proposals by scoring windows according to MLBP feature in a sliding window manner, we are following equation (1) to calculate filter score of a window. The convolution operation in equation (1) can be replaced with several bit operation after approximation, as

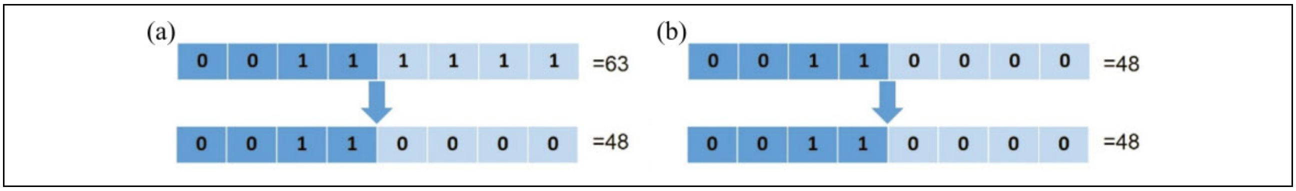

Figure 8. Two examples of MLBP feature approximation using its top four binary bits. (a) The situation of maximum error is 15-pixel difference and (b) the minimum error is 0-pixel difference.

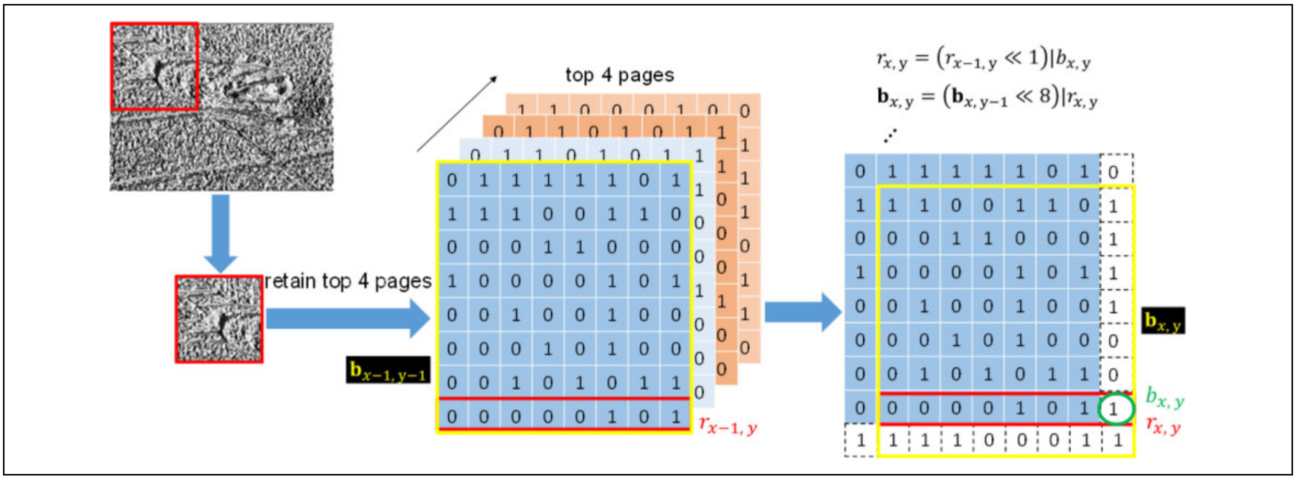

Figure 9. Illustration of translating data access into bit shift procedure when scoring window in a sliding window manner. Variable explanation: a MLBP window $\mathbf{b}_{x,y}$, its last row $r_{x,y}$, and last element $b_{x,y}$. MLBP: modified LBP.

much as possible reducing computation cost. Speeding up procedure includes approximation representation and INT64 data storage. The 64-D linear filter w is approximately represented as a linear combination of n_w binary vectors a_j^+ and their complements $\overline{a_j^+}$ weighted by corresponding coefficient β_j, where a_j^+, $\overline{a_j^+} \in \{0,1\}^{64}$ and $a_j = a_j^+ - \overline{a_j^+}$. Accordingly, scoring formula can be expressed as

$$\begin{aligned}\langle w, b_k\rangle &\approx \left\langle \sum_{j=1}^{n_w} \beta_j a_j, b_k \right\rangle \\ &\approx \sum_{j=1}^{n_w} \beta_j \left\langle a_j^+ - \overline{a_j^+}, b_k \right\rangle \\ &\approx \sum_{j=1}^{n_w} \beta_j (2\langle a_j^+, b_k\rangle - |b_k|) \end{aligned} \tag{4}$$

