

# BING: Binarized normed gradients for objectness estimation at 300fps

Ming-Ming Cheng<sup>1,\*</sup> (✉), Yun Liu<sup>1,\*</sup>, Wen-Yan Lin<sup>2</sup>, Ziming Zhang<sup>3</sup>, Paul L. Rosin<sup>4</sup>, and Philip H. S. Torr<sup>5</sup>

© The Author(s) 2018. This article is published with open access at Springerlink.com

**Abstract** Training a generic objectness measure to produce object proposals has recently become of significant interest. We observe that generic objects with well-defined closed boundaries can be detected by looking at the norm of gradients, with a suitable resizing of their corresponding image windows to a small fixed size. Based on this observation and computational reasons, we propose to resize the window to  $8 \times 8$  and use the norm of the gradients as a simple 64D feature to describe it, for explicitly training a generic objectness measure. We further show how the binarized version of this feature, namely binarized normed gradients (BING), can be used for efficient objectness estimation, which requires only a few atomic operations (e.g., ADD, BITWISE SHIFT, etc.). To improve localization quality of the proposals while maintaining efficiency, we propose a novel fast segmentation method and demonstrate its effectiveness for improving BING's localization performance, when used in multi-thresholding straddling expansion (MTSE) post-processing. On the challenging PASCAL VOC2007 dataset, using 1000 proposals per image and intersection-over-union threshold of 0.5, our proposal method achieves a 95.6% object detection rate and 78.6% mean average best overlap in less than 0.005 second per image.

**Keywords** object proposals; objectness; visual attention; category agnostic proposals

## 1 Introduction

As suggested in pioneering research [1, 2], *objectness* is usually taken to mean a value which reflects how likely an image window covers an object in *any category*. A generic objectness measure has great potential to be used as a pre-filter for many vision tasks, including object detection [3–5], visual tracking [6, 7], object discovery [8, 9], semantic segmentation [10, 11], content aware image retargeting [12], and action recognition [13]. Especially for object detection, proposal-based detectors have dominated recent state-of-the-art performance. Compared with sliding windows, objectness measures can significantly improve computational efficiency by reducing the search space, and system accuracy by allowing the use of complex subsequent processing during testing. However, designing a good generic objectness measure method is difficult, and should:

- achieve a *high object detection rate* (DR), as any undetected objects rejected at this stage cannot be recovered later;
- possess *high proposal localization accuracy*, measured by average best overlap (ABO) for each object in each class and mean average best overlap (MABO) across all classes;
- be *highly computationally efficient* so that it is useful in realtime and large-scale applications;
- produce a *small number of proposals*, to reduce the amount of subsequent processing;
- possess *good generalization* to unseen object categories, so that the proposals can be used in various vision tasks without category biases.

To the best of our knowledge, no prior method can satisfy all of these ambitious goals simultaneously.

Research from cognitive psychology [14, 15] and

1 CCS, Nankai University, Tianjin 300350, China. E-mail: cmm@nankai.edu.cn (✉).

2 Institute for Infocomm Research, Singapore, 138632.

3 MERL, Cambridge, MA 02139-1955, US.

4 Cardiff University, Wales, CF24 3AA, UK.

5 University of Oxford, Oxford, OX1 3PJ, UK.

\* These authors contributed equally to this work.

Manuscript received: 2018-05-08; accepted: 2018-05-26

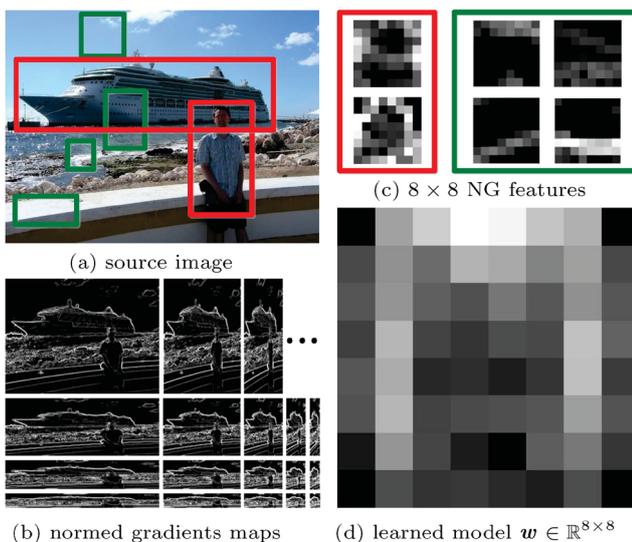
neurobiology [16, 17] suggests that humans have a strong ability to perceive objects before identifying them. Based on the observed human reaction time and the biological estimated signal transmission time, human attention theories hypothesize that the human visual system processes only parts of an image in detail, while leaving others nearly unprocessed. This further suggests that before identifying objects, simple mechanisms in the human visual system select possible object locations.

In this paper, we propose a surprisingly simple and powerful feature which we call “BING”, to help search for objects using objectness scores. Our work is motivated by the concept that objects are stand-alone things with well-defined closed boundaries and centers [2, 18, 19], even if the visibility of these boundaries depends on the characteristics of the background and of occluding foreground objects. We observe that generic objects with well-defined closed boundaries share surprisingly strong correlation in terms of the norm of their gradients (see Fig. 1 and Section 3), after resizing their corresponding image windows to a small fixed size (e.g.,  $8 \times 8$ ). Therefore, in order to efficiently quantify the objectness of an image window, we resize it to  $8 \times 8$  and use the norm of the gradients as a simple 64D feature for learning a generic objectness measure in a cascaded SVM framework. We further show how the binarized

version of the norm of gradients feature, namely binarized normed gradients (*BING*), can be used for efficient objectness estimation of image windows, using only a few atomic CPU operations (ADD, BITWISE SHIFT, etc.). The BING feature’s simplicity, while using advanced speed-up techniques to make the computational time tractable, contrasts with recent state-of-the-art techniques [2, 20, 21] which seek increasingly sophisticated features to obtain greater discrimination.

The original conference presentation of BING [22] has received much attention. Its efficiency and high detection rates make BING a good choice in a large number of successful applications that require *category independent object proposals* [23–29]. Recently, deep neural network based object proposal generation methods have become very popular due to their high recall and computational efficiency, e.g., RPN [30], YOLO900 [31], and SSD [32]. However, these methods generalize poorly to unseen categories, and rely on training with many ground-truth annotations for the target classes. For instance, the detected object proposals of RPN are highly related to the training data: after training it on the PASCAL VOC dataset [33], the trained model will aim to only detect the 20 classes of objects therein and performs poorly on other datasets like MS COCO (see Section 5.4). Its poor generalization ability has restricted its usage, so *RPN is usually only used in object detection*. In comparison, BING is based on low-level cues concerning enclosing boundaries and thus can produce category independent object proposals, which has demonstrated applications in multi-label image classification [23], semantic segmentation [25], video classification [24], co-salient object detection [29], deep multi-instance learning [26], and video summarisation [27]. However, several researchers [34–37] have noted that BING’s proposal localization is weak.

This manuscript further improves proposal localization over the method described in the conference version [22] by applying multi-thresholding straddling expansion (MTSE) [38] as a postprocessing step. Standard MTSE would introduce a significant computational bottleneck because of its image segmentation step. Therefore we propose a novel image segmentation method, which generates accurate segments much more efficiently. Our



**Fig. 1** Although object (red) and non-object (green) windows vary greatly in image space (a), at proper scales and aspect ratios which correspond to a small fixed size (b), their corresponding normed gradients (NG features) (c), share strong correlation. We learn a single 64D linear model (d) for selecting object proposals based on their NG features.

approach starts with a GPU version of the SLIC method [39, 40] to quickly obtain initial seed regions (superpixels) by performing oversegmentation. Region merging is then performed based on average pixel distances. We replace the method from Ref. [41] in MTSE with this novel grouping method [42], and dub the new proposal system BING-E.

We have extensively evaluated our objectness methods on the PASCAL VOC2007 [33] and Microsoft COCO [43] datasets. The experimental results show that our method efficiently (at 300 fps for BING and 200 fps for BING-E) generates a small set of data-driven, category-independent, and high-quality object windows. BING is able to achieve 96.2% detection rate (DR) with 1000 windows and intersection-over-union (IoU) threshold 0.5. At the increased IoU threshold of 0.7, BING-E can obtain 81.4% DR and 78.6% mean average best overlap (MABO). Feeding the proposals to the fast R-CNN framework [4] for an object detection task, BING-E achieves 67.4% mean average precision (MAP). Following Refs. [2, 20, 21], we also verify the generalization ability of our method. When training our objectness measure on the VOC2007 training set and testing on the challenging COCO validation set, our method still achieves competitive performance. Compared to most popular alternatives [2, 20, 21, 34, 36, 44–50], our method achieves competitive performance using a smaller set of proposals, while being 100–1000 times faster than them. Thus, our proposed method achieves significantly higher efficiency while providing state-of-the-art generic object proposals. This performance fulfils a key previously stated requirement for a good objectness detector. Our source code is published with the paper.

## 2 Related works

Being able to perceive objects before identifying them is closely related to bottom up visual attention (saliency). According to how saliency is defined, we broadly classify related research into three categories: fixation prediction, salient object detection, and objectness proposal generation.

### 2.1 Fixation prediction

Fixation prediction models aim to predict human eye movements [51, 52]. Inspired by neurobiological research on early primate visual systems, Itti et

al. [53] proposed one of the first computational models for saliency detection, which estimates center-surround differences across multi-scale image features. Ma and Zhang [54] proposed a fuzzy growing model to analyze local contrast based saliency. Harel et al. [55] proposed normalizing center-surrounded feature maps for highlighting conspicuous parts. Although fixation point prediction models have developed remarkably, the prediction results tend to highlight edges and corners rather than entire objects. Thus, these models are unsuitable for generating generic object proposals.

### 2.2 Salient object detection

Salient object detection models try to detect the most attention-grabbing objects in a scene, and then segment the whole extent of those objects [56–58]. Liu et al. [59] combined local, regional, and global saliency measurements in a CRF framework. Achanta et al. [60] localized salient regions using a frequency-tuned approach. Cheng et al. [61] proposed a salient object detection and segmentation method based on region contrast analysis and iterative graph based segmentation. More recent research has also tried to produce high-quality saliency maps in a filtering-based framework [62]. Such salient object segmentation has achieved great success for simple images in image scene analysis [63–65], and content aware image editing [66, 67]; it can be used as a cheap tool to process a large number of Internet images or build robust applications [68–73] by automatically selecting good results [61, 74]. However, these approaches are less likely to work for complicated images in which many objects are present but are rarely dominant (e.g., PASCAL VOC images).

### 2.3 Objectness proposal generation

These methods avoid making decisions early on, by proposing a small number (e.g., 1000) of category-independent proposals that are expected to cover all objects in an image [2, 20, 21]. Producing rough segmentations [21, 75] as object proposals has been shown to be an effective way of reducing search spaces for category-specific classifiers, whilst allowing the usage of strong classifiers to improve accuracy. However, such methods [21, 75] are very computationally expensive. Alexe et al. [2] proposed a cue integration approach to get better prediction performance more efficiently. Broadly speaking, two main categories of object proposal

generation methods exist, region based methods and edge based methods.

Region based object proposal generation methods mainly look for sets of regions produced by image segmentation and use the bounding boxes of these sets of regions to generate object proposals. Since image segmentation aims to cluster pixels into regions expected to represent objects or object-parts, merging certain regions is likely to find complete objects. A large literature has focused on this approach. Uijlings et al. [20] proposed a selective search approach, which combined the strength of both an exhaustive search and segmentation, to achieve higher prediction performance. Pont-Tuset et al. [36] proposed a multi-scale method to generate segmentation hierarchies, and then explored the combinatorial space of these hierarchical regions to produce high-quality object proposals. Other well-known algorithms [21, 45–47, 49] fall into this category as well.

Edge based object proposal generation approaches use edges to explore where in an image complete objects occur. As pointed out in Ref. [2], complete objects usually have well-defined closed boundaries in space, and various methods have achieved high performance using this intuitive cue. Zitnick and Dollár [34] proposed a simple box objectness score that measured the number of contours wholly enclosed by a bounding box, generating object bounding box proposals directly from edges in an efficient way. Lu et al. [76] proposed a closed contour measure defined by a closed path integral. Zhang et al. [44] proposed a cascaded ranking SVM approach with an oriented gradient feature for efficient proposal generation.

Generic object proposals are widely used in object detection [3–5], visual tracking [6, 7], video classification [24], pedestrian detection [28], content aware image retargeting [12], and action recognition [13]. Thus a generic objectness measure can benefit many vision tasks. In this paper, we describe a simple and intuitive object proposal generation method which generally achieves state-of-the-art detection performance, and is 100–1000 times faster than most popular alternatives [2, 20, 21] (see Section 5).

### 3 BING for objectness measure

#### 3.1 Preliminaries

Inspired by the ability of the human visual system to efficiently perceive objects before identifying them

[14–17], we introduce a simple 64D norm-of-gradients (NG) feature (Section 3.2), as well as its binary approximation, i.e., the binarized normed gradients (BING) feature (Section 3.4), for efficiently capturing the objectness of an image window.

To find generic objects within an image, we scan over a predefined set of *quantized window sizes* (scales and aspect ratios<sup>Ⓛ</sup>). Each window is scored with a linear model  $\mathbf{w} \in \mathbb{R}^{64}$  (Section 3.3):

$$s_l = \langle \mathbf{w}, \mathbf{g}_l \rangle \quad (1)$$

$$l = (i, x, y) \quad (2)$$

where  $s_l$ ,  $\mathbf{g}_l$ ,  $l$ ,  $i$ , and  $(x, y)$  are filter score, NG feature, location, size, and position of a window, respectively. Using non-maximal suppression (NMS), we select a small set of proposals from each size  $i$ . Zhao et al. [37] showed that this choice of window sizes along with the NMS is close to optimal. Some sizes (e.g.,  $10 \times 500$ ) are less likely than others (e.g.,  $100 \times 100$ ) to contain an object instance. Thus we define the objectness score (i.e., the calibrated filter score) as

$$o_l = v_i \cdot s_l + t_i \quad (3)$$

where  $v_i, t_i \in \mathbb{R}$  are learnt coefficient and bias terms for each quantized size  $i$  (Section 3.3). Note that calibration using Eq. (3), although very fast, is only required when re-ranking the small set of final proposals.

