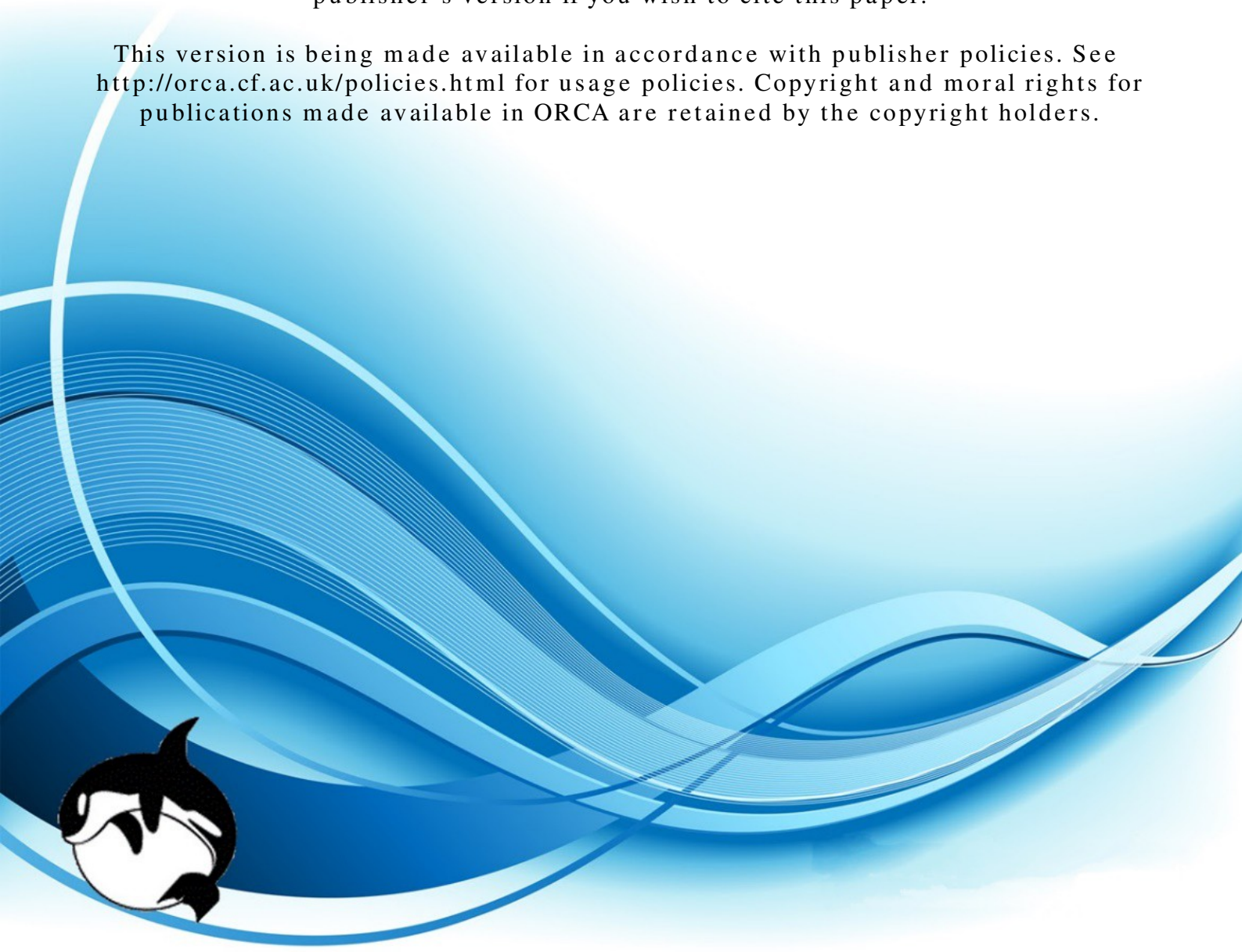


Computational Visual Media
DOI 10.1007/s41095-xxx-xxxx-x Vol. x, No. x, month year, xx–xx

Article Type (Research/Review)

BING: Binarized Normed Gradients for Objectness Estimation at 300fps

Ming-Ming Cheng[1]✉, Yun Liu[1], Wen-Yan Lin[2], Ziming Zhang[3], Paul L. Rosin[4] and Philip Torr[5]

Abstract Training a generic objectness measure to produce object proposals has recently become a hot topic. 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 8×8 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 the proposal localization quality while maintain efficiency, we propose a novel fast segmentation method and demonstrate its effectiveness for improving BING's localization performance, when used in the multi-thresholding straddling expansion (MTSE) post-processing. In experiments on the challenging PASCAL VOC2007 dataset, using 10^3 proposals per image and IoU threshold 0.5, our proposal method achieves 95.6% object detection rate (DR) and 78.6% mean average best overlap (MABO) within 0.005 second per image.