The approximation of extracted MLBP feature is represented by replacing each MLBP code using its top n_f binary bits, which can be expressed as

$$f_k = \sum_{i=1}^{n_f} 2^{8-i} b_{k,i} \tag{5}$$

where $b_{k,i}$ denotes the binary value of MLBP code at the i th bit. $n_f = 4$ means replacing MLBP feature using its top four binary bits, within accepting error range while reducing following computation to 50%. Two examples of MLBP feature approximation are illustrated in Figure 8.

According to above algorithms, the filter score of an image window can be efficiently evaluated using

$$s_l \approx \sum_{j=1}^{n_w} \beta_j \sum_{i=1}^{n_f} C_{j,i} \tag{6}$$

where $C_{j,i} = 2^{8-i}(2\langle a_j^+, b_k\rangle - |b_k|)$. In addition, we translate data access for sliding window into bit shift procedure (expounded in Figure 9) using INT64 variables, which should further increase computing efficiency of our system.

Experimental evaluation

In order to facilitate comparing our algorithm with previous approaches, we train our model and perform quantitative evaluation on PASCAL VOC 2007[26] data set. PASCAL VOC 2007[26] is a standard data set of image and annotation, in which each image is annotated with ground truth bounding boxes of objects from 20 categories (bird, aeroplane, cow, etc.). Since we want to find all objects in the image irrespective of their categories, we train our model on official training set with 6 object categories and evaluate it on testing set with other 14 unseen categories. The results of experiment show that our method reaches a higher performance than original BING.

Table 1. Performance comparison of different extensions of LBP combined with BING.[a]

#WIN	1000	2000	3000	5000
Original LBP + BING (%)	(95.8, 68.2)	(97, 68.5)	(98.3, 70.1)	(98.4, 70.1)
Max LBP + BING (%)	(96, 68)	(97.3, 68.7)	(98.5, 70.3)	(98.6, 70.3)
MLBP + BING (%)	(96.1, 68.2)	(97.5, 69.3)	(98.7, 70.5)	(99, 70.7)

DR: detection rate; MABO: mean average best overlap.

[a] The values within parenthesis means (DR × 100, MABO × 100).