#### 3.2 Normed gradients (NG) and objectness

Objects are stand-alone things with well-defined closed boundaries and centers [2, 18, 19] although the visibility of these boundaries depends on the characteristics of the background and occluding foreground objects. When resizing windows corresponding to real world objects to a small fixed size (e.g.,  $8 \times 8$ , chosen for computational reasons that will be explained in Section 3.4), the norms (i.e., magnitude) of the corresponding image gradients become good discriminative features, because of the limited variation that closed boundaries could present in such an abstracted view. As demonstrated in Fig. 1, although the cruise ship and the person have huge differences in terms of color, shape, texture, illumination, etc., they share clear similarity in normed gradient space. To utilize this observation

<sup>Ⓛ</sup> In all experiments, we test 36 quantized target window sizes  $\{(W_o, H_o)\}$ , where  $W_o, H_o \in \{16, 32, 64, 128, 256, 512\}$ . We resize the input image to 36 sizes so that  $8 \times 8$  windows in the downsized images (from which we extract features), correspond to target windows.

to efficiently predict the existence of object instances, we firstly resize the input image to different *quantized sizes* and calculate the normed gradients of each resized image. The values in an  $8 \times 8$  region of these resized normed gradients maps are defined as a 64D vector of *normed gradients* ( $NG$ )<sup>ⓐ</sup> feature of its corresponding window.

Our NG feature, as a dense and compact objectness feature for an image window, has several advantages. Firstly, no matter how an object changes its position, scale, and aspect ratio, its corresponding NG feature will remain roughly unchanged because the region for computing the feature is normalized. In other words, NG features are insensitive to change of translation, scale, and aspect ratio, which will be very useful for detecting objects of arbitrary categories. Such insensitivity in a property is one that a good objectness proposal generation method should have. Secondly, the dense compact representation of the NG feature makes it allow to be very efficiently calculated and verified, with great potential for realtime applications.

The cost of introducing such advantages to the NG feature is loss of discriminative ability. However, this is not a problem as BING can be used as a pre-filter, and the resulting false-positives can be processed and eliminated by subsequent category specific detectors. In Section 5, we show that our method results in a small set of high-quality proposals that cover 96.2% of the true object windows in the challenging VOC2007 dataset.

### 3.3 Learning objectness measurement with NG

To learn an objectness measure for image windows, we follow the two stage cascaded SVM approach [44].

*Stage I.* We learn a single model  $w$  for Eq. (1) using a linear SVM [77]. NG features of ground truth object windows and randomly sampled background windows are used as positive and negative training samples respectively.

*Stage II.* To learn  $v_i$  and  $t_i$  in Eq. (3) using a linear SVM [77], we evaluate Eq. (1) at size  $i$  for training images and use the selected (NMS) proposals as training samples, their filter scores as 1D features, and check their labeling using training image annotations (see Section 5 for evaluation criteria).

<sup>ⓐ</sup> The *normed gradient* represents Euclidean norm of the gradient.

As can be seen in Fig. 1(d), the learned linear model  $w$  (see Section 5 for experimental settings) looks similar to the multi-size center-surrounded patterns [53] hypothesized as a biologically plausible architecture in primates [15, 16, 78]. The large weights along the borders of  $w$  favor a boundary that separates an object (center) from its background (surround). Compared to manually designed center-surround patterns [53], our learned  $w$  captures a more sophisticated natural prior. For example, lower object regions are more often occluded than upper parts. This is represented by  $w$  placing less confidence in the lower regions.

### 3.4 Binarized normed gradients (BING)

To make use of recent advantages in binary model approximation [79, 80], we describe an accelerated version of the NG feature, namely binarized normed gradients (BING), to speed up the feature extraction and testing process. Our learned linear model  $w \in \mathbb{R}^{64}$  can be approximated by a set of basis vectors  $w \approx \sum_{j=1}^{N_w} \beta_j \mathbf{a}_j$  using Algorithm 1, where  $N_w$  denotes the number of basis vectors,  $\mathbf{a}_j \in \{-1, 1\}^{64}$  denotes a single basis vector, and  $\beta_j \in \mathbb{R}$  denotes its corresponding coefficient. By further representing each  $\mathbf{a}_j$  using a binary vector and its complement:  $\mathbf{a}_j = \mathbf{a}_j^+ - \mathbf{a}_j^-$ , where  $\mathbf{a}_j^+ \in \{0, 1\}^{64}$ , a binarized feature  $\mathbf{b}$  can be tested using fast BITWISE AND and BIT COUNT operations (see Ref. [79]):

$$\langle w, \mathbf{b} \rangle \approx \sum_{j=1}^{N_w} \beta_j (2\langle \mathbf{a}_j^+, \mathbf{b} \rangle - |\mathbf{b}|) \quad (4)$$

The key challenge is how to binarize and calculate NG features efficiently. We approximate the normed gradient values (each saved as a BYTE value) using the top  $N_g$  binary bits of the BYTE values. Thus, a 64D NG feature  $\mathbf{g}_l$  can be approximated by  $N_g$  binarized normed gradients (BING) features as

$$\mathbf{g}_l = \sum_{k=1}^{N_g} 2^{8-k} \mathbf{b}_{k,l} \quad (5)$$

Notice that these BING features have different

---

#### Algorithm 1 Binary approximate model $w$ [79]

---

*Input:*  $w, N_w$

*Output:*  $\{\beta_j\}_{j=1}^{N_w}, \{\mathbf{a}_j\}_{j=1}^{N_w}$

*Initialize residual:*  $\varepsilon = w$

**for**  $j = 1$  to  $N_w$  **do**

$\mathbf{a}_j = \text{sign}(\varepsilon)$

$\beta_j = \langle \mathbf{a}_j, \varepsilon \rangle / \|\mathbf{a}_j\|^2$  (project  $\varepsilon$  onto  $\mathbf{a}_j$ )

$\varepsilon \leftarrow \varepsilon - \beta_j \mathbf{a}_j$  (update residual)

**end for**

---

weights according to their corresponding bit position in the BYTE values.

Naively determining an  $8 \times 8$  BING feature requires a loop computing access to 64 positions. By exploring two special characteristics of an  $8 \times 8$  BING feature, we develop a fast BING feature calculation algorithm (Algorithm 2), which enables using atomic updates (BITWISE SHIFT and BITWISE OR) to avoid computing the loop. Firstly, a BING feature  $\mathbf{b}_{x,y}$  and its last row  $\mathbf{r}_{x,y}$  are saved in a single INT64 and a BYTE variable, respectively. Secondly, adjacent BING features and their rows have a simple cumulative relation. As shown in Fig. 2 and Algorithm 2, the operator BITWISE SHIFT shifts  $\mathbf{r}_{x-1,y}$  by one bit, automatically through the bit which does not belong to  $\mathbf{r}_{x,y}$ , and makes room to insert the new bit  $b_{x,y}$  using the BITWISE OR operator. Similarly BITWISE SHIFT shifts  $\mathbf{b}_{x,y-1}$  by 8 bits automatically through the bits which do not belong to  $\mathbf{b}_{x,y}$ , and makes room to insert  $\mathbf{r}_{x,y}$ .

Our efficient BING feature calculation shares its *cumulative* nature with the integral image representation [81]. Instead of calculating a single scalar value over an arbitrary rectangle range [81], our method uses a few atomic operations (e.g., ADD, BITWISE, etc.) to calculate *a set of binary patterns* over an  $8 \times 8$  fixed range.

---

**Algorithm 2** Get BING features for  $W \times H$  positions

---

*Comments:* see Fig. 2 for an explanation of variables

*Input:* binary normed gradient map  $b_{W \times H}$

*Output:* BING feature matrix  $\mathbf{b}_{W \times H}$

*Initialize:*  $\mathbf{b}_{W \times H} = 0, \mathbf{r}_{W \times H} = 0$

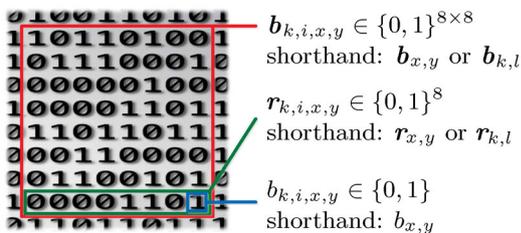
for each position  $(x, y)$  in scan-line order **do**

$\mathbf{r}_{x,y} = (\mathbf{r}_{x-1,y} \ll 1) \mid b_{x,y}$

$\mathbf{b}_{x,y} = (\mathbf{b}_{x,y-1} \ll 8) \mid \mathbf{r}_{x,y}$

**end for**

---



**Fig. 2** Variables: a BING feature  $\mathbf{b}_{x,y}$ , its last row  $\mathbf{r}_{x,y}$ , and last element  $b_{x,y}$ . Notice that the subscripts  $i, x, y, l, k$ , introduced in Eq. (2) and Eq. (5), are locations of the whole vector rather than the indices of vector elements. We can use a single atomic variable (INT64 and BYTE) to represent a BING feature and its last row, respectively, enabling efficient feature computation.

The filter score Eq. (1) for an image window corresponding to BING features  $\mathbf{b}_{k,l}$  can be efficiently computed using:

$$s_l \approx \sum_{j=1}^{N_w} \beta_j \sum_{k=1}^{N_g} C_{j,k} \quad (6)$$

where  $C_{j,k} = 2^{8-k} (2 \langle \mathbf{a}_j^+, \mathbf{b}_{k,l} \rangle - |\mathbf{b}_{k,l}|)$  can be tested using fast BITWISE and POPCNT SSE operators.

To implement these ideas, we use the 1-D kernel  $[-1, 0, 1]$  to find image gradients  $g_x$  and  $g_y$  in the horizontal and vertical directions, calculate normed gradients using  $\min(|g_x| + |g_y|, 255)$  and save them in BYTE values. By default, we calculate gradients in RGB color space.

## 4 Enhancing BING with region cues

BING is not only very efficient, but can also achieve a high object detection rate. However, in comparison to ABO or MABO, its performance is disappointing. When further applying BING in some object detection frameworks which use object proposals as input, like fast R-CNN, the detection rate is also poor. This suggests that BING does not provide good proposal localization.

Two reasons may cause this. On one hand, given an object, BING tries to capture its closed boundaries by resizing it to a small fixed size and setting larger weights at the most probable positions. However, as shapes of objects are varied, the closed boundaries of objects will be mapped to different positions in the fixed size windows. The learned model of NG features cannot adequately represent this variability across objects. On the other hand, BING is designed to only test a limited set of *quantized window sizes*. However, the sizes of objects are variable. Thus, to some extent, bounding boxes generated by BING are unable to tightly cover all objects.

In order to improve this unsatisfactory localization, we use multi-thresholding straddling expansion (MTSE) [38], which is an effective method to refine object proposals using segments. Given an image and corresponding initial bounding boxes, MTSE first aligns boxes with potential object boundaries preserved by superpixels, and then multi-thresholding expansion is performed with respect to superpixels straddling each box. In this way, each bounding box tightly covers a set of internal superpixels, significantly improving the localization quality of proposals. However, the MTSE algorithm is too slow;

the bottleneck is segmentation [41]. Thus, we use a new fast image segmentation method [42] to replace the segmentation method in MTSE.

Recently, SLIC [40] has become a popular superpixel generation method because of its efficiency; gSLICr, the GPU version of SLIC [39], can achieve a speed of 250 fps. SLIC aims to generate small superpixels and is not good at producing large image segments. In the MTSE algorithm, large image segments are needed to ensure accuracy, so it is not straightforward to use SLIC within MTSE. However, the high efficiency of SLIC makes it a good start for developing new segmentation methods. We first use gSLICr to segment an image into many small superpixels. Then, we view each superpixel as a node whose color is denoted by the average color of all its pixels, and the distance between two adjacent nodes is computed as the Euclidean distance of their color values. Finally, we feed these nodes into a graph-based segmentation method to produce the final image segmentation [42].

We employ the full MTSE pipeline, and modify it to use our new segmentation algorithm, reducing the computation time from 0.15 s down to 0.0014 s per image. Incorporating this improved version of MTSE as a postprocessing enhancement step for BING gives our new proposal system, which we call BING-E.

## 5 Evaluation

### 5.1 Background

We have extensively evaluated our method on the challenging PASCAL VOC2007 [33] and Microsoft COCO [43] datasets. PASCAL VOC2007 contains 20 object categories, and consists of training, validation, and test sets, with 2501, 2510, and 4952 images respectively, having corresponding bounding box annotations. We use the training set to train our BING model and test on the test set. Microsoft COCO consists of 82,783 images for training and 40,504 images for validation, with about 1 million annotated instances in 80 categories. COCO is more challenging because of its large size and complex image contents.

We compared our method to various competitive methods: EdgeBoxes [34]<sup>①</sup>, CSVM [44]<sup>②</sup>, MCG

[36]<sup>③</sup>, RPN [30]<sup>④</sup>, Endres [21], Objectness [2], GOP [48], LPO [49], Rahtu [45], RandomPrim [46], Rantalankila [47], and SelectiveSearch [20], using publicly available code [82] downloaded from <https://github.com/Cloud-CV/object-proposals>. All parameters for these methods were set to default values, except for Ref. [48], in which we employed (180,9) as suggested on the author's homepage. To make the comparison fair, all methods except for the deep learning based RPN [30] were tested on the same device with an Intel i7-6700k CPU and NVIDIA GeForce GTX 970 GPU, with data parallelization enabled. For RPN, we utilized an NVIDIA GeForce GTX TITAN X GPU for computation.

Since objectness is often used as a preprocessing step to reduce the number of windows considered in subsequent processing, too many proposals are unhelpful. Therefore, we only used the top 1000 proposals for comparison.

In order to evaluate the generalization ability of each method, we tested them on the COCO validation dataset using the same parameters as for VOC2007 without retraining. Since at least 60 categories in COCO differ from those in VOC2007, COCO is a good test of the generalization ability of the methods.

### 5.2 Experimental setup

#### 5.2.1 Discussion of BING

As shown in Table 1, by using the binary approximation to the learned linear filter (Section 3.4) and BING features, computing the response score for each image window only needs a fixed small number of atomic operations. It is easy to see that the number of positions at each quantized scale and aspect ratio is  $O(N)$ , where  $N$  is the number of pixels in the image. Thus, computing response scores at all scales and aspect ratios also has computational complexity  $O(N)$ . Furthermore, extracting the BING feature and computing the response score at each potential position (i.e., an image window) can be calculated with information given by its 2 neighbors left and above. This means that the space complexity is also  $O(N)$ .