Keywords Object proposals, objectness, generic proposals, efficient method, visual attention, category agnostic proposals.

1 CCCE, Nankai University, Tianjin, China, 300350. https://mmcheng.net/
2 Institute for Infocomm Research, Singapore, 138632
3 MERL, Cambridge, MA 02139-1955, U.S
4 Cardiff University, Wales, CF24 3AA, U.K
5 University of Oxford, Oxford OX1 3PJ, U.K
∗ The first two authors contributed equally to this paper.

1 Introduction

As suggested in the pioneering research [3, 4], *objectness* is usually represented as a value which reflects how likely an image window covers an object of *any category*. A generic objectness measure has great potential to be used as a pre-filtering process for many vision tasks, including object detection [32, 33, 38], visual tracking [52, 77], object discovery [22, 47], semantic segmentation [5, 9], content aware image retargeting [73], and action recognition [71]. Especially for object detection, proposal based detectors have dominated recent state-of-the-art performance. Compared with sliding windows, objectness measures can significantly improve: i) computational efficiency by reducing the search space, and ii) 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 **high object detection rate** (DR), as any undetected objects at this stage cannot be recovered later;
- gain high **proposal localization accuracy** which is measured by the average best overlap (ABO) for each object in each class and the mean average best overlap (MABO) across all classes;
- obtain **high computational efficiency** so that the method can be easily incorporated in various applications, especially for realtime and large-scale applications;
- produce **a small number of proposals** for reducing computational time of subsequent precessing;
- have **good generalization ability** to unseen object categories, so that the proposals can be reused by various of vision tasks without category biases.

To the best of our knowledge, no prior method can

satisfy all these ambitious goals simultaneously.

Research from cognitive psychology [74, 79] and neurobiology [25, 48] suggests 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 visual 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 visual 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 [4, 31, 40] although the visibility of these boundaries depends on the characteristics of the background 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 gradient (see Fig. 1 and Sec. 3), after resizing of their corresponding image windows to a small fixed size (*e.g.* 8×8). Therefore, in order to efficiently quantify the objectness of an image window, we resize it to 8×8 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**), can be used for efficient objectness estimation of image windows, which requires only a few atomic CPU operations (*i.e.* 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 [4, 26, 75] which seek increasingly sophisticated features to obtain greater discrimination.

The original conference version of BING [19] has received much attention. Its efficiency and high detection rates makes BING a good choice in a large number of successful applications that requires *category independent object proposals* [53, 62, 64, 78, 80–82]. Recently, deep neural network based object proposal generation methods have become very popular due to their high recall and computational efficiency, *e.g.* RPN [70], YOLO900 [68] and SSD [58]. 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

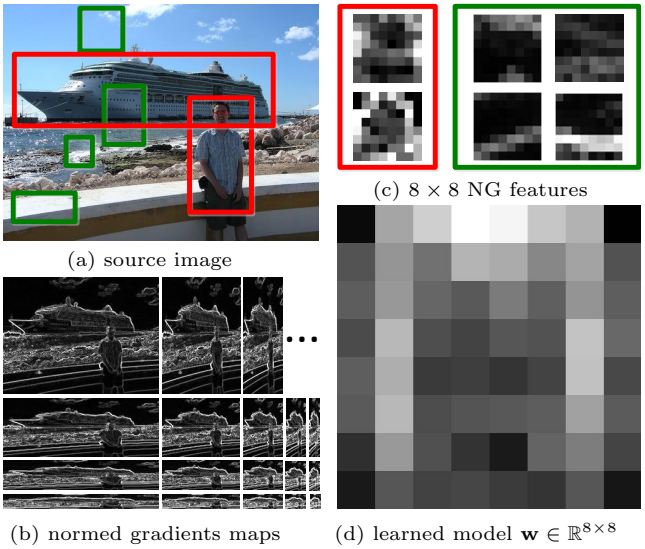


Fig. 1 Although object (red) and non-object (green) windows present huge variation in image space (a), at proper scales and aspect ratios which correspond to a small fixed size (b), their corresponding normed gradients, *i.e.* a NG feature (c), share strong correlation. We learn a single 64D linear model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

object proposals of RPN are highly related to the training data: when training it on the PASCAL VOC dataset [27], the trained model will aim to only detect the 20-classes objects in PASCAL VOC and perform poorly on another dataset like MS COCO (see Sec. 5.4). Its poor generalization ability has restricted its usage, so *RPN is usually only used in object detection*. By contrast, BING is built based on low-level cues about enclosed boundaries and thus can produce ***category independent object proposals***, which has demonstrated applications in multi-label image classification [78], semantic segmentation [64], video classification[81], co-salient object detection [82], deep multi instance learning [80], and video summarisation [53]. However, several researchers [41, 65, 86, 90] have noted that BING's proposal localization is weak.