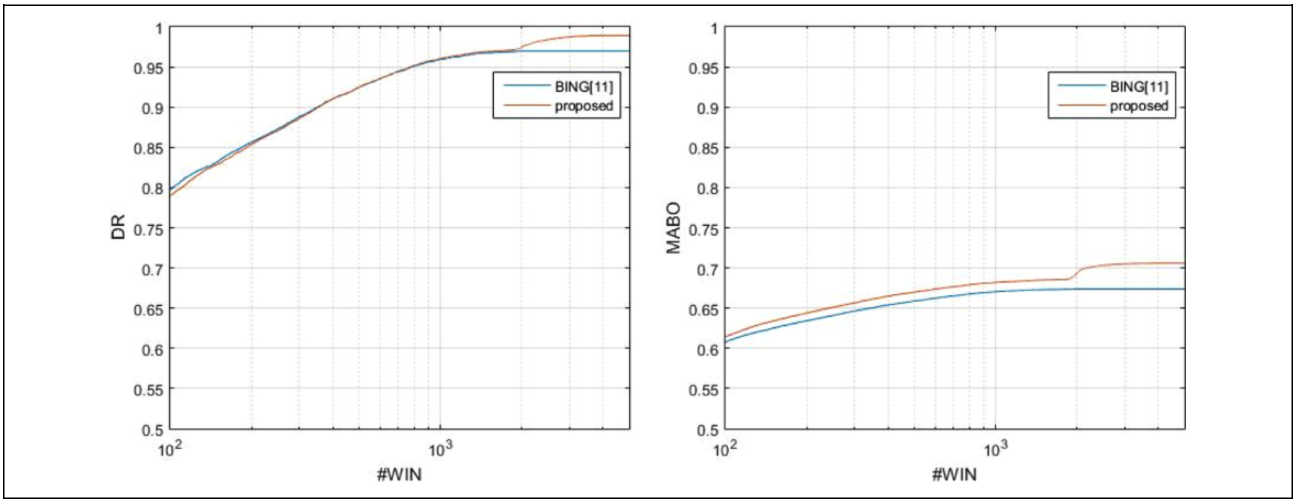

Figure 10. Comparison of our proposed method (curves in red) with the original BING (curves in blue) in DR metric and MABO metric. DR and MABO of proposed method achieve higher values than those of BING when more than 2000 windows are generated. DR: detection rate; MABO: mean average best overlap.

We follow the protocol in the studies by Uijlings et al., Desai et al., and Alexe et al.[1,6,10] for evaluation. One metric is the detection rate (DR) based on the overlapping area between predicted bounding box (r_t) and the ground truth annotation (r_g). An object is considered to be found out successfully if the overlap score $s = \frac{\text{area}(r_t \cap r_g)}{\text{area}(r_t \cup r_g)}$ is higher than the threshold 0.5. Another widely used metric is MABO, defined as the mean ABO over all classes. We choose MLBP in several extensions of LBPs mentioned in “Local binary patterns” section because it shows better behavior than others. Table 1 reflects performance comparison of different extensions of LBP. The success of our method indicates that combining gradient feature and texture feature is feasible for improving performance of objectness detection. Since the purpose of our system is to produce reliable proposals discriminating objects from amorphous backgrounds stuff, we mainly focus on performance evaluation when generating more than 500 proposals or reaching higher than 90% DR. According to the results of the experiment, our method shows more advantage when producing more than 500 proposals because that proposals quality (from combined outputs of two submodels) becomes more stable gradually. As shown in Figure 10, our method achieves a 97.6% DR and a 69.3% MABO using 2000 proposals, with 0.5% and 2% increase, respectively, compared with BING, which fully demonstrates the efficiency of our improved method. Table 2 gives the specific detection rate.

Figure 11 gives several examples to explicitly illustrate that our proposed method is better than the original BING at detecting difficult instances, especially objects with partial truncation or occlusion and objects in poor lighting conditions. BING has difficulties in finding these kinds of objects because BING relies on simple gradient feature while it is uneasy to find complete contours in occlusion situations or poor lighting conditions. Instead, our proposed method combines gradient feature and texture feature complementally to learn our model, which therefore enhances the detection rate of these hard object instances.

In addition, LBP is excellent in discriminating an object from its surroundings especially when their textures are entirely different. Thus, it is easier for our method to get accurate location of an object. Figure 12 illustrates comparison results of proposals quality between BING and our method. We can produce higher quality proposals than BING, locating objects with more compact coordinates and reaching a higher average score. Different with BING, proposals generated by our method cover most objects more accurately and thus can provide more reliable input for subsequent detectors. Besides evaluation metrics mentioned above, computation efficiency

Table 2. DR and MABO results for our method compared with the original BING.[a]

#WIN	1000	2000	3000	4000	5000
BING[14] (%)	(95.9, 67)	(97, 67.4)	(97, 67.4)	(97, 67.4)	(97, 67.4)
Our method (%)	(96.1, 68.2)	(97.6, 69.3)	(98.7, 70.5)	(98.9, 70.6)	(99, 70.7)

DR: detection rate; MABO: mean average best overlap.
[a]The values within parenthesis means (DR × 100, MABO × 100).