For training, we flip the images and the corresponding annotations. The positive samples are boxes that have IoU overlap with a ground truth box

<sup>①</sup> <https://github.com/pdollar/edges>  
<sup>②</sup> <https://zimingzhang.wordpress.com/>

<sup>③</sup> <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/mcg/>  
<sup>④</sup> <https://github.com/rbgirshick/py-faster-rcnn>

**Table 1** Average number of atomic operations for computing objectness of each image window at different stages: calculate normed gradients, extract BING features, and get objectness score

	BITWISE			FLOAT		INT,BYTE	
	SHIFT	, &	CNT	+	×	+, -	min
Gradient	0	0	0	0	0	9	2
Get BING	12	12	0	0	0	0	0
Get score	0	8	12	1	2	8	0

of at least 0.5, while the maximum IoU overlap with the ground truth for the negative sampling boxes is less than 0.5.

Some window sizes whose aspect ratios are too large are ignored as there are too few training samples (less than 50) in VOC2007 for each of them. Our training on 2501 VOC2007 images takes only 20 seconds (excluding XML loading time).

We further illustrate in Table 2 how different approximation levels influence the result quality. From this comparison, we decided in all further experiments to use  $N_w = 2$ ,  $N_g = 4$ .

### 5.2.2 Implementation of BING-E

In BING-E, removing some small BING windows, with  $W_o < 30$  or  $H_o < 30$ , hardly degrades the proposal quality of BING-E while reducing the runtime spent on BING processing by half. When using gSLICr [39] to segment images into superpixels, we set the expected size of superpixels to  $4 \times 4$ . In the graph-based segmentation system [41, 42], we use the scale parameter  $k = 120$ , and the minimum number of superpixels in each produced segment is set to 6. We utilize the default multi-thresholds of MTSE:  $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ . After refinement, non-maximal suppression (NMS) is performed to obtain the final boxes, with an IoU threshold of NMS set to 0.8. All experiments used these settings.

## 5.3 PASCAL VOC2007

### 5.3.1 Results

As demonstrated by Refs. [2, 20], a small set of coarse locations with high detection recall (DR) is sufficient for effective object detection, and it allows expensive features and complementary cues to be involved in subsequent detection to achieve better quality and

**Table 2** Average result quality (DR using 1000 proposals) of BING at different approximation levels, measured by  $N_w$  and  $N_g$  in Section 3.4. N/A represents unbinned

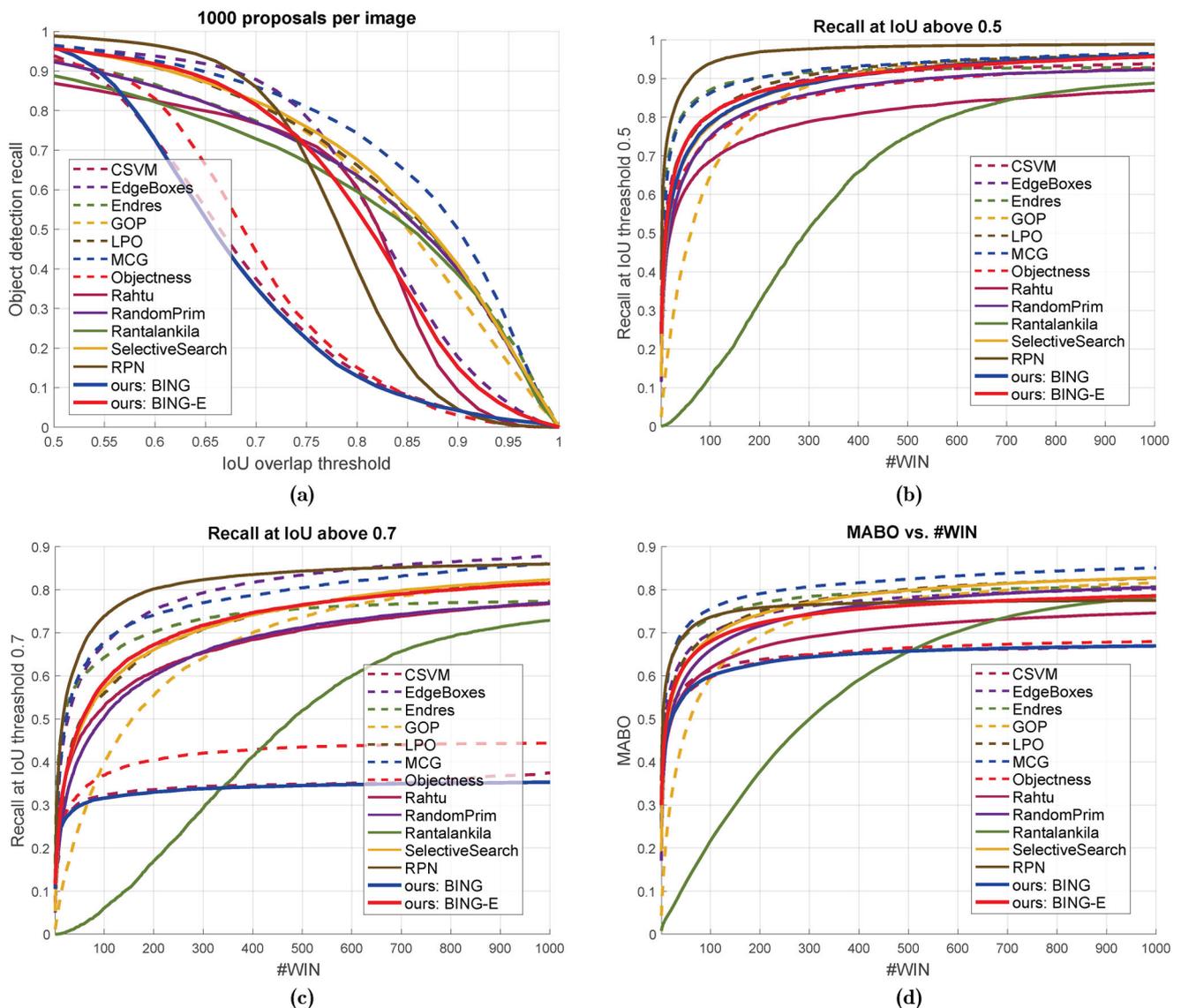
$(N_w, N_g)$	(2,3)	(2,4)	(3,2)	(3,3)	(3,4)	N/A
DR (%)	95.9	96.2	95.8	96.2	96.1	96.3

higher efficiency than traditional methods. Thus, we first compare our method with some competitors using detection recall metrics. Figure 3(a) shows detection recall when varying the IoU overlap threshold using 1000 proposals. EdgeBoxes and MCG outperform many other methods in all cases. RPN achieves very high performance when the IoU threshold is less than 0.7, but then drops rapidly. Note that RPN is the only deep learning based method amongst these competitors. BING's performance is not competitive when the IoU threshold increases, but BING-E provides close to the best performance. It should be emphasized that both BING and BING-E are *more than 100* times faster than most popular alternatives [20, 21, 34, 36] (see details in Table 3). The performance of BING and CSVM [44] almost coincide in all three subfigures, but BING is 100 times faster than CSVM. The significant improvement from BING to BING-E illustrates that BING is a strong basis that can be extended and improved in various ways. Since BING is able to run at about *300 fps*, its variants can still be very fast. For example, BING-E can generate competitive candidates at *over 200 fps*, which is far beyond the performance of most other detection algorithms.

Figures 3(b)–3(d) show detection recall and MABO versus the number of proposals (#WIN) respectively. When the IoU threshold is 0.5, both BING and BING-E perform very well; when the

**Table 3** Detection recall (%) using different IoU thresholds and #WIN on the VOC2007 test set

Method	IoU=0.5			IoU=0.7			Time(s)
	100	500	1000	100	500	1000	
CSVM	80.6	92.0	93.9	32.3	34.8	37.5	0.33
EdgeBoxes	80.4	93.1	96.1	67.3	83.4	<b>87.8</b>	0.25
Endres	87.1	92.4	92.8	64.3	75.7	77.4	19.94
GOP	64.7	93.0	96.0	39.7	73.7	82.3	0.29
LPO	80.4	93.8	96.0	56.0	76.3	81.8	0.46
MCG	86.2	94.0	96.5	67.9	80.4	86.1	17.46
Objectness	74.5	89.1	92.7	36.9	43.5	44.4	0.91
Rahtu	68.6	82.5	86.9	52.9	70.7	76.8	0.67
RandomPrim	74.9	89.5	92.3	50.4	71.2	76.9	0.12
Rantalankila	12.9	75.1	88.8	6.0	51.9	72.9	3.57
SelectiveSearch	77.8	92.4	95.7	57.1	76.2	82.3	1.60
RPN	<b>93.9</b>	<b>98.4</b>	<b>98.8</b>	<b>73.9</b>	<b>84.3</b>	86.0	0.10
BING	78.3	92.4	96.2	31.6	34.5	35.3	<b>0.0033</b>
BING+MTSE	81.2	93.6	96.3	56.5	77.7	83.4	0.022
BING-E	80.6	92.4	95.6	58.5	76.5	81.4	0.0047



**Fig. 3** Testing results on PASCAL VOC2007 test set: (a) object detection recall versus IoU overlap threshold; (b, c) recall versus the number of candidates at IoU threshold 0.5 and 0.7 respectively; (d) MABO versus the number of candidates using at most 1000 proposals.

number of candidates is sufficient, BING and BING-E outperform all other methods. In Fig. 3(c), the recall curve of BING drops significantly, as it does in the MABO evaluation. This may be because the proposal localization quality of BING is poor. However, the performance of BING-E is consistently close to the best performance, indicating that it overcomes BING's localization problem.

We show a numerical comparison of recall vs. #WIN in Table 3. BING-E always performs better than most competitors. The speeds of BING and BING-E are obviously faster than all of the other methods. Although EdgeBoxes, MCG, and SelectiveSearch perform very well, they are too

slow for many applications. In contrast, BING-E is more attractive. It is also interesting to find that the detection recall of BING-E increases by 46.1% over BING using 1000 proposals with IoU threshold 0.7, which suggests that the accuracy of BING has lots of room for improvement by applying postprocessing. Table 4 compares ABO & MABO scores with the competitors. MCG always outperforms others by a big gap, but BING-E is competitive with all other methods.

Since proposal generation is usually a preprocessing step in vision tasks, we fed candidate boxes produced by objectness methods into the fast R-CNN [4] object detection framework to test the effectiveness

**Table 4** ABO & MABO (%) using at most 1000 proposals per image on the VOC2007 test set

Method																					MABO
CSVM	67.9	66.9	62.8	62.8	58.2	68.8	64.4	69.5	62.0	65.0	69.6	68.1	67.5	66.6	62.4	59.6	63.9	69.9	69.0	63.1	67.0
EdgeBoxes	77.0	81.4	78.5	76.8	66.1	83.8	76.9	82.4	76.3	82.2	80.8	83.4	81.3	80.9	73.6	71.9	80.8	82.6	80.0	81.5	80.2
Endres	71.0	80.8	73.8	66.8	60.8	84.9	79.4	89.0	72.8	79.2	86.9	87.4	83.0	82.4	70.7	68.4	76.1	89.6	84.8	78.9	80.7
GOP	74.2	80.5	76.1	73.5	64.2	86.3	80.6	88.0	76.4	82.1	86.3	85.9	79.8	79.6	73.7	71.2	78.6	88.1	82.5	83.3	81.6
LPO	76.4	80.4	77.4	73.4	61.0	87.2	81.3	89.5	74.9	82.7	84.9	87.5	82.3	82.4	73.3	71.5	79.8	89.0	84.5	81.6	82.6
MCG	81.4	<b>83.2</b>	79.3	76.2	70.0	<b>88.1</b>	<b>81.6</b>	<b>89.9</b>	<b>79.6</b>	<b>84.6</b>	<b>88.6</b>	88.5	<b>84.4</b>	<b>83.2</b>	<b>78.2</b>	<b>74.6</b>	<b>82.8</b>	<b>91.0</b>	<b>86.6</b>	<b>85.8</b>	<b>85.1</b>
Objectness	65.1	66.5	63.8	63.0	56.1	69.4	63.3	72.4	62.6	65.0	72.8	70.9	69.2	66.9	62.3	60.1	63.7	72.3	70.7	63.1	68.0
Rahtu	72.9	73.6	67.6	70.4	46.8	78.8	67.6	80.7	61.5	71.9	79.9	79.7	78.3	73.3	64.9	58.0	68.1	80.2	80.6	73.1	74.6
RandomPrim	79.2	80.9	74.5	74.7	59.4	83.4	76.4	86.9	74.4	78.5	87.6	85.6	80.3	80.8	70.5	66.5	72.3	89.1	82.5	79.6	80.5
Rantalankila	73.0	74.4	72.7	68.0	53.9	80.4	72.2	88.9	68.1	75.6	82.1	85.9	80.1	75.6	65.4	62.4	72.9	86.6	81.6	76.6	78.3
SelectiveSearch	<b>81.8</b>	82.4	<b>79.8</b>	<b>77.5</b>	62.8	84.0	78.0	89.8	76.5	82.9	87.1	<b>89.1</b>	82.0	81.8	72.9	70.9	79.9	89.3	84.0	82.8	82.8
RPN	71.6	78.5	75.1	72.9	<b>70.7</b>	76.8	77.0	78.6	76.1	78.7	79.0	78.9	78.1	77.1	76.4	72.3	76.6	78.1	77.1	77.0	77.5
<i>ours: BING</i>	65.1	65.7	63.7	62.5	60.8	65.8	64.1	70.6	63.2	65.3	69.4	67.8	65.8	65.8	63.8	62.6	63.9	68.7	68.6	63.4	66.9
<i>ours: BING-E</i>	76.7	78.2	75.3	74.2	63.6	81.8	74.3	82.9	74.7	77.9	82.7	82.1	77.8	77.4	72.0	70.7	75.9	84.0	79.5	78.7	78.6

of proposals in practical applications. The CNN model of fast R-CNN was retrained using boxes from the respective methods. Table 5 shows the evaluation results. In terms of MAP (mean average precision), the overall detection rates of all methods are quite similar. RPN performs slightly better, while our BING-E method gives very close to the best performance. Although MCG almost dominates the recall, ABO, and MABO metrics, it does not achieve the best performance on object detection, and is worse than BING-E. In summary we may

say that BING-E provides state-of-the-art generic object proposals at a much higher speed than other methods. Finally, we illustrate sample results of varying complexity provided by our improved BING-E method for VOC2007 test images in Fig. 5, to demonstrate our high-quality proposals.