This manuscript further improves the proposal localization of the conference version [19] by applying multi-thresholding straddling expansion (MTSE) [15] as a postprocessing step. The 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 approach starts with a GPU version of the SLIC method [2, 69], to quickly obtain initial seed regions (superpixels) by performing oversegmentation. A region merging process is then performed based on the average pixel distance. We replace [30] in MTSE

with this novel grouping method [16], and dub the new proposal system BING-E.

We have extensively evaluated our objectness methods on the PASCAL VOC2007 [27] and Microsoft COCO [56] datasets. The experimental results show that our method efficiently (300fps for BING and 200fps 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 1,000 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% MABO. Feeding the proposals to the fast R-CNN [32] framework for an object detection task, BING-E achieves 67.4% mean average precision (mAP). Following [4, 26, 75], 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 [4, 26, 44, 49, 50, 61, 65–67, 75, 85, 90], our method achieves competitive performance using a smaller set of proposals, while being 100-1,000 times faster than them. Thus, our proposed method achieves significantly high efficiency while obtaining state-of-the-art generic object proposals. These performances fulfill the previously stated requirements for 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 human eye movement [8, 46]. Inspired by neurobiology research about early primate visual system, Itti *et al.*[45] proposed one of the first computational models for saliency detection, which estimates center-surrounded differences across multi-scale image features. Ma and Zhang [60] proposed a fuzzy growing model to analyze local contrast based saliency. Harel et al. [36] proposed normalizing center-surrounded feature maps for highlighting conspicuous parts. Although fixation point prediction models have achieved remarkable development, the prediction results tend to highlight edges and corners rather than the entire objects. Thus, these models are not suitable for generating generic object proposals.

Salient object detection models try to detect the most attention-grabbing object in a scene, and then segment the whole extent of that object [6, 7, 55]. Liu *et al.*[57] 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.*[18] 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 [63]. Such salient object segmentation for simple images achieved great success in image scene analysis [20, 54, 87], content aware image editing [83, 89], and it can be used as a cheap tool to process a large number of Internet images or build robust applications [12, 13, 21, 37, 42, 43] by automatically selecting good results [17, 18]. However, these approaches are less likely to work for complicated images where many objects are presented and they are rarely dominant (*e.g.* PASCAL VOC images).

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 [4, 26, 75]. Producing rough segmentations [10, 26] 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 [10, 26] are very computationally expensive. Alexe *et al.*[4] proposed a cue integration approach to get better prediction performance more efficiently. Broadly speaking, there are two main categories of object proposal generation methods: 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 that are expected to represent objects or object-parts, merging together some regions is likely to find complete objects. A large literature has focused on this aspect. Uijlings *et al.*[75] 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.*[65] proposed a multi-scale segmenter to generate segmentation hierarchies, and then explored the combinatorial space of these hierarchical regions to produce high-quality object proposals. Some other well-known algorithms [26, 50, 61, 66, 67] fall into this category as well.

Edge based object proposal generation approaches use edges to explore where in an image the complete objects occur. As pointed out in [4], complete objects usually have well-defined closed boundaries in space. Some methods have achieved high performance using this intuitive cue. Zitnick *et al.*[90] proposed a simple box objectness score that measured the number of contours wholly enclosed by a bounding box. They generated object bounding box proposals directly from edges in an efficient way. Lu *et al.*[59] proposed a closed contour measure that is defined using closed path integral. Zhang *et al.*[85] proposed a cascaded ranking SVM approach with an oriented gradient feature for efficient proposal generation.

Generic object proposals can be widely used in object detection [32, 33, 38], visual tracking [52, 77], video classification [81], pedestrian detection [62], content aware image retargeting [73], and action recognition [71]. 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 100-1,000 times faster than most popular alternatives [4, 26, 75] (see Sec. 5).