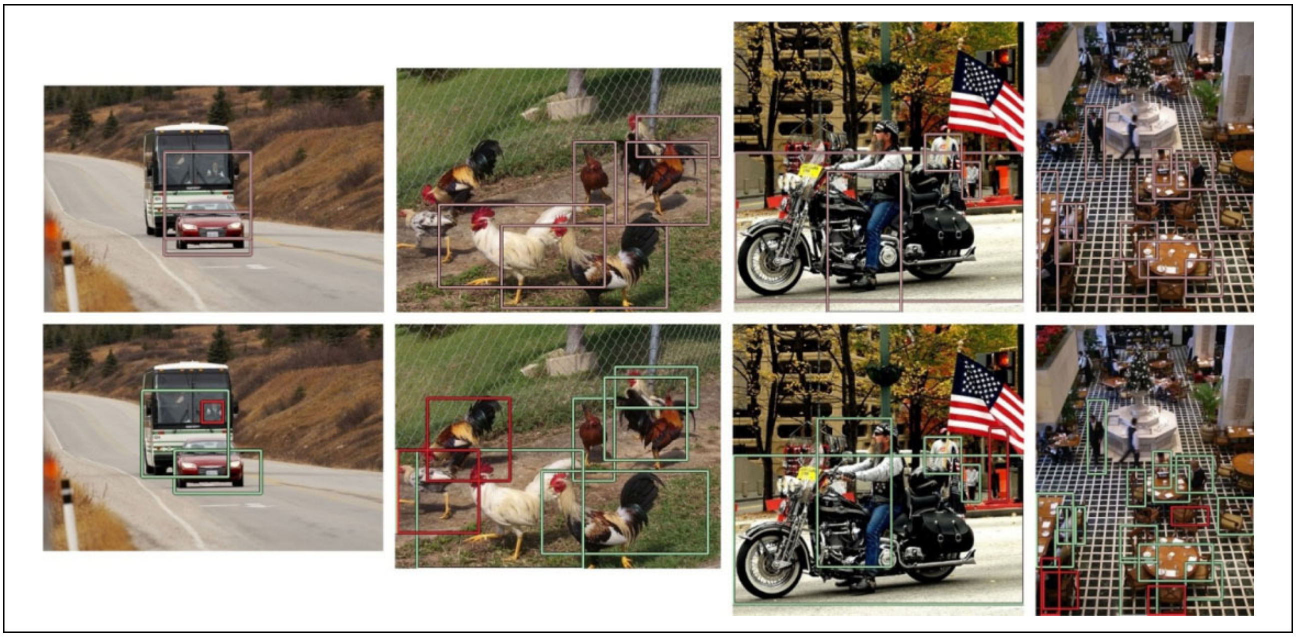

Figure 11. Examples of comparing proposal quality between BING and proposed method on VOC 2007[26] test images. The first row shows objects covered by BING's proposals (pink boxes), and the second row shows objects covered by proposed method's proposals (green and red boxes). Clearly, our method is better at detecting incomplete objects and objects in poor lighting conditions (red boxes).