### 5.3.2 Discussion

In order to perform further analysis, we divided the ground truths into different sets according to their window sizes, and tested some of the most competitive methods on these sets. Table 6 shows the

**Table 5** Detection average precision (%) using fast R-CNN on the VOC2007 test set with 1000 proposals

Method																					mAP
CSVM	68.0	71.3	60.3	44.1	33.7	73.0	69.1	77.1	28.7	68.1	58.7	71.5	78.3	69.5	60.7	25.6	57.4	61.4	72.5	55.7	60.2
EdgeBoxes	<b>73.4</b>	78.1	68.4	<b>55.7</b>	39.2	79.5	76.8	81.0	41.7	73.7	65.6	82.8	82.6	76.2	68.1	34.8	66.2	70.1	77.1	58.9	67.5
Endres	63.3	75.0	63.4	43.0	31.2	77.2	70.5	78.1	32.8	66.8	67.6	75.3	78.7	70.9	61.1	28.0	61.6	66.3	75.9	61.3	62.4
GOP	67.2	76.3	65.7	51.5	32.4	78.4	78.6	81.1	40.7	74.1	64.2	78.7	80.5	74.3	67.3	30.7	65.4	<b>70.6</b>	76.5	66.1	66.0
LPO	67.4	76.9	<b>68.8</b>	52.1	30.4	81.3	75.0	79.9	37.9	73.9	67.6	76.4	80.3	70.1	66.1	33.5	65.0	68.0	76.4	63.9	65.6
MCG	69.8	77.2	67.2	51.8	42.5	80.0	76.8	78.6	43.9	71.4	68.1	77.1	81.5	70.9	67.8	33.0	65.5	68.2	77.1	64.8	66.7
Objectness	64.7	73.5	60.4	40.1	34.8	72.7	69.5	76.8	31.5	67.4	59.0	77.7	79.1	71.4	60.8	30.5	54.6	62.0	73.5	57.5	60.9
Rahtu	69.2	68.6	59.1	53.8	23.1	78.4	67.2	79.9	26.9	66.6	68.5	76.7	79.7	70.3	58.0	26.9	57.1	64.2	77.2	60.5	61.6
RandomPrim	69.8	78.4	61.5	52.6	25.3	76.0	69.3	78.3	39.2	67.5	<b>69.8</b>	76.2	82.7	69.5	58.8	27.6	53.7	67.5	76.3	58.5	62.9
Rantalankila	68.0	67.7	63.1	42.3	21.5	71.5	64.5	78.7	29.8	69.2	67.6	74.3	77.1	66.9	54.7	25.2	60.6	63.8	75.9	59.9	60.1
SelectiveSearch	72.9	78.3	66.0	54.3	34.7	81.3	76.8	83.3	41.5	<b>74.5</b>	66.4	79.8	82.2	76.2	65.5	35.2	65.6	70.1	<b>77.4</b>	65.9	67.4
RPN	67.5	<b>78.5</b>	67.3	51.9	<b>51.5</b>	76.2	<b>79.8</b>	<b>84.4</b>	<b>50.2</b>	74.3	66.9	<b>83.2</b>	80.0	73.9	<b>76.5</b>	<b>37.1</b>	<b>69.4</b>	65.7	76.5	<b>74.2</b>	<b>69.2</b>
<i>ours: BING</i>	65.0	68.6	61.8	46.8	42.2	72.1	71.4	77.7	31.4	69.7	56.3	74.0	75.7	66.3	65.4	27.1	62.1	60.6	68.7	60.0	61.2
<i>ours: BING-E</i>	69.3	78.3	66.5	55.0	39.0	<b>81.7</b>	75.9	83.9	39.6	74.4	67.5	80.1	<b>83.7</b>	<b>76.3</b>	67.0	35.2	67.2	68.8	75.8	61.7	67.4

**Table 6** Recall/MABO (%) vs. area on VOC2007 test set with 1000 proposals and IoU threshold 0.5

Method \ Area		2 <sup>8</sup>	2 <sup>9</sup>	2 <sup>10</sup>	2 <sup>11</sup>	2 <sup>12</sup>	2 <sup>13</sup>	2 <sup>14</sup>	2 <sup>15</sup>	2 <sup>16</sup>	2 <sup>17</sup>	2 <sup>18</sup>
Recall	EdgeBoxes(Recall)	2.1	32.6	56.2	74.0	89.1	97.3	99.5	99.8	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
	MCG	<b>43.8</b>	<b>57.1</b>	<b>73.5</b>	<b>81.9</b>	89.9	95.5	98.0	99.6	99.7	<b>100.0</b>	<b>100.0</b>
	SelectiveSearch	6.3	28.8	58.7	75.2	87.2	95.1	98.6	99.8	99.9	<b>100.0</b>	<b>100.0</b>
	ours: <i>BING-E</i>	0.0	10.3	40.9	73.7	<b>91.5</b>	<b>98.8</b>	<b>99.8</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>	<b>100.0</b>
MABO	EdgeBoxes(Recall)	25.5	39.9	54.2	63.5	<b>71.6</b>	77.0	80.0	81.9	83.4	85.7	85.0
	MCG	<b>48.9</b>	<b>53.9</b>	<b>61.8</b>	<b>66.5</b>	<b>71.6</b>	<b>77.1</b>	<b>81.8</b>	<b>86.6</b>	<b>90.2</b>	<b>94.0</b>	97.7
	SelectiveSearch	22.3	41.4	55.9	62.6	67.8	73.5	78.9	83.6	87.7	92.2	<b>98.0</b>
	ours: <i>BING-E</i>	18.5	32.4	47.6	61.0	68.3	74.5	78.1	80.9	82.7	86.1	95.6

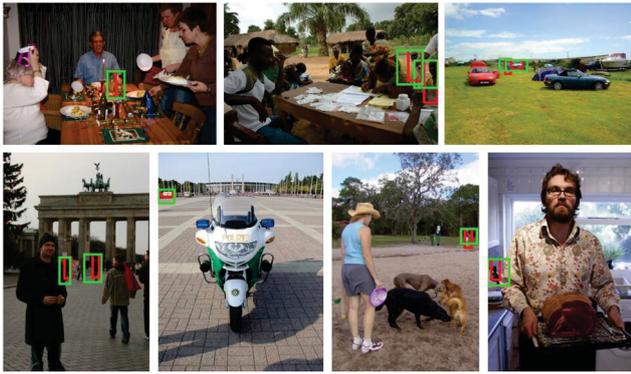
results. When the ground truth area is small, BING-E performs much worse than the other methods. As the ground truth area increases, the gap between BING-E and other state-of-the-art methods gradually narrows, and BING-E outperforms all of them on the recall metric when the area is larger than 2<sup>12</sup>. Figure 4 shows some failing examples produced by BING-E. Note that almost all falsely detected objects

are small. Such small objects may have blurred boundaries making them hard to distinguish from background.

Note that MCG achieves much better performance on small objects, and it may be the main cause of the drop in detection rate when using MCG in the fast R-CNN framework. The fast R-CNN uses the VGG16 [83] model, in which the convolutional layers



**Fig. 4** True positive object proposals for VOC2007 test images using BING-E.



**Fig. 5** Some failure examples of BING-E. Failure means that the overlap between the best detected box (green) and ground truth (red) is less than 0.5. All images are from the VOC2007 test set.

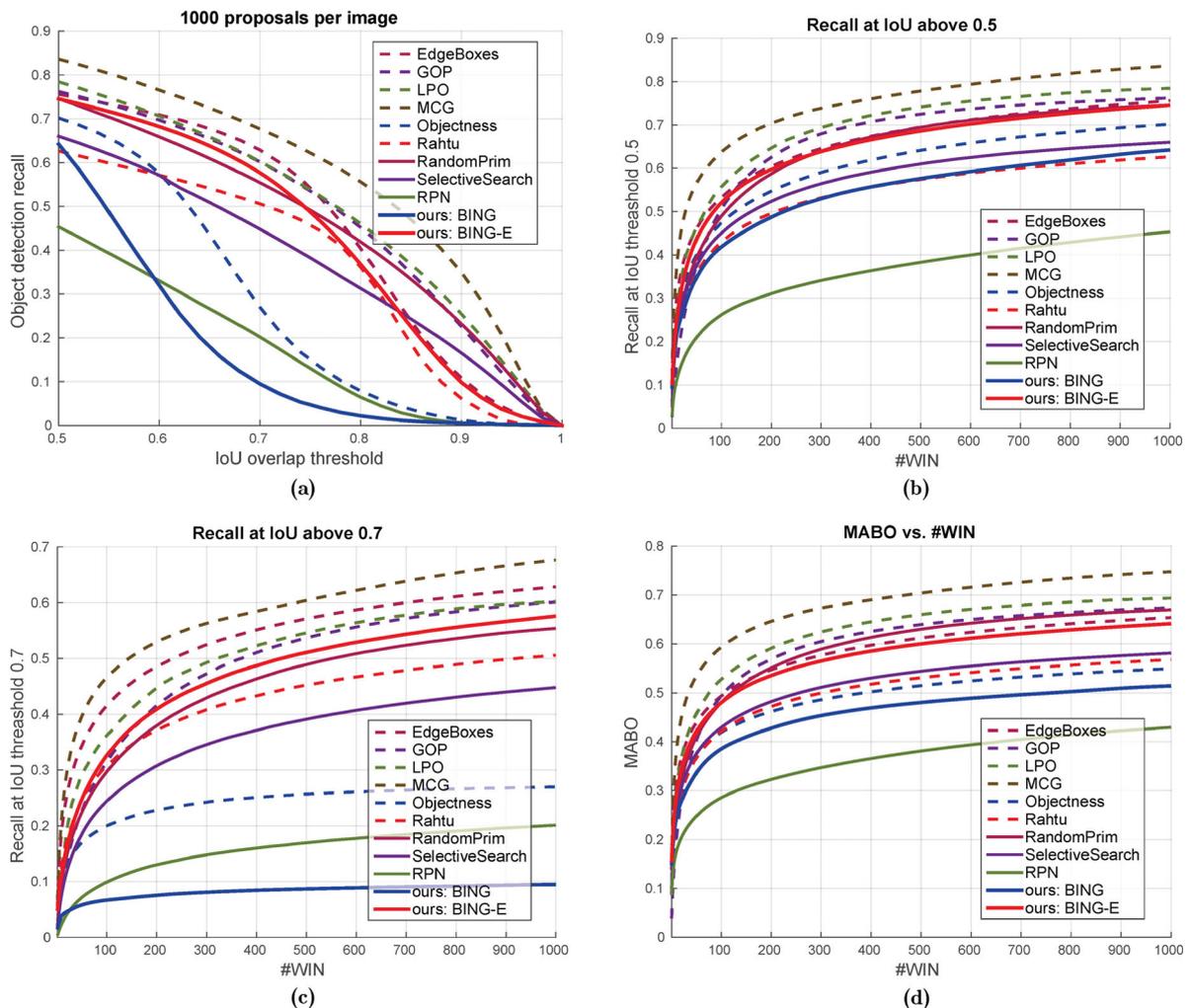
are pooled several times. The size of a feature map will be just  $1/2^4$  size of the original object when it arrives at the last convolutional layer of VGG16,

and the feature map will be too coarse to classify such small instances. Thus, using MCG proposals to retrain the CNN model may confuse the network because of the detected small object proposals. As a result, MCG does not achieve the best performance in the object detection task although it outperforms others on recall and MABO metrics.

### 5.4 Microsoft COCO

In order to test the generalization ability of the various methods, we extensively evaluated them on the COCO validation set using the same parameters as for the VOC2007 dataset, without retraining. As this dataset is so large, we only compared against some of the more efficient methods.

Figure 6(a) shows object detection recall versus IoU overlap threshold using different numbers of proposals. MCG always dominates the performance,



**Fig. 6** Testing results on COCO validation dataset: (a) object detection recall versus IoU overlap threshold; (b, c) recall versus number of candidates at IoU thresholds 0.5 and 0.7 respectively; (d) MABO versus the number of candidates using at most 1000 proposals.

but its low speed makes it unsuited to many vision applications. EdgeBoxes performs well when the IoU threshold is small, and LPO performs well for large IoU thresholds. The performance of BING-E is slightly worse than state-of-the-art performance. Both BING, Rahtu, and Objectness struggle on the COCO dataset, suggesting that these methods may be not robust in complex scenes. RPN performs very poorly on COCO, which means it is highly dependent on the training data. As noted in Ref. [82], a good object proposal algorithm should be category independent. Although RPN achieves good results on VOC2007, it is not consistent with the goal of designing a category independent object proposal method.

Figures 6(b)–6(d) show recall and MABO when varying the number of proposals. Clearly, RPN suffers a big drop in performance over VOC2007. Its recall at IoU 0.5 and MABO are even worse than those of BING. BING and BING-E are very robust when transferring to different object classes. Table 7 shows a statistical comparison. Although BING and BING-E do not achieve the best performance, they obtain very high computational efficiency with a moderate drop in accuracy. The significant improvement from BING to BING-E suggests that BING would be a good basis for combining with other more accurate bounding box refinement methods in cases where the increased computational load is acceptable.

**Table 7** Detection recall (%) using different IoU thresholds and #WIN on COCO validation set

Method \ #WIN	IoU=0.5			IoU=0.7			MABO (1000)
	100	500	1000	100	500	1000	
EdgeBoxes	53.3	69.5	75.6	41.6	57.1	62.9	65.4
GOP	50.6	72.5	76.2	31.5	53.7	60.2	67.4
LPO	55.8	74.1	78.4	36.2	54.6	60.2	69.4
MCG	<b>63.8</b>	<b>77.8</b>	<b>83.6</b>	<b>46.6</b>	<b>60.4</b>	<b>67.7</b>	<b>74.8</b>
Objectness	47.4	64.1	70.2	20.0	25.7	27.0	54.9
Rahtu	43.0	57.4	62.6	30.8	45.2	50.6	56.8
RandomPrim	49.0	69.4	74.6	29.7	48.9	55.4	67.0
SelectiveSearch	45.0	61.0	66.0	24.4	39.1	44.8	58.1
RPN	26.2	38.3	45.3	9.9	17.0	20.1	43.0
<i>ours: BING</i>	41.8	57.6	64.2	6.7	8.7	9.5	51.4
<i>ours: BING-E</i>	52.1	68.6	74.6	32.6	51.1	57.6	64.2

## 6 Conclusions and future work

### 6.1 Conclusions

We have presented a surprisingly simple, fast, and high-quality objectness measure using  $8 \times 8$  binarized normed gradients (BING) features. Computing the objectness of each image window at any scale and aspect ratio only needs a few atomic (ADD, BITWISE, etc.) operations. To improve the localization quality of BING, we further proposed BING-E which incorporates an efficient image segmentation strategy. Evaluation results using the most widely used benchmarks (VOC2007 and COCO) and evaluation metrics show that BING-E can generate state-of-the-art generic object proposals at a significantly higher speed than other methods. Our evaluation demonstrates that BING is a good basis for object proposal generation.