3 BING for Objectness Measure

Inspired by the ability of the human visual system which efficiently perceives objects before identifying them [25, 48, 74, 79], we introduce a simple 64D norm of the gradients (NG) feature (Sec. 3.1), as well as its binary approximation, *i.e.* the 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 set of *quantized window sizes* (scales and aspect ratios[1]). Each window is scored with a linear model $\mathbf{w} \in \mathbb{R}^{64}$ (Sec. 3.2),

$$s_l = \langle \mathbf{w}, \mathbf{g}_l \rangle, \tag{1}$$

$$l = (i, x, y), \tag{2}$$

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.*[86] show that this choice of window sizes along with the NMS is close to optimal. Some sizes (*e.g.* 10×500) are less likely than others (*e.g.* 100×100) to contain an object instance. Thus we define the objectness score (*i.e.* the calibrated filter score) as

$$o_l = v_i \cdot s_l + t_i, \tag{3}$$

where $v_i, t_i \in \mathbb{R}$ are learnt coefficient and bias terms for each quantised size i (Sec. 3.2). Note that calibration using Eq. (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 [4, 31, 40] although the visibility of these boundaries depends on the characteristics of the background of occluding foreground objects. When resizing windows corresponding to real world objects to a small fixed size (*e.g.* 8×8, 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 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 do share clear similarity 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 8×8 region of these resized normed gradients maps are defined as a 64D ***normed gradients (NG)***[2] 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. And these insensitive 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,

[1] 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 8×8 windows in the downsized images (from which we extract features), correspond to target windows.

[2] The *normed gradient* represents Euclidean norm of the gradient.

thus having great potential to be involved in realtime applications.

The cost of introducing such advantages to the NG feature is the loss of discriminative ability. However, this is not a problem as BING can be used as a pre-filter, and the resulting false-positives will be processed and eliminated by subsequent category specific detectors. In Sec. 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.

3.2 Learning objectness measurement with NG

To learn an objectness measure of image windows, we follow the two stage cascaded SVM approach [85].

Stage I. We learn a single model $\mathbf{w}$ for Eq. (1) using a linear SVM [28]. 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 Eq. (3) using a linear SVM [28], 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 Sec. 5 for evaluation criteria).

Discussion. As illustrated in Fig. 1d, the learned linear model $\mathbf{w}$ (see Sec. 5 for experimental settings), looks similar to the multi-size center-surrounded patterns [45] hypothesized as biologically plausible architecture of primates [34, 48, 79]. The large weights along the borders of $\mathbf{w}$ favor a boundary that separates an object (center) from its background (surround). Compared to manually designed center surround patterns [45], 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 binary model approximation [35, 88], 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 $\mathbf{w} \in \mathbb{R}^{64}$ 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

Algorithm 1 Binary approximate model $\mathbf{w}$ [35].

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**
  $\mathbf{a}_j$ =sign(ε)
  $\beta_j = \langle \mathbf{a}_j, \varepsilon \rangle / \|\mathbf{a}_j\|^2$ (project ε onto $\mathbf{a}_j$)
  $\varepsilon \leftarrow \varepsilon - \beta_j \mathbf{a}_j$ (update residual)
end for

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 [35]),

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

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 $\mathbf{g}_l$ 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 their corresponding bit position in the BYTE values.

Naively getting an 8×8 BING feature requires a loop computing access to 64 positions. By exploring two special characteristics of an 8×8 BING feature, we develop a fast BING feature calculation algorithm (Alg. 2), which enables using atomic updates (BITWISE SHIFT and BITWISE OR) to avoid computing the loop. First, 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. 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

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

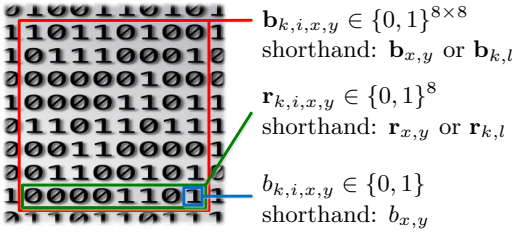


Fig. 2 Illustration of variables: a BING feature $\mathbf{b}_{x,y}$, its last row $\mathbf{r}_{x,y}$ and last element $b_{x,y}$. Notice that the subscripts i, x, y, l, k, introduced in Eq. (2) and Eq. (5), are locations of the whole vector rather than index of vector element. We can use a single atomic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

$b_{x,y}$ 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 [76]. Instead of calculating a single scalar value over an arbitrary rectangle range [76], our method uses a few atomic operations (*e.g.* ADD, BITWISE, *etc.*) to calculate *a set of binary patterns* over an 8×8 fixed range.

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

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

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 kernel $[-1, 0, 1]$ to find image gradients g_x and g_y in the 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.