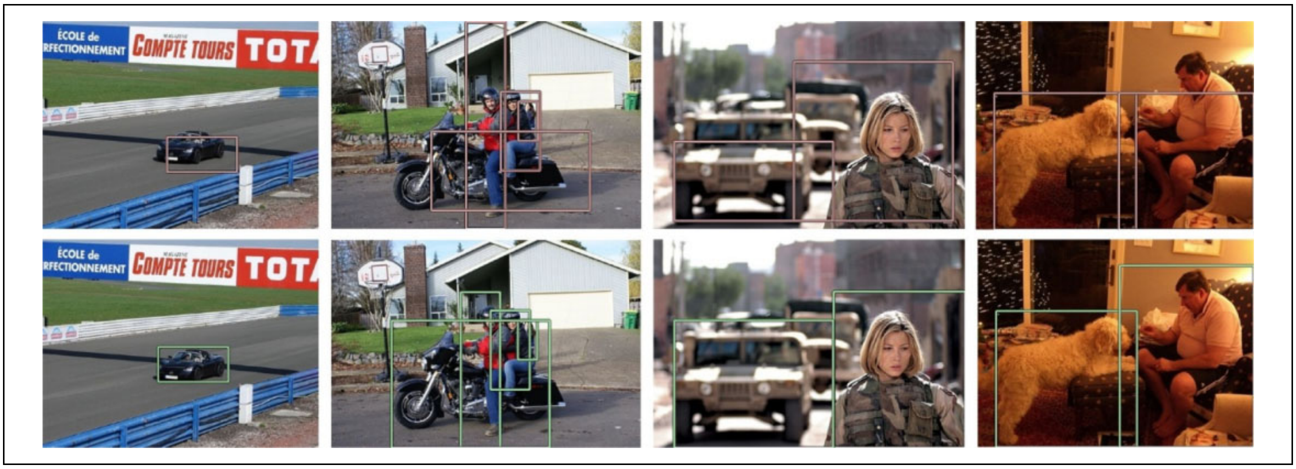

Figure 12. Comparison of proposals generated by BING and our proposed method. The first row shows objects covered by BING's proposals (pink boxes), and the second row shows objects covered by proposed method's proposals (green boxes). Proposals in the second row cover objects more compactly and more accurately.

is also indispensable for an outstanding detection system. Our proposed method is very efficient and takes only about 0.006 s per image using a laptop with an Intel Core i7-3940XM CPU@4.00GHZ. Figure 13 shows more instances of our method's detection result compared with BING on VOC 2007[26] data set.