### 6.2 Limitations

BING and BING-E predict a small set of object bounding boxes. Thus, they share similar limitations with all other bounding box based objectness measure methods [2, 44] and classic sliding window based object detection methods [84, 85]. For some object categories (snakes, wires, etc.), a bounding box might not localize object instances as well as a segmentation region [21, 47, 75].

### 6.3 Future work

The high quality and efficiency of our method make it suitable for many realtime vision applications and uses based on large scale image collections (e.g., ImageNet [86]). In particular, the binary operations and memory efficiency make our BING method suitable for low-power devices [79, 80]. Our speed-up strategy of reducing the number of tested windows is complementary to other speed-up techniques which try to reduce the subsequent processing time required for each location. The efficiency of our method solves the computational bottleneck of proposal based vision tasks such as object detection methods [4, 87], enabling real-time high-quality object detection.

We have demonstrated how to generate a small set (e.g., 1000) of proposals to cover nearly all potential object regions, using very simple BING features and a postprocessing step. It would be interesting to introduce other additional cues to further reduce the number of proposals while

maintaining a high detection rate [88, 89], and to explore more applications [23–27, 29, 90] using BING and BING-E. To encourage future work, the source code will be kept up-to-date at <http://mmcheng.net/bing>.

## Acknowledgements

This research was supported by the National Natural Science Foundation of China (Nos. 61572264, 61620106008).

## References

- [1] Alexe, B.; Deselaers, T.; Ferrari, V. Measuring the objectness of image windows. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 34, No. 11, 2189–2202, 2012.
- [2] Alexe, B.; Deselaers, T.; Ferrari, V. What is an object? In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 73–80, 2010.
- [3] Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 580–587, 2014.
- [4] Girshick, R. Fast R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, 1440–1448, 2015.
- [5] He, K.; Zhang, X.; Ren, S.; Sun, J. Spatial pyramid pooling in deep convolutional networks for visual recognition. In: *Computer Vision–ECCV 2014. Lecture Notes in Computer Science, Vol. 8691*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 346–361, 2014.
- [6] Wang, N.; Li, S.; Gupta, A.; Yeung, D.-Y. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*, 2015.
- [7] Kwak, S.; Cho, M.; Laptev, I.; Ponce, J.; Schmid, C. Unsupervised object discovery and tracking in video collections. In: Proceedings of the IEEE International Conference on Computer Vision, 3173–3181, 2015.
- [8] Kading, C.; Freytag, A.; Rodner, E.; Bodesheim, P.; Denzler, J. Active learning and discovery of object categories in the presence of unnameable instances. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 4343–4352, 2015.
- [9] Cho, M.; Kwak, S.; Schmid, C.; Ponce, J. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1201–1210, 2015.
- [10] Arbeláez, P.; Hariharan, B.; Gu, C.; Gupta, S.; Bourdev, L.; Malik, J. Semantic segmentation using regions and parts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3378–3385, 2012.
- [11] Carreira, J.; Caseiro, R.; Batista, J.; Sminchisescu, C. Semantic segmentation with second-order pooling. In: *Computer Vision–ECCV 2012. Lecture Notes in Computer Science, Vol. 7578*. Fitzgibbon, A.; Lazebnik, S.; Perona, P.; Sato, Y.; Schmid, C. Eds. Springer Berlin Heidelberg, 430–443, 2012.
- [12] Sun, J.; Ling, H. Scale and object aware image retargeting for thumbnail browsing. In: Proceedings of the International Conference on Computer Vision, 1511–1518, 2011.
- [13] Sener, F.; Bas, C.; Ikizler-Cinbis, N. On recognizing actions in still images via multiple features. In: *Computer Vision–ECCV 2012. Workshops and Demonstrations. Lecture Notes in Computer Science, Vol. 7585*. Fusiello, A.; Murino, V.; Cucchiara, R. Eds. Springer Berlin Heidelberg, 263–272, 2012.
- [14] Teuber, H.-L. Physiological psychology. *Annual Review of Psychology* Vol. 6, 267–296, 1955.
- [15] Wolfe, J. M.; Horowitz, T. S. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience* Vol. 5, 495–501, 2004.
- [16] Koch, C.; Ullman, S. Shifts in selective visual attention: Towards the underlying neural circuitry. *Human Neurobiology* Vol. 4, No. 4, 219–227, 1985.
- [17] Desimone, R.; Duncan, J. Neural mechanisms of selective visual attention. *Annual Review of Neuroscience* Vol. 18, 193–222, 1995.
- [18] Forsyth, D. A.; Malik, J.; Fleck, M. M.; Greenspan, H.; Leung, T.; Belongie, S.; Carson, C.; Bregler, C. Finding pictures of objects in large collections of images. In: *Object Representation in Computer Vision II. Lecture Notes in Computer Science, Vol. 1144*. Ponce, J.; Zisserman, A.; Hebert, M. Eds. Springer Berlin Heidelberg, 335–360, 1996.
- [19] Heitz, G.; Koller, D. Learning spatial context: Using stuff to find things. In: *Computer Vision–ECCV 2008. Lecture Notes in Computer Science, Vol. 5302*. Forsyth, D.; Torr, P.; Zisserman, A. Eds. Springer Berlin Heidelberg, 30–43, 2008.
- [20] Uijlings, J. R. R.; van de Sande, K. E. A.; Gevers, T.; Smeulders, A. W. M. Selective search for object recognition. *International Journal on Computer Vision* Vol. 104, No. 2, 154–171, 2013.

- [21] Endres, I.; Hoiem, D. Category-independent object proposals with diverse ranking. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 36, No. 2, 222–234, 2014.
- [22] Cheng, M.-M.; Zhang, Z.; Lin, W.-Y.; Torr, P. H. S. BING: Binarized normed gradients for objectness estimation at 300fps. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3286–3293, 2014.
- [23] Wei, Y.; Xia, W.; Lin, M.; Huang, J.; Ni, B.; Dong, J.; Zhao, Y.; Yan, S. HCP: A flexible CNN framework for multi-label image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 38, No. 9, 1901–1907, 2016.
- [24] Zha, S.; Luisier, F.; Andrews, W.; Srivastava, N.; Salakhutdinov, R. Exploiting image-trained CNN architectures for unconstrained video classification. In: Proceedings of the British Machine Vision Conference, 2015.
- [25] Pinheiro, P. O.; Collobert, R. From image-level to pixel-level labeling with convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1713–1721, 2015.
- [26] Wu, J.; Yu, Y.; Huang, C.; Yu, K. Deep multiple instance learning for image classification and auto-annotation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3460–3469, 2015.
- [27] Lee, Y. J.; Grauman, K. Predicting important objects for egocentric video summarization. *International Journal on Computer Vision* Vol. 114, No. 1, 38–55, 2015.
- [28] Paisitkriangkrai, S.; Shen, C.; Hengel, A. v. d. Pedestrian detection with spatially pooled features and structured ensemble learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 38, No. 6, 1243–1257, 2016.
- [29] Zhang, D.; Han, J.; Li, C.; Wang, J.; Li, X. Detection of co-salient objects by looking deep and wide. *International Journal on Computer Vision* Vol. 120, No. 2, 215–232, 2016.
- [30] Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 39, No. 6, 1137–1149, 2015.
- [31] Redmon, J.; Farhadi, A. YOLO9000: Better, faster, stronger. *arXiv preprint* arXiv:1612.08242, 2016.
- [32] Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.-Y.; Berg, A. C. SSD: Single shot multibox detector. In: *Computer Vision–ECCV 2016. Lecture Notes in Computer Science, Vol. 9905*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 21–37, 2016.
- [33] Everingham, M.; Van Gool, L.; Williams, C. K. I.; Winn, J.; Zisserman, A. The PASCAL visual object classes (VOC) challenge. *International Journal on Computer Vision* Vol. 88, No. 2, 303–338, 2010.
- [34] Zitnick, C. L.; Dollár, P. Edge boxes: Locating object proposals from edges. In: *Computer Vision–ECCV 2014. Lecture Notes in Computer Science, Vol. 8693*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 391–405, 2014.
- [35] Hosang, J.; Benenson, R.; Dollár, P.; Schiele, B. What makes for effective detection proposals? *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 38, No. 4, 814–830, 2016.
- [36] Pont-Tuset, J.; Arbeláez, P.; Barron, J. T.; Marques, F.; Malik, J. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 39, No. 1, 128–140, 2017.
- [37] Zhao, Q.; Liu, Z.; Yin, B. Cracking BING and beyond. In: Proceedings of the British Machine Vision Conference, 2014.
- [38] Chen, X.; Ma, H.; Wang, X.; Zhao, Z. Improving object proposals with multi-thresholding straddling expansion. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2587–2595, 2015.
- [39] Ren, C. Y.; Prisacariu, V. A.; Reid, I. D. gSLICr: SLIC superpixels at over 250Hz. *arXiv preprint* arXiv:1509.04232, 2015.
- [40] Achanta, R.; Shaji, A.; Smith, K.; Lucchi, A.; Fua, P.; SÁijsstrunk, S. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 34, No. 11, 2274–2282, 2012.
- [41] Felzenszwalb, P. F.; Huttenlocher, D. P. Efficient graph-based image segmentation. *International Journal on Computer Vision* Vol. 59, No. 2, 167–181, 2004.
- [42] Cheng, M.-M.; Liu, Y.; Hou, Q.; Bian, J.; Torr, P.; Hu, S.-M.; Tu, Z. HFS: Hierarchical feature selection for efficient image segmentation. In: *Computer Vision–ECCV 2016. Lecture Notes in Computer Science, Vol. 9907*. Leibe, B.; Matas, J.; Sebe, N.; Welling, M. Eds. Springer Cham, 867–882, 2016.
- [43] Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C. L. Microsoft COCO: Common objects in context. In: *Computer Vision–ECCV 2014. Lecture Notes in Computer Science, Vol. 8693*. Fleet, D.; Pajdla, T.;

- Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 740–755, 2014.
- [44] Zhang, Z.; Warrell, J.; Torr, P. H. S. Proposal generation for object detection using cascaded ranking SVMs. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1497–1504, 2011.
- [45] Rahtu, E.; Kannala, J.; Blaschko, M. B. Learning a category independent object detection cascade. In: Proceedings of the International Conference on Computer Vision, 1052–1059, 2011.
- [46] Manen, S.; Guillaumin, M.; Van Gool, L. Prime object proposals with randomized Prim’s algorithm. In: Proceedings of the IEEE International Conference on Computer Vision, 2536–2543, 2013.
- [47] Rantalankila, P.; Kannala, J.; Rahtu, E. Generating object segmentation proposals using global and local search. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2417–2424, 2014.
- [48] Krähenbühl, P.; Koltun, V. Geodesic object proposals. In: *Computer Vision–ECCV 2014. Lecture Notes in Computer Science, Vol. 8693*. Fleet, D.; Pajdla, T.; Schiele, B.; Tuytelaars, T. Eds. Springer Cham, 725–739, 2014.
- [49] Krähenbühl, P.; Koltun, V. Learning to propose objects. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1574–1582, 2015.
- [50] Humayun, A.; Li, F.; Rehg, J. M. RIGOR: Reusing inference in graph cuts for generating object regions. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 336–343, 2014.
- [51] Borji, A.; Cheng, M. M.; Jiang, H. et al. Salient object detection: A survey. *arXiv preprint* arXiv:1411.5878, 2014.
- [52] Judd, T.; Durand, F.; Torralba, A. A benchmark of computational models of saliency to predict human fixations. Technical Report. MIT Tech Report, 2012.
- [53] Itti, L.; Koch, C.; Niebur, E. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 20, No. 11, 1254–1259, 1998.
- [54] Ma, Y.-F.; Zhang, H.-J. Contrast-based image attention analysis by using fuzzy growing. In: Proceedings of the 11th ACM International Conference on Multimedia, 374–381, 2003.
- [55] Harel, J.; Koch, C.; Perona, P. Graph-based visual saliency. In: Proceedings of the 19th International Conference on Neural Information Processing Systems, 545–552, 2006.
- [56] Borji, A.; Sihite, D. N.; Itti, L. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing* Vol. 22, No. 1, 55–69, 2013.
- [57] Li, Y.; Hou, X.; Koch, C.; Rehg, J. M.; Yuille, A. L. The secrets of salient object segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 280–287, 2014.
- [58] Borji, A.; Cheng, M.-M.; Jiang, H.; Li, J. Salient object detection: A benchmark. *IEEE Transactions on Image Processing* Vol. 24, No. 12, 5706–5722, 2015.
- [59] Liu, T.; Sun, J.; Zheng, N.; Tang, X.; Shum, H. Learning to detect a salient object. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1–8, 2007.
- [60] Achanta, R.; Hemami, S.; Estrada, F.; Susstrunk, S. Frequency-tuned salient region detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1597–1604, 2009.
- [61] Cheng, M.-M.; Mitra, N. J.; Huang, X.; Torr, P. H. S.; Hu, S.-M. Global contrast based salient region detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 37, No. 3, 569–582, 2015.
- [62] Perazzi, F.; Krähenbühl, P.; Pritch, Y.; Hornung, A. Saliency filters: Contrast based filtering for salient region detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 733–740, 2012.
- [63] Cheng, M.-M.; Zheng, S.; Lin, W.-Y.; Vineet, V.; Sturgess, P.; Crook, N.; Mitra, N. J.; Torr, P. ImageSpirit: Verbal guided image parsing. *ACM Transactions on Graphics* Vol. 34, No. 1, Article No. 3, 2014.
- [64] Zheng, S.; Cheng, M.-M.; Warrell, J.; Sturgess, P.; Vineet, V.; Rother, C.; Torr, P. H. S. Dense semantic image segmentation with objects and attributes. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3214–3221, 2014.
- [65] Li, K.; Zhu, Y.; Yang, J.; Jiang, J. Video super-resolution using an adaptive superpixel-guided autoregressive model. *Pattern Recognition* Vol. 51, 59–71, 2016.
- [66] Zhang, G.-X.; Cheng, M.-M.; Hu, S.-M.; Martin, R. R. A shape-preserving approach to image resizing. *Computer Graphics Forum* Vol. 28, No. 7, 1897–1906, 2009.
- [67] Zheng, Y.; Chen, X.; Cheng, M.-M.; Zhou, K.; Hu, S.-M.; Mitra, N. J. Interactive images: Cuboid proxies for smart image manipulation. *ACM Transactions on Graphics* Vol. 31, No. 4, Article No. 99, 2012.