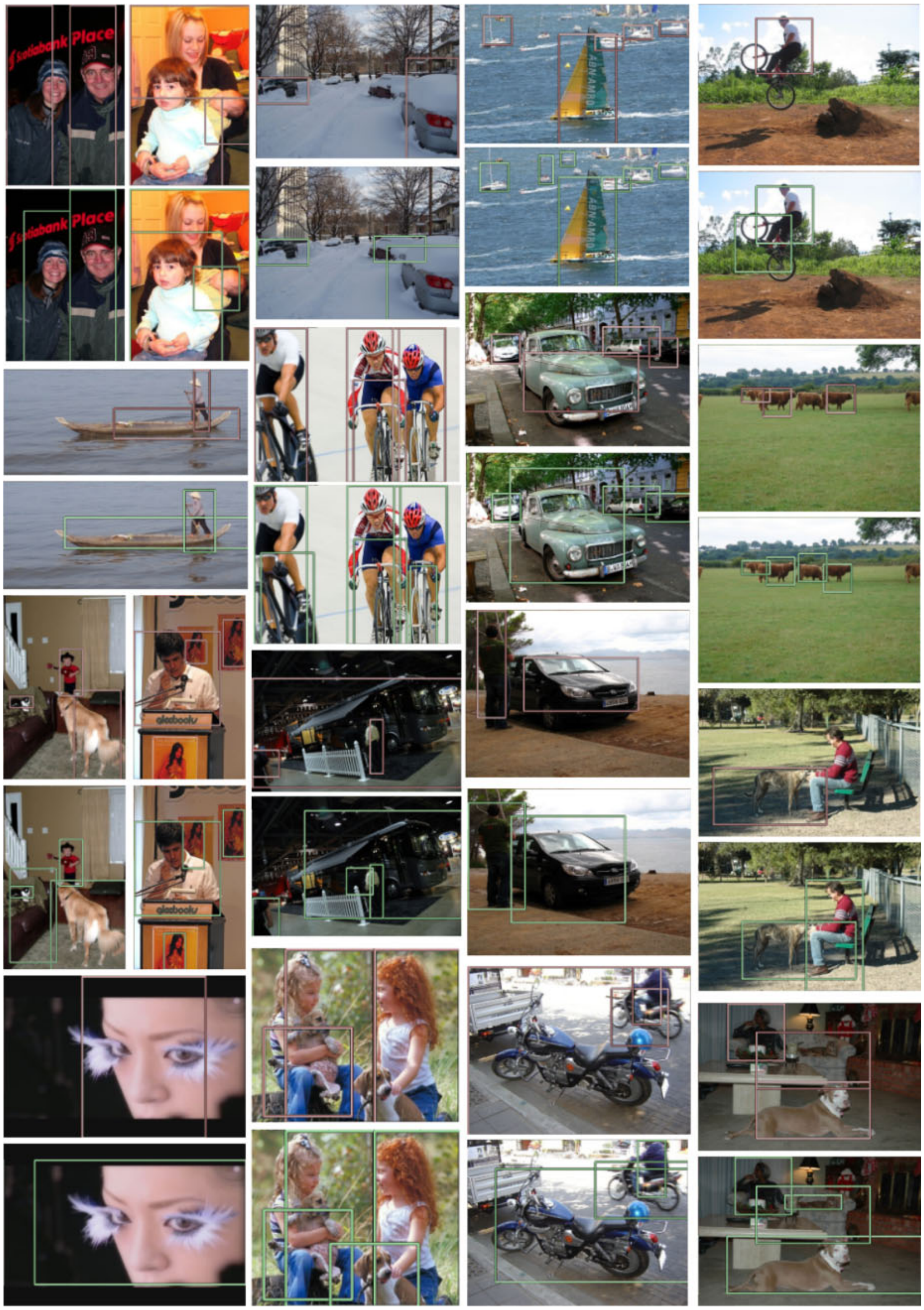

Figure 13. Detection results of some testing images on VOC 2007[26] data set. The results of BING[14] are shown in odd rows (pink boxes), and the corresponding ones of proposed method are displayed in its next row (green boxes).

Conclusion and future work

In this article, we have proposed an effective objectness estimation framework, which outputs an order set of windows covering almost all object instances. The framework we presented is mainly based on the observation that incorporating texture feature into model training would be helpful for detecting incomplete objects and objects in dim lighting. Our framework consists of two analogous cascaded submodels—one original model as in BING and another training with MLBP feature. Each submodel generates a set of proposals separately, and then they are combined to produce final output—a series high-ranking windows. We evaluate our proposed method on official PASCAL VOC 2007 data set, and the results of experiment indicate that we achieve more accurate detection results while preserving the speed advantage of BING in the meantime. By this way, we could be able to provide input of higher quality in the following object recognition stage.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

Reference

1. Uijlings JR, van de Sande KE, Gevers T, et al. Selective search for object recognition. *Int J Comput Vision* 2013; 104(2): 154–171.
2. Girshick R, Donahue J, Darrell T, et al. Rich feature hierarchies for accurate object detection and semantic segmentation. *Computer Science* (2013): 580–587.
3. Szegedy C, Reed S, Erhan D, et al. Scalable, high-quality object detection. 2014. arXiv preprint:1412.1441.
4. Felzenszwalb PF, Girshick RB, McAllester D, et al. Object detection with discriminatively trained part-based models. *IEEE Transa Pattern Anal Mach Intell* 2010; 32(9): 1627–1645.
5. Dalal N and Triggs B. Histograms of oriented gradients for human detection. In: *2005 IEEE computer society conference on computer vision and pattern recognition (CVPR'05)*, San Diego, 20–26 June 2005, vol. 1, pp. 886–893. IEEE.
6. Desai C, Ramanan D, and Folkess C. Discriminative models for multi-class object layout. *Int J Comput Vision* 2011; 95(1): 1–12.
7. Endres I and Hoiem D. Category independent object proposals. In: *European conference on computer vision*, Hersonissos, 5–11 September 2010, pp. 575–588. Berlin Heidelberg: Springer.
8. He K, Zhang X, Ren S, et al. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: *European conference on computer vision*, Zurich, 6–12 September 2014, pp. 346–361. Springer International Publishing.
9. Alexe B, Deselaers T, and Ferrari V. Measuring the objectness of image windows. *IEEE Trans Pattern Anal Mach Intell* 2012; 34(11): 2189–2202.
10. Alexe B, Deselaers T, and Ferrari V. What is an object? In: *IEEE conference on computer vision and pattern recognition (CVPR)*, San Francisco, 13–18 June 2010, pp. 73–80. IEEE.
11. Gokberk Cinbis R, Verbeek J, and Schmid C. Segmentation driven object detection with fisher vectors. In: *Proceedings of the IEEE international conference on computer vision*, Sydney, 3–6 December 2013, pp. 2968–2975.
12. Girshick R. Fast R-CNN. In: *Proceedings of the IEEE international conference on computer vision*, Santiago, 13–16 December 2015, pp. 1440–1448.
13. Hosang J, Benenson R, Dollár P, et al. What makes for effective detection proposals? *IEEE Trans Pattern Anal Mach Intell* 2016; 38(4): 814–830.
14. Cheng MM, Zhang Z, Lin WY, et al. BING: binarized normed gradients for objectness estimation at 300fps. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*, Columbus, 24–27 June 2014, pp. 3286–3293.
15. Zitnick CL and Dollár P. Edge boxes: locating object proposals from edges. In: *European conference on computer vision*, Zurich, 6–12 September 2014, pp. 391–405. Springer International Publishing.
16. Dollár P and Zitnick CL. Structured forests for fast edge detection. In: *Proceedings of the IEEE international conference on computer vision*, Sydney, 3–6 December 2013, pp. 1841–1848.
17. Dollár P and Zitnick CL. Fast edge detection using structured forests. *IEEE Trans Pattern Anal Mach Intell* 2015; 37(8): 1558–1570.
18. Zhang Z, Warrell J, and Torr PH. Proposal generation for object detection using cascaded ranking SVMs. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, Colorado Springs, 20–25 June 2011, pp. 1497–1504. IEEE.
19. Endres I and Hoiem D. Category-independent object proposals with diverse ranking. *IEEE Trans Pattern Anal Mach Intell* 2014; 36(2): 222–234.
20. Ouyang W and Wang X. Joint deep learning for pedestrian detection. In: *Proceedings of the IEEE international conference on computer vision*, Sydney, 3–6 December 2013, pp. 2056–2063
21. Liu H and Sun F. Discovery of topical objects from video: a structured dictionary learning approach. *Cognit Comput* 2016; 8(3): 519–528.
22. Liu H, Yu Y, Sun F, et al. Visual–tactile fusion for object recognition. *IEEE Trans Automat Sci Eng* 2017; 14(2): 996–1008.
23. Jiang P, Ling H, Yu J, et al. Salient region detection by UFO: uniqueness, focusness and objectness. In: *Proceedings of the IEEE international conference on computer vision*, Sydney, 3–6 December 2013, pp. 1976–1983.
24. Maji S, Berg AC, and Malik J. Classification using intersection kernel support vector machines is efficient. In: *IEEE conference on computer vision and pattern recognition 2008 CVPR*, Anchorage, 24–26 June 2008, pp. 1–8. IEEE.
25. Vedaldi A and Zisserman A. Efficient additive kernels via explicit feature maps. *IEEE Trans Pattern Anal Mach Intell* 2012; 34(3): 480–492.
26. Everingham M, Van Gool L, Williams CKI, et al. The pascal visual object classes (voc) challenge. *IJCV* 2010; 88(2): 303–338.
27. Alexe B, Deselaers T, and Ferrari V. Measuring the objectness of image windows. *IEEE Trans Pattern Anal Mach Intell* 2012; 34(11): 2189–2202.

28. Forsyth DA, Malik J, Fleck MM, et al. Finding pictures of objects in large collections of images. In: *International workshop on object representation in computer vision*, April 1996, pp. 335–360. Berlin Heidelberg: Springer.

29. Harwood D, Ojala T, Pietikäinen M, et al. Texture classification by center-symmetric auto-correlation, using Kullback discrimination of distributions. *Pattern Recognit Lett* 1995; 16(1): 1–10.

30. Guo Z, Zhang L, and Zhang D. Rotation invariant texture classification using LBP variance (LBPV) with global matching. *Pattern recognit* 2010; 43(3): 706–719.

31. Ahonen T, Hadid A, and Pietikäinen M. Face recognition with local binary patterns. In: *European conference on computer vision*, Prague, 11–14 May 2004, pp. 469–481. Berlin Heidelberg: Springer.

32. Ojala T, Pietikainen M, and Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans Pattern Anal Mach Intell* 2002; 24(7): 971–987.

33. Wang X, Han TX, and Yan S. An HOG-LBP human detector with partial occlusion handling. In: *2009 IEEE 12th international conference on computer vision*, Kyoto, 29 September–2 October 2009, pp. 32–39. IEEE.

34. Zhang J, Huang K, Yu Y, et al. Boosted local structured HOG-LBP for object localization. In: *IEEE conference on computer vision and pattern recognition (CVPR)*, Colorado Springs, 20–25 June 2011, pp. 1393–1400. IEEE.