- [68] Chen, T.; Cheng, M.-M.; Tan, P.; Shamir, A.; Hu, S.-M. Sketch2Photo: Internet image montage. *ACM Transactions on Graphics* Vol. 28, No. 5, Article No. 124, 2009.
- [69] Huang, H.; Zhang, L.; Zhang, H.-C. Arcimboldo-like collage using internet images. *ACM Transactions on Graphics* Vol. 30, No. 6, Article No. 155, 2011.
- [70] Chia, A. Y.-S.; Zhuo, S.; Gupta, R. K.; Tai, Y.-W.; Cho, S.-Y.; Tan, P.; Lin, S. Semantic colorization with internet images. *ACM Transactions on Graphics* Vol. 30, No. 6, Article No. 156, 2011.
- [71] He, J.; Feng, J.; Liu, X.; Cheng, T.; Lin, T.-H.; Chung, H.; Chang, S.-F. Mobile product search with bag of hash bits and boundary reranking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 3005–3012, 2012.
- [72] Chen, T.; Tan, P.; Ma, L.-Q.; Cheng, M.-M.; Shamir, A.; Hu, S.-M. PoseShop: Human image database construction and personalized content synthesis. *IEEE Transactions on Visualization and Computer Graphics* Vol. 19, No. 5, 824–837, 2013.
- [73] Hu, S.-M.; Chen, T.; Xu, K.; Cheng, M.-M.; Martin, R. R. Internet visual media processing: A survey with graphics and vision applications. *The Visual Computer* Vol. 29, No. 5, 393–405, 2013.
- [74] Cheng, M.-M.; Mitra, N. J.; Huang, X.; Hu, S.-M. SalientShape: Group saliency in image collections. *The Visual Computer* Vol. 30, No. 4, 443–453, 2014.
- [75] Carreira, J.; Sminchisescu, C. CPMC: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 34, No. 7, 1312–1328, 2012.
- [76] Lu, C.; Liu, S.; Jia, J.; Tang, C.-K. Contour box: Rejecting object proposals without explicit closed contours. In: Proceedings of the IEEE International Conference on Computer Vision, 2021–2029, 2015.
- [77] Fan, R.-E.; Chang, K.-W.; Hsieh, C.-J.; Wang, X.-R.; Lin, C.-J. LIBLINEAR: A library for large linear classification. *The Journal of Machine Learning Research* Vol. 9, 1871–1874, 2008.
- [78] Gottlieb, J. P.; Kusunoki, M.; Goldberg, M. E. The representation of visual salience in monkey parietal cortex. *Nature* Vol. 391, No. 6666, 481–484, 1998.
- [79] Hare, S.; Saffari, A.; Torr, P. H. S. Efficient online structured output learning for keypoint-based object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1894–1901, 2012.
- [80] Zheng, S.; Sturges, P.; Torr, P. H. S. Approximate structured output learning for constrained local models with application to real-time facial feature detection and tracking on low-power devices. In: Proceedings of the 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition, 1–8, 2013.
- [81] 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, I–I, 2001.
- [82] Chavali, N.; Agrawal, H.; Mahendru, A.; Batra, D. Object-proposal evaluation protocol is ‘gameable’. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 835–844, 2016.
- [83] Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.
- [84] Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Vol. 1, 886–893, 2005.
- [85] Felzenszwalb, P. F.; Girshick, R. B.; McAllester, D.; Ramanan, D. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 32, No. 9, 1627–1645, 2010.
- [86] Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; Fei-Fei, L. ImageNet: A large-scale hierarchical image database. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 248–255, 2009.
- [87] He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 770–778, 2016.
- [88] Kuo, W.; Hariharan, B.; Malik, J. DeepBox: Learning objectness with convolutional networks. In: Proceedings of the IEEE International Conference on Computer Vision, 2479–2487, 2015.
- [89] Zhang, Z.; Liu, Y.; Chen, X.; Zhu, Y.; Cheng, M.-M.; Saligrama, V.; Torr, P. H. Sequential optimization for efficient high-quality object proposal generation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol. 40, No. 5, 1209–1223, 2018.
- [90] Chen, W.; Xiong, C.; Xu, R.; Corso, J. J. Actionness ranking with lattice conditional ordinal random fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 748–755, 2014.



**Ming-Ming Cheng** received his Ph.D. degree from Tsinghua University in 2012. Then he worked for 2 years as a research fellow with Prof. Philip Torr in Oxford. He is now an associate professor at Nankai University, leading the Media Computing Lab. His research interests include computer graphics, computer

vision, and image processing.

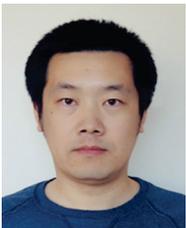


**Yun Liu** is a Ph.D. candidate with the College of Computer Science, Nankai University, under the supervision of Prof. Ming-Ming Cheng. His major research interests are computer vision and machine learning.



**Wen-Yan Lin** received his Ph.D. degree from the National University of Singapore in 2012, supervised by Prof. Loong-Fah Cheong and Dr. Dong Guo. He subsequently worked for the Institute of Infocomm Research Singapore and Prof. Philip Torr. He is currently a post-

doc at the Advanced Digital Sciences Center, Singapore.



**Ziming Zhang** is a research scientist at Mitsubishi Electric Research Laboratories (MERL). Before joining MERL he was a research assistant professor at Boston University. He received his Ph.D. degree in 2013 from Oxford Brookes University, UK, under the supervision of Prof. Philip Torr.



**Paul L. Rosin** is a professor at the School of Computer Science & Informatics, Cardiff University, Wales. His research interests include the representation, segmentation, and grouping of curves, knowledge-based vision systems, early image representations, low-level image

processing, machine vision approaches to remote sensing, methods for evaluation of approximation algorithms, medical and biological image analysis, mesh processing, non-photorealistic rendering, and the analysis of shape in art and architecture.



**Philip H. S. Torr** received his Ph.D. degree from Oxford University. After working for another three years at Oxford, he worked for six years as a research scientist for Microsoft Research, first in Redmond, then in Cambridge, founding the vision side of the Machine Learning and Perception Group. He is

now a professor at Oxford University. He has won awards from several top vision conferences, including ICCV, CVPR, ECCV, NIPS, and BMVC. He is a Royal Society Wolfson Research Merit Award holder.

**Open Access** The articles published in this journal are distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

Other papers from this open access journal are available free of charge from <http://www.springer.com/journal/41095>. To submit a manuscript, please go to <https://www.editorialmanager.com/cvmj>.

# BING: Binarized Normed Gradients for Objectness Estimation at 300fps

Ming-Ming Cheng<sup>1</sup> Ziming Zhang<sup>2</sup> Wen-Yan Lin<sup>3</sup> Philip Torr<sup>1</sup>

<sup>1</sup>The University of Oxford <sup>2</sup>Boston University <sup>3</sup>Brookes Vision Group

## Abstract

Training a generic objectness measure to produce a small set of candidate object windows, has been shown to speed up the classical sliding window object detection paradigm. We observe that generic objects with well-defined closed boundary can be discriminated by looking at the norm of gradients, with a suitable resizing of their corresponding image windows in to a small fixed size. Based on this observation and computational reasons, we propose to resize the window to  $8 \times 8$  and use the norm of the gradients as a simple 64D feature to describe it, for explicitly training a generic objectness measure.

We further show how the binarized version of this feature, namely binarized normed gradients (BING), can be used for efficient objectness estimation, which requires only a few atomic operations (e.g. ADD, BITWISE SHIFT, etc.). Experiments on the challenging PASCAL VOC 2007 dataset show that our method efficiently (300fps on a single laptop CPU) generates a small set of category-independent, high quality object windows, yielding 96.2% object detection rate (DR) with 1,000 proposals. Increasing the numbers of proposals and color spaces for computing BING features, our performance can be further improved to 99.5% DR.

## 1. Introduction

As one of the most important areas in computer vision, object detection has made great strides in recent years. However, most state-of-the-art detectors still require each *category specific* classifiers to evaluate many image windows in a sliding window fashion [17, 25]. In order to reduce the number of windows each classifier needs to consider, training an objectness measure which is *generic over categories* has recently becomes popular [2, 3, 21, 22, 48, 49, 57]. *Objectness* is usually represented as a value which reflects how likely an image window covers an object of *any category* [3]. A generic objectness measure has great potential to be used in a pre-filtering process to significantly improve: i) computational efficiency by reducing the search space, and ii) detection accuracy by allowing the usage

of strong classifiers during testing. However, designing a good generic objectness measure method is difficult, which should:

- achieve **high object detection rate** (DR), as any undetected objects at this stage cannot be recovered later;
- produce **a small number of proposals** for reducing computational time of subsequent detectors;
- obtain **high computational efficiency** so that the method can be easily involved in various applications, especially for realtime and large-scale applications;
- have **good generalization ability** to unseen object categories, so that the proposals can be reused by many category specific detectors to greatly reduce the computation for each of them.

To the best of our knowledge, no prior method can satisfy all these ambitious goals simultaneously.

Research from cognitive psychology [47, 54] and neurobiology [20, 38] suggest that humans have a strong ability to perceive objects before identifying them. Based on the human reaction time that is observed and the biological signal transmission time that is estimated, human attention theories hypothesize that the human vision system processes only parts of an image in detail, while leaving others nearly unprocessed. This further suggests that before identifying objects, there are simple mechanisms in the human vision system to select possible object locations.

In this paper, we propose a surprisingly simple and powerful feature “BING” to help the search for objects using objectness scores. Our work is motivated by the fact that objects are stand-alone things with well-defined closed boundaries and centers [3, 26, 32]. We observe that generic objects with well-defined closed boundaries share surprisingly strong correlation when looking at the norm of the gradient (see Fig. 1 and Sec. 3), after resizing of their corresponding image windows to small fixed size (e.g.  $8 \times 8$ ). Therefore, in order to efficiently quantify the objectness of an image window, we resize it to  $8 \times 8$  and use the norm of the gradients as a simple 64D feature for learning a generic objectness measure in a cascaded SVM framework. We further show how the binarized version of the NG feature, namely binarized normed gradients (**BING**) feature, can be used for efficient objectness estimation of image windows, which re-

quires only a few atomic CPU operations (*i.e.* ADD, BITWISE SHIFT, etc.). The BING feature’s simplicity, contrast with recent state of the art techniques [3, 22, 48] which seek increasingly sophisticated features to obtain greater discrimination, while using advanced speed up techniques to make the computational time tractable.

We have extensively evaluated our method on the PASCAL VOC2007 dataset [23]. The experimental results show that our method efficiently (300fps on a single laptop CPU) generates a small set of data-driven, category-independent, high quality object windows, yielding 96.2% detection rate (DR) with 1,000 windows ( $\approx 0.2\%$  of full sliding windows). Increasing the number of object windows to 5,000, and estimating objectness in 3 different color spaces, our method can achieve 99.5% DR. Following [3, 22, 48], we also verify the generalization ability of our method. When training our objectness measure on 6 object categories and testing on other 14 *unseen* categories, we observed similar high performance as in standard settings (see Fig. 3). Compared to most popular alternatives [3, 22, 48], the BING features allow us to achieve better DR using a smaller set of proposals, is much simpler and 1000+ times faster, while being able to predict *unseen* categories. This fulfills aforementioned requirements of a good objectness detector. Our source code will be published with the paper.

## 2. Related works

Being able to perceive objects before identifying them is closely related to bottom up visual attention (saliency). According to how saliency is defined, we broadly classify the related research into three categories: fixation prediction, salient object detection, and objectness proposal generation.

**Fixation prediction** models aim at predicting saliency points of human eye movement [4, 37]. Inspired by neurobiology research about early primate visual system, Itti *et al.* [36] proposed one of the first computational models for saliency detection, which estimates center-surrounded difference across multi-scale image features. Ma and Zhang [42] proposed a fuzzy growing model to analyze local contrast based saliency. Harel *et al.* [29] proposed normalizing center-surrounded feature maps for highlighting conspicuous parts. Although fixation point prediction models have achieved remarkable development, the prediction results tends to highlight edges and corners rather than the entire objects. Thus, these models are not suitable for generating object proposals for detection purpose.

**Salient object detection** models try to detect the most attention-grabbing object in a scene, and then segment the whole extent of that object [5, 40]. Liu *et al.* [41] combined local, regional, and global saliency measurements

in a CRF framework. Achanta *et al.* [1] localized salient regions using a frequency-tuned approach. Cheng *et al.* [11, 14] proposed a salient object detection and segmentation method based on region contrast analysis and iterative graph based segmentation. More recent research also tried to produce high quality saliency maps in a filtering based framework [46], using efficient data representation [12], or consider hierarchical structures [55]. Such salient object segmentation for simple images achieved great success in image scene analysis [15, 58], content aware image editing [13, 56, 60], and it can be used as a cheap tool to process large number of Internet images or build robust applications [7, 8, 16, 31, 34, 35] by automatically selecting good results [10, 11]. However, these approaches are less likely to work for complicated images where many objects are presented and they are rarely dominant (e.g. VOC [23]).

**Objectness proposal generation** methods avoid making decisions early on, by proposing a small number (e.g. 1,000) of category-independent proposals, that are expected to cover all objects in an image [3, 22, 48]. Producing rough segmentations [6, 21] as object proposals has been shown to be an effective way of reducing search spaces for category-specific classifiers, whilst allowing the usage of strong classifiers to improve accuracy. However, these two methods are computationally expensive, requiring 2-7 minutes per image. Alexe *et al.* [3] proposed a cue integration approach to get better prediction performance more efficiently. Zhang *et al.* [57] proposed a cascaded ranking SVM approach with orientated gradient feature for efficient proposal generation. Uijlings *et al.* [48] proposed a selective search approach to get higher prediction performance. We propose a simple and intuitive method which generally achieves better detection performance than others, and is 1,000+ times faster than most popular alternatives [3, 22, 48] (see Sec. 4).

In addition, for efficient sliding window object detection, keeping the computational cost feasible is very important [43, 51]. Lampert *et al.* [39] presented an elegant branch-and-bound scheme for detection. However, it can only be used to speed up classifiers that users can provide a good bound on highest score. Also, some other efficient classifiers [17] and approximate kernels [43, 51] have been proposed. These methods aim to reduce computational cost of evaluating one window, and naturally can be combined with objectness proposal methods to further reduce the cost.

## 3. Methodology

Inspired by the ability of human vision system which efficiently perceives objects before identifying them [20, 38, 47, 54], we introduce a simple 64D norm of the gradients (NG) feature (Sec. 3.1), as well as its binary approximation, *i.e.* binarized normed gradients (BING) feature (Sec. 3.3), for efficiently capturing the objectness of an image window.

To find generic objects within an image, we scan over a predefined *quantized window sizes* (scales and aspect ratios<sup>1</sup>). Each window is scored with a linear model  $\mathbf{w} \in \mathbb{R}^{64}$  (Sec. 3.2),

$$s_l = \langle \mathbf{w}, \mathbf{g}_l \rangle, \quad (1)$$

$$l = (i, x, y), \quad (2)$$

where  $s_l$ ,  $\mathbf{g}_l$ ,  $l$ ,  $i$  and  $(x, y)$  are filter score, NG feature, location, size and position of a window respectively. Using non-maximal suppression (NMS), we select a small set of proposals from each size  $i$ . Some sizes (e.g.  $10 \times 500$ ) are less likely than others to contain an object instance (e.g.  $100 \times 100$ ). Thus we define the objectness score (i.e. calibrated filter score) as

$$o_l = v_i \cdot s_l + t_i, \quad (3)$$

where  $v_i, t_i \in \mathbb{R}$  are sperately learnt coefficient and a bias terms for each quantised size  $i$  (Sec. 3.2). Note that calibration using (3), although very fast, is only required when re-ranking the small set of final proposals.

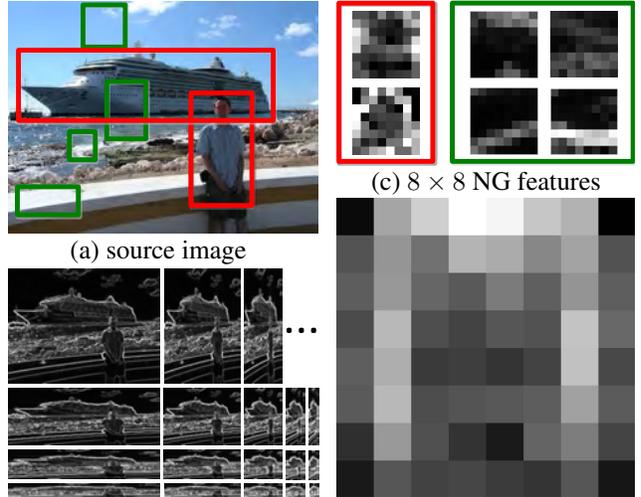
### 3.1. Normed gradients (NG) and objectness

Objects are stand-alone things with well-defined closed boundaries and centers [3, 26, 32]. When resizing windows corresponding to real world objects to a small fixed size (e.g.  $8 \times 8$ , chosen for computational reasons that will be explained in Sec. 3.3), the norm (i.e. magnitude) of the corresponding image gradients becomes a good discriminative feature, because of the little variation that closed boundaries could present in such abstracted view. As demonstrated in Fig. 1, although the cruise ship and the person have huge difference in terms of color, shape, texture, illumination *etc.*, they do share clear correlation in normed gradient space. To utilize this observation for efficiently predicting the existence of object instances, we firstly resize the input image to different *quantized sizes* and calculate the normed gradients of each resized image. The values in an  $8 \times 8$  region of these resized normed gradients maps are defined as a 64D *normed gradients (NG)*<sup>2</sup> feature of its corresponding window.

Our NG feature, as a dense and compact objectness feature for an image window, has several advantages. Firstly, no matter how an object changes its position, scale and aspect ratio, its corresponding NG feature will remain roughly unchanged because of the normalized support region of this feature. In other words, NG features are insensitive to change of translation, scale and aspect ratio, which will be very useful for detecting objects of arbitrary categories.

<sup>1</sup>In all experiments, we test 36 quantized target window sizes  $\{(W_o, H_o)\}$ , where  $W_o, H_o \in \{10, 20, 40, 80, 160, 320\}$ . We resize the input image to 36 sizes so that  $8 \times 8$  windows in the resized smaller images (from which we extract features), correspond to target windows.

<sup>2</sup>The *normed gradient* represents Euclidean norm of the gradient.



(a) source image (b) normed gradients maps (c)  $8 \times 8$  NG features (d) learned model  $\mathbf{w} \in \mathbb{R}^{8 \times 8}$   
 Figure 1. Although object (red) and non-object (green) windows present huge variation in the image space (a), in proper scales and aspect ratios where they correspond to a small fixed size (b), their corresponding normed gradients, i.e. a NG feature (c), share strong correlation. We learn a single 64D linear model (d) for selecting object proposals based on their NG features.

And these insensitivity properties are what a good objectness proposal generation method should have. Secondly, the dense compact representation of the NG feature makes it very efficient to be calculated and verified, thus having great potential to be involved in realtime applications.

The cost of introducing such advantages to NG feature is the loss of discriminative ability. Lucky, the resulted false-positives will be processed by subsequent category specific detectors. In Sec. 4, we show that our method results in a small set of high quality proposals that cover 96.2% true object windows in the challenging VOC2007 dataset.

### 3.2. Learning objectness measurement with NG

To learn an objectness measure of image windows, we follow the general idea of the two stages cascaded SVM [57].

**Stage I.** We learn a single model  $\mathbf{w}$  for (1) using linear SVM [24]. NG features of the ground truth object windows and random sampled background windows are used as positive and negative training samples respectively.

**Stage II.** To learn  $v_i$  and  $t_i$  in (3) using a linear SVM [24], we evaluate (1) at size  $i$  for training images and use the selected (NMS) proposals as training samples, their filter scores as 1D features, and check their labeling using training image annotations (see Sec. 4 for evaluation criteria).

**Discussion.** As illustrated in Fig. 1d, the learned linear model  $\mathbf{w}$  (see Sec. 4 for experimental settings), looks sim-

---

**Algorithm 1** Binary approximate model  $\mathbf{w}$  [28].

---

**Input:**  $\mathbf{w}, N_w$   
**Output:**  $\{\beta_j\}_{j=1}^{N_w}, \{\mathbf{a}_j\}_{j=1}^{N_w}$   
**Initialize residual:**  $\varepsilon = \mathbf{w}$   
**for**  $j = 1$  to  $N_w$  **do**  
     $\mathbf{a}_j = \text{sign}(\varepsilon)$   
     $\beta_j = \langle \mathbf{a}_j, \varepsilon \rangle / \|\mathbf{a}_j\|^2$  (project  $\varepsilon$  onto  $\mathbf{a}_j$ )  
     $\varepsilon \leftarrow \varepsilon - \beta_j \mathbf{a}_j$  (update residual)  
**end for**

---

ilar to the multi-size center-surrounded patterns [36] hypothesized as biologically plausible architecture of primates [27, 38, 54]. The large weights along the borders of  $\mathbf{w}$  favor a boundary that separate an object (center) from its background (surrounded). Compared to manually designed center surround patterns [36], our learned  $\mathbf{w}$  captures a more sophisticated, natural prior. For example, lower object regions are more often occluded than upper parts. This is represented by  $\mathbf{w}$  placing less confidence in the lower regions.

### 3.3. Binarized normed gradients (BING)

To make use of recent advantages in model binary approximation [28, 59], we propose an accelerated version of NG feature, namely binarized normed gradients (BING), to speed up the feature extraction and testing process. Our learned linear model  $\mathbf{w} \in \mathbb{R}^{64}$  can be approximated with a set of basis vectors  $\mathbf{w} \approx \sum_{j=1}^{N_w} \beta_j \mathbf{a}_j$  using Alg. 1, where  $N_w$  denotes the number of basis vectors,  $\mathbf{a}_j \in \{-1, 1\}^{64}$  denotes a basis vector, and  $\beta_j \in \mathbb{R}$  denotes the corresponding coefficient. By further representing each  $\mathbf{a}_j$  using a binary vector and its complement:  $\mathbf{a}_j = \mathbf{a}_j^+ - \overline{\mathbf{a}_j^+}$ , where  $\mathbf{a}_j^+ \in \{0, 1\}^{64}$ , a binarized feature  $\mathbf{b}$  could be tested using fast BITWISE AND and BIT COUNT operations (see [28]),

$$\langle \mathbf{w}, \mathbf{b} \rangle \approx \sum_{j=1}^{N_w} \beta_j (2 \langle \mathbf{a}_j^+, \mathbf{b} \rangle - |\mathbf{b}|). \quad (4)$$

The key challenge is how to binarize and calculate our NG features efficiently. We approximate the normed gradient values (each saved as a BYTE value) using the top  $N_g$  binary bits of the BYTE values. Thus, a 64D NG feature

---

**Algorithm 2** Get BING features for  $W \times H$  positions.

---

**Comments:** see Fig. 2 for illustration of variables  
**Input:** binary normed gradient map  $b_{W \times H}$   
**Output:** BING feature matrix  $\mathbf{b}_{W \times H}$   
**Initialize:**  $\mathbf{b}_{W \times H} = 0, \mathbf{r}_{W \times H} = 0$   
**for each** position  $(x, y)$  in scan-line order **do**  
     $\mathbf{r}_{x,y} = (\mathbf{r}_{x-1,y} \ll 1) \mid b_{x,y}$   
     $\mathbf{b}_{x,y} = (\mathbf{b}_{x,y-1} \ll 8) \mid \mathbf{r}_{x,y}$   
**end for**

---

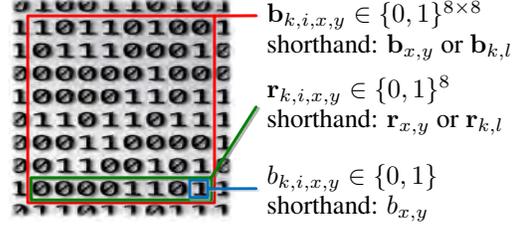


Figure 2. Illustration of variables: a BING feature  $\mathbf{b}_{x,y}$ , its last row  $\mathbf{r}_{x,y}$  and last element  $b_{x,y}$ . Notice that the subscripts  $i, x, y, l, k$ , introduced in (2) and (5), are locations of the whole vector rather than index of vector element. We can use a single atomic variable (INT64 and BYTE) to represent a BING feature and its last row, enabling efficient feature computation (Alg. 2).

$\mathbf{g}_l$  can be approximated by  $N_g$  *binarized normed gradients (BING)* features as

$$\mathbf{g}_l = \sum_{k=1}^{N_g} 2^{8-k} \mathbf{b}_{k,l}. \quad (5)$$

Notice that these BING features have different weights according to its corresponding bit position in BYTE values.

Naively getting an  $8 \times 8$  BING feature requires a loop computing access to 64 positions. By exploring two special characteristics of an  $8 \times 8$  BING feature, we develop a fast BING feature calculation algorithm (Alg. 2), which enables using atomic updates (BITWISE SHIFT and BITWISE OR) to avoid the loop computing. First, a BING feature  $\mathbf{b}_{x,y}$  and its last row  $\mathbf{r}_{x,y}$  could be saved in a single INT64 and a BYTE variables, respectively. Second, adjacent BING features and their rows have a simple cumulative relation. As shown in Fig. 2 and Alg. 2, the operator BITWISE SHIFT shifts  $\mathbf{r}_{x-1,y}$  by one bit, automatically through the bit which does not belong to  $\mathbf{r}_{x,y}$ , and makes room to insert the new bit  $b_{x,y}$  using the BITWISE OR operator. Similarly BITWISE SHIFT shifts  $\mathbf{b}_{x,y-1}$  by 8 bits automatically through the bits which do not belong to  $\mathbf{b}_{x,y}$ , and makes room to insert  $\mathbf{r}_{x,y}$ .

Our efficient BING feature calculation shares the *cumulative* nature with the integral image representation [52]. Instead of calculating a single scalar value over an arbitrary rectangle range [52], our method uses a few atomic operations (e.g. ADD, BITWISE, etc.) to calculate a set of binary patterns over an  $8 \times 8$  fixed range.

The filter score (1) of an image window corresponding to BING features  $\mathbf{b}_{k,l}$  can be efficiently tested using:

$$s_l \approx \sum_{j=1}^{N_w} \beta_j \sum_{k=1}^{N_g} C_{j,k}, \quad (6)$$

where  $C_{j,k} = 2^{8-k} (2 \langle \mathbf{a}_j^+, \mathbf{b}_{k,l} \rangle - |\mathbf{b}_{k,l}|)$  can be tested using fast BITWISE and POPCNT SSE operators.

**Implementation details.** We use the 1-D mask  $[-1, 0, 1]$  to find image gradients  $g_x$  and  $g_y$  in horizontal and vertical directions, while calculating normed gradients using

$\min(|g_x| + |g_y|, 255)$  and saving them in BYTE values. By default, we calculate gradients in RGB color space. In our C++ implementation, POPCNT SSE instructions and OPENMP options are enabled.

#### 4. Experimental Evaluation

We extensively evaluate our method on VOC2007 [23] using the DR-#WIN<sup>3</sup> evaluation metric, and compare our results with 3 state-of-the-art methods [3, 48, 57] in terms of proposal quality, generalize ability, and efficiency. As demonstrated by [3, 48], a small set of coarse locations with high detection rate (DR) are sufficient for effective object detection, and it allows expensive features and complementary cues to be involved in detection to achieve better quality and higher efficiency than traditional methods. Note that in all comparisons, we use the authors’ public implementations<sup>4</sup> with their suggested parameter settings.

**Proposal quality comparisons.** Following [3, 48, 57], we evaluate DR-#WIN on VOC2007 test set, which consists of 4,952 images with bounding box annotation for the object instances from 20 categories. The large number of objects and high variety of categories, viewpoint, scale, position, occlusion, and illumination, make this dataset very suitable to our evaluation as we want to find *all* objects in the images. Fig. 3 shows the statistical comparison between our method and state-of-the-art alternatives: OBN [3], SEL [48], and CSVM [57]. As observed by [48], increasing the divergence of proposals by collecting the results from different parameter settings would improve the DR at the cost of increasing the number of proposals (#WIN). SEL [48] uses 80 different parameters to get combined results and achieves 99.1% DR using more than 10,000 proposals. Our method achieves 99.5% DR using only 5,000 proposals by simply collecting the results from 3 color spaces (*BING-diversified* in Fig. 3): RGB, HSV, and GRAY. As shown in these DR-#WIN statistics, our simple method achieves better performance than others, in general, and is **more than three orders of magnitude (i.e. 1,000+ times) faster** than most popular alternatives [3, 22, 48] (see Tab. 1). We illustrate sample results with varies complexity in Fig. 4.

**Generalize ability test.** Following [3], we show that our objectness proposals are *generic* over categories by testing our method on images containing objects whose categories

<sup>3</sup>DR-#WIN [3] means detection rate (DR) given #WIN proposals. This evaluation metric is also used in [22, 48] with slightly different names. An object is considered as being covered by a proposal if the strict PASCAL criterion is satisfied. That is, the INT-UION [23] score is no less than 0.5.

<sup>4</sup>Implementations and results can be seen at the websites of the original authors: <http://cms.brookes.ac.uk/research/visiongroup/code.php>, <http://groups.inf.ed.ac.uk/calvin/objectness/>, <http://disi.unitn.it/~uijlings/>, and <http://vision.cs.uiuc.edu/proposals/>.

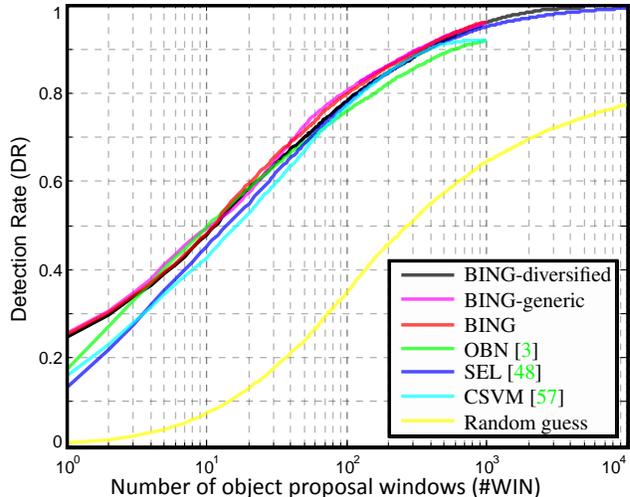


Figure 3. Tradeoff between #WIN and DR for different methods. Our method achieves 96.2% DR using 1,000 proposals, and 99.5% DR using 5,000 proposals. The 3 methods [3, 48, 57] have been evaluated on the same benchmark and shown to outperform other alternative proposal methods [6, 21, 25, 30, 50], saliency measures [33, 36], interesting point detectors [44], and HOG detector [17] (see [3] for the comparisons). Best viewed in color.

are not used for training. Specifically, we train our method using 6 object categories (*i.e.* bird, car, cow, dog, and sheep) and test it using the rest 14 categories (*i.e.* aeroplane, bicycle, boat, bottle, bus, chair, dining-table, horse, motorbike, person, potted-plant, sofa, train, and tv-monitor). In Fig. 3, the statistics for training and testing on same or different object categories are represented by *BING* and *BING-generic*, respectively. As we see, the behavior of these two curves are almost identical, which demonstrates the *generalize* ability of our proposals.

Notice that the recent work [18] enables 20 seconds testing time for detecting 100,000 object classes, by reducing the computational complexity of traditional multi-class detection from  $O(LC)$  to  $O(L)$ , where  $L$  is the number of locations or window proposals and  $C$  is the number of classifiers. The ability of our method to get a small set of high quality proposals of any category (including both trained and unseen categories), could be used to further reduce the computational complexity significantly by reducing  $L$ .

**Computational time.** As shown in Tab. 1, our method is able to efficiently propose a few thousands high quality object windows at 300fps, while other methods require several

Method	[22]	OBN [3]	CSVM [57]	SEL [48]	Our BING
Time (seconds)	89.2	3.14	1.32	11.2	<b>0.003</b>

Table 1. Average computational time on VOC2007.



Figure 4. Illustration of the true positive object proposals for VOC2007 test images. See Fig. 3 for statistical results.

seconds for one image. Note that these methods are usually considered to be highly efficient state-of-the-art algorithms and difficult to further speed up. Moreover, our training on 2501 images (VOC2007) takes much less time (20 seconds excluding xml loading time) than testing a single image using some state-of-the-art alternatives [6, 21] (2+ minutes).

As shown in Tab. 2, with the binary approximation to the learned linear filter (Sec. 3.3) and BING features, computing response score for each image window only needs a fixed small number of atomic operations. It is easy to see that the number of positions at each quantized scale and aspect ratio is equivalent to  $O(N)$ , where  $N$  is the number of pixels in images. Thus, Computing response scores

	BITWISE			FLOAT		INT,BYTE	
	SHIFT	, &	CNT	+	×	+, -	min
Gradient	0	0	0	0	0	9	2
Get BING	12	12	0	0	0	0	0
Get score	0	8	12	1	2	8	0

Table 2. Average number of atomic operations for computing objectness of each image window at different stages: calculate normed gradients, extract BING features, and get objectness score.

at all scales and aspect ratios also has the computational complexity  $O(N)$ . Further, extracting BING feature and computing response score at each potential position (*i.e.* an image window) can be calculated with information given by

$(N_w, N_g)$	(2,3)	(2,4)	(3,2)	(3,3)	(3,4)	N/A
DR (%)	95.9	96.2	95.8	96.2	96.1	96.3

Table 3. Average result quality (DR using 1000 proposals) at different approximation levels, measured by  $N_w$  and  $N_g$  in Sec. 3.3. N/A represents without binarization.

its 2 neighboring positions (*i.e.* left and upper). This means that the space complexity is also  $O(N)$ . We compare our running time with baseline methods [3, 22, 48, 57] on the same laptop with an Intel i7-3940XM CPU.

We further illustrate in Tab. 3 how different approximation levels influence the result quality. According to this comparison, we use  $N_w = 2$ ,  $N_g = 4$  in other experiments.

## 5. Conclusion and Future Work

We present a surprisingly simple, fast, and high quality objectness measure by using  $8 \times 8$  binarized normed gradients (BING) features, with which computing the objectness of each image window at any scale and aspect ratio only needs a few atomic (*i.e.* ADD, BITWISE, etc.) operations. Evaluation results using the most widely used benchmark (VOC2007) and evaluation metric (DR-#WIN) show that our method not only outperforms other state-of-the-art methods, but also runs more than three orders of magnitude faster than most popular alternatives [3, 22, 48].

**Limitations.** Our method predicts a small set of object bounding boxes. Thus, it shares similar limitations as all other bounding box based objectness measure methods [3, 57] and classic sliding window based object detection methods [17, 25]. For some object categories, a bounding box might not localize the object instances as accurately as a segmentation region [6, 21, 22, 45], *e.g.* a snake, wires, etc.

**Future works.** The high quality and efficiency of our method make it suitable for realtime multi-category object detection applications and large scale image collections (*e.g.* ImageNet [19]). The binary operations and memory efficiency make our method suitable to run on low power devices [28, 59].

Our speed-up strategy by reducing the number of windows is complementary to other speed-up techniques which try to reduce the classification time required for each location. It would be interesting to explore the combination of our method with [18] to enable realtime detection of thousands of object categories on a single machine. The efficiency of our method solves the efficiency bottleneck of proposal based object detection method [53], possibly enabling realtime high quality object detection.

We have demonstrated how to get a small set (*e.g.* 1,000) of proposals to cover nearly all (*e.g.* 96.2%) potential object

regions, using very simple BING features. It would be interesting to introduce other additional cues to further reduce the number of proposals while maintaining high detection rate, and explore more applications [9] using BING.

To encourage future works, we make the source code, links to related methods, FAQs, and live discussions available in the project page: <http://mmcheng.net/bing/>.

**Acknowledges:** We acknowledge support of the EPSRC and financial support was provided by ERC grant ERC-2012-AdG 321162-HELIOS.

## References

- [1] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In *CVPR*, 2009.
- [2] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *CVPR*, pages 73–80, 2010.
- [3] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE TPAMI*, 34(11), 2012.
- [4] A. Borji, D. Sihite, and L. Itti. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE TIP*, 2012.
- [5] A. Borji, D. N. Sihite, and L. Itti. Salient object detection: A benchmark. In *ECCV*, 2012.
- [6] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *IEEE TPAMI*, 34(7):1312–1328, 2012.
- [7] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: Internet image montage. *ACM TOG*, 28(5):124:1–10, 2009.
- [8] T. Chen, P. Tan, L.-Q. Ma, M.-M. Cheng, A. Shamir, and S.-M. Hu. Poseshop: Human image database construction and personalized content synthesis. *IEEE TVCG*, 19(5), 2013.
- [9] W. Chen, C. Xiong, and J. J. Corso. Actionness ranking with lattice conditional ordinal random fields. In *CVPR*, 2014.
- [10] M.-M. Cheng, N. J. Mitra, X. Huang, and S.-M. Hu. Salientshape: Group saliency in image collections. *The Visual Computer*, pages 1–10, 2013.
- [11] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Salient object detection and segmentation. Technical report, Tsinghua Univ., 2011. (TPAMI-2011-10-0753).
- [12] M.-M. Cheng, J. Warrell, W.-Y. Lin, S. Zheng, V. Vineet, and N. Crook. Efficient salient region detection with soft image abstraction. In *IEEE ICCV*, pages 1529–1536, 2013.
- [13] M.-M. Cheng, F.-L. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu. RepFinder: Finding Approximately Repeated Scene Elements for Image Editing. *ACM TOG*, 29(4):83:1–8, 2010.
- [14] M.-M. Cheng, G.-X. Zhang, N. J. Mitra, X. Huang, and S.-M. Hu. Global contrast based salient region detection. In *CVPR*, pages 409–416, 2011.
- [15] M.-M. Cheng, S. Zheng, W.-Y. Lin, J. Warrell, V. Vineet, P. Sturges, N. Crook, N. Mitra, and P. Torr. ImageSpirit: Verbal guided image parsing. *ACM TOG*, 2014.
- [16] Y. S. Chia, S. Zhuo, R. K. Gupta, Y.-W. Tai, S.-Y. Cho, P. Tan, and S. Lin. Semantic colorization with internet images. *ACM TOG*, 30(6):156:1–156:8, 2011.

- [17] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, volume 1, pages 886–893, 2005.
- [18] T. Dean, M. A. Ruzon, M. Segal, J. Shlens, S. Vijayanarasimhan, and J. Yagnik. Fast, accurate detection of 100,000 object classes on a single machine. In *CVPR*, 2013.
- [19] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.
- [20] R. Desimone and J. Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 1995.
- [21] I. Endres and D. Hoiem. Category independent object proposals. In *ECCV*, pages 575–588, 2010.
- [22] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *IEEE TPAMI*, to appear.
- [23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. <http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html>.
- [24] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.
- [25] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE TPAMI*, 32(9):1627–1645, 2010.
- [26] D. A. Forsyth, J. Malik, M. M. Fleck, H. Greenspan, T. Leung, S. Belongie, C. Carson, and C. Bregler. *Finding pictures of objects in large collections of images*. Springer, 1996.
- [27] J. P. Gottlieb, M. Kusunoki, and M. E. Goldberg. The representation of visual salience in monkey parietal cortex. *Nature*, 391(6666):481–484, 1998.
- [28] S. Hare, A. Saffari, and P. H. Torr. Efficient online structured output learning for keypoint-based object tracking. In *CVPR*, pages 1894–1901, 2012.
- [29] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *NIPS*, pages 545–552, 2006.
- [30] H. Harzallah, F. Jurie, and C. Schmid. Combining efficient object localization and image classification. In *ICCV*, 2009.
- [31] J. He, J. Feng, X. Liu, T. Cheng, T.-H. Lin, H. Chung, and S.-F. Chang. Mobile product search with bag of hash bits and boundary reranking. In *CVPR*, pages 3005–3012, 2012.
- [32] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *ECCV*, pages 30–43, 2008.
- [33] X. Hou and L. Zhang. Saliency detection: A spectral residual approach. In *CVPR*, pages 1–8, 2007.
- [34] S.-M. Hu, T. Chen, K. Xu, M.-M. Cheng, and R. R. Martin. Internet visual media processing: a survey with graphics and vision applications. *The Visual Computer*, pages 1–13, 2013.
- [35] H. Huang, L. Zhang, and H.-C. Zhang. Arcimboldo-like collage using internet images. *ACM TOG*, 30, 2011.
- [36] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE TPAMI*, 20(11):1254–1259, 1998.
- [37] T. Judd, F. Durand, and A. Torralba. A benchmark of computational models of saliency to predict human fixations. Technical report, MIT tech report, 2012.
- [38] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurobiology*, 4:219–227, 1985.
- [39] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, pages 1–8, 2008.
- [40] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille. The secrets of salient object segmentation. In *CVPR*, 2014.
- [41] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, T. X., and S. H. Y. Learning to detect a salient object. *IEEE TPAMI*, 2011.
- [42] Y.-F. Ma and H.-J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *ACM Multimedia*, 2003.
- [43] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, pages 1–8, 2008.
- [44] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *IJCV*, 60(1):63–86, 2004.
- [45] R. P. and K. J. . R. E. Generating object segmentation proposals using global and local search. In *CVPR*, 2014.
- [46] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *CVPR*, pages 733–740, 2012.
- [47] H. Teuber. Physiological psychology. *Annual Review of Psychology*, 6(1):267–296, 1955.
- [48] J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 2013.
- [49] K. E. van de Sande, J. R. Uijlings, T. Gevers, and A. W. Smeulders. Segmentation as selective search for object recognition. In *ICCV*, pages 1879–1886, 2011.
- [50] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009.
- [51] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *IEEE TPAMI*, 34(3):480–492, 2012.
- [52] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001.
- [53] X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, 2013.
- [54] J. M. Wolfe and T. S. Horowitz. What attributes guide the deployment of visual attention and how do they do it? *Nature Reviews Neuroscience*, pages 5:1–7, 2004.
- [55] Q. Yan, L. Xu, J. Shi, and J. Jia. Hierarchical saliency detection. In *CVPR*, 2013.
- [56] G.-X. Zhang, M.-M. Cheng, S.-M. Hu, and R. R. Martin. A shape-preserving approach to image resizing. *Computer Graphics Forum*, 28(7):1897–1906, 2009.
- [57] Z. Zhang, J. Warrell, and P. H. Torr. Proposal generation for object detection using cascaded ranking svms. In *CVPR*, pages 1497–1504, 2011.
- [58] S. Zheng, M.-M. Cheng, J. Warrell, P. Sturgess, V. Vineet, C. Rother, and P. Torr. Dense semantic image segmentation with objects and attributes. In *IEEE CVPR*, 2014.
- [59] S. Zheng, P. Sturgess, and P. H. S. Torr. Approximate structured output learning for constrained local models with application to real-time facial feature detection and tracking on low-power devices. In *IEEE FG*, 2013.
- [60] Y. Zheng, X. Chen, M.-M. Cheng, K. Zhou, S.-M. Hu, and N. J. Mitra. Interactive images: Cuboid proxies for smart image manipulation. *ACM TOG*, 2012.