4 Enhancing BING with Region Cues

BING is not only very efficient, but also can achieve high object detection rate. However, when considering ABO or MABO, its performance is disappointing. When further applying BING to some object detection frameworks which use object proposals as input, like fast-RCNN, the detection rate is also bad. This situation suggests BING does not obtain good proposal localization quality.

Two reasons may cause this phenomenon. On the 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, but the problem is that the shapes of objects are varied, which means that the closed boundaries of objects will be mapped to different positions in the fixed size windows. So 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 the unsatisfactory localization quality caused by above reasons, we consider multi-thresholding straddling expansion (MTSE) [15], which is an effective method for refining 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 for each box. By this means, each bounding box covers tightly a set of internal superpixels, and thus the localization quality of proposals is significantly improved. However, MTSE algorithm is too slow and the bottleneck is segmentation [30]. Considering this situation, we use a new fast image segmentation method [16] to replace the segmentation method in MTSE.

Recently, SLIC [2] has become a popular superpixel generation method because of its efficiency, and the GPU version of SLIC (*i.e.* gSLICr) [69] can achieve a fast speed of 250fps. 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 apply 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 value of all the pixels in this superpixel, and the distance between two adjacent nodes is computed using the Euclidean distance of color values. Finally, we feed these nodes into the graph-based segmentation method to produce the final image segmentation [16].

We employ the full MTSE pipeline which is modified to use our new segmentation algorithm, and manage to reduce the computation time from 0.15 second down to 0.0014 second per image. Incorporating this improved version of MTSE as a post processing enhancement step

of BING, we obtain a new proposal system, and call it BING-E.

5 Evaluation

We extensively evaluate our method on the challenging PASCAL VOC2007 [27] and Microsoft COCO [56] datasets. PASCAL VOC2007 contains 20 object categories, and consists of training, validation and test sets, with 2501, 2510 and 4952 images respectively and corresponding bounding box annotations. We use the training set to train our BING model and test on the test set. Microsoft COCO consists of 82783 images for training and 40504 images for validation, which contains about 1M annotated instances from 80 categories. COCO is more challenging because of its large size and complex image contents.

We compare against various competitive methods: EdgeBoxes [90][3], CSVM [85][4], MCG [65] [5], RPN [70][6], Endres [26], Objectness [4], GOP [49], LPO [50], Rahtu [66], RandomPrim [61], Rantalankila [67], and SelectiveSearch [75] [7] using publicly available code. All the parameters of these method are set using default values, except for [49], in which we employ (180,9) as highlighted on the author's homepage. To make the comparison fair, all the methods except the deep learning based RPN [70] are tested on the same device with an Intel i7-6700k CPU and a NVIDIA GeForce GTX 970 GPU, and data parallelization is enabled. For RPN, we utilize an NVIDIA GeForce GTX TITAN X GPU for computation. Since objectness is often used as a preprocessing step to reduce the number of windows subsequent processing needs to consider, too many proposals are contrary to this principle. Therefore, we only use the top 1000 proposals for comparison. In order to evaluate the generalization ability of each method, we test them on the COCO validation dataset using the same parameters as on VOC2007 without retraining. Since there are at least 60 categories in COCO different to those in VOC2007, using COCO to test the generalization ability of the proposal methods is a good choice.

[3] https://github.com/pdollar/edges

[4] https://zimingzhang.wordpress.com/.

[5] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/mcg/.

[6] https://github.com/rbgirshick/py-faster-rcnn

[7] We download the code of other methods from [11] https://github.com/Cloud-CV/object-proposals.

	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

Tab. 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.

(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

Tab. 2 The average result quality (DR using 1000 proposals) of BING at different approximation levels, measured by N_w and N_g in Sec. 3.3. N/A represents without binarization.

5.1 Experimental Setup

Discussion of BING. As shown in Tab. 1, with the binary approximation to the learned linear filter (Sec. 3.3) 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 equivalent to $O(N)$, 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 neighboring positions (*i.e.* 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 of at least 0.5, while the max IoU overlap with ground truth for the negative sampling boxes is less than 0.5. In addition, some window sizes whose aspect ratios are too large are ignored because the number of training samples in VOC2007 for each of them is too small (less than 50). Our training on 2501 images (VOC2007) takes only 20 seconds (excluding xml loading time). We further illustrate in Tab. 2 how different approximation levels influence the result quality. According to this comparison, in all further experiments we use $N_w = 2$, $N_g = 4$.

Implementation details of BING-E. In the implementation of BING-E, we find that removing some small BING windows, with $W_o < 30$ or $H_o <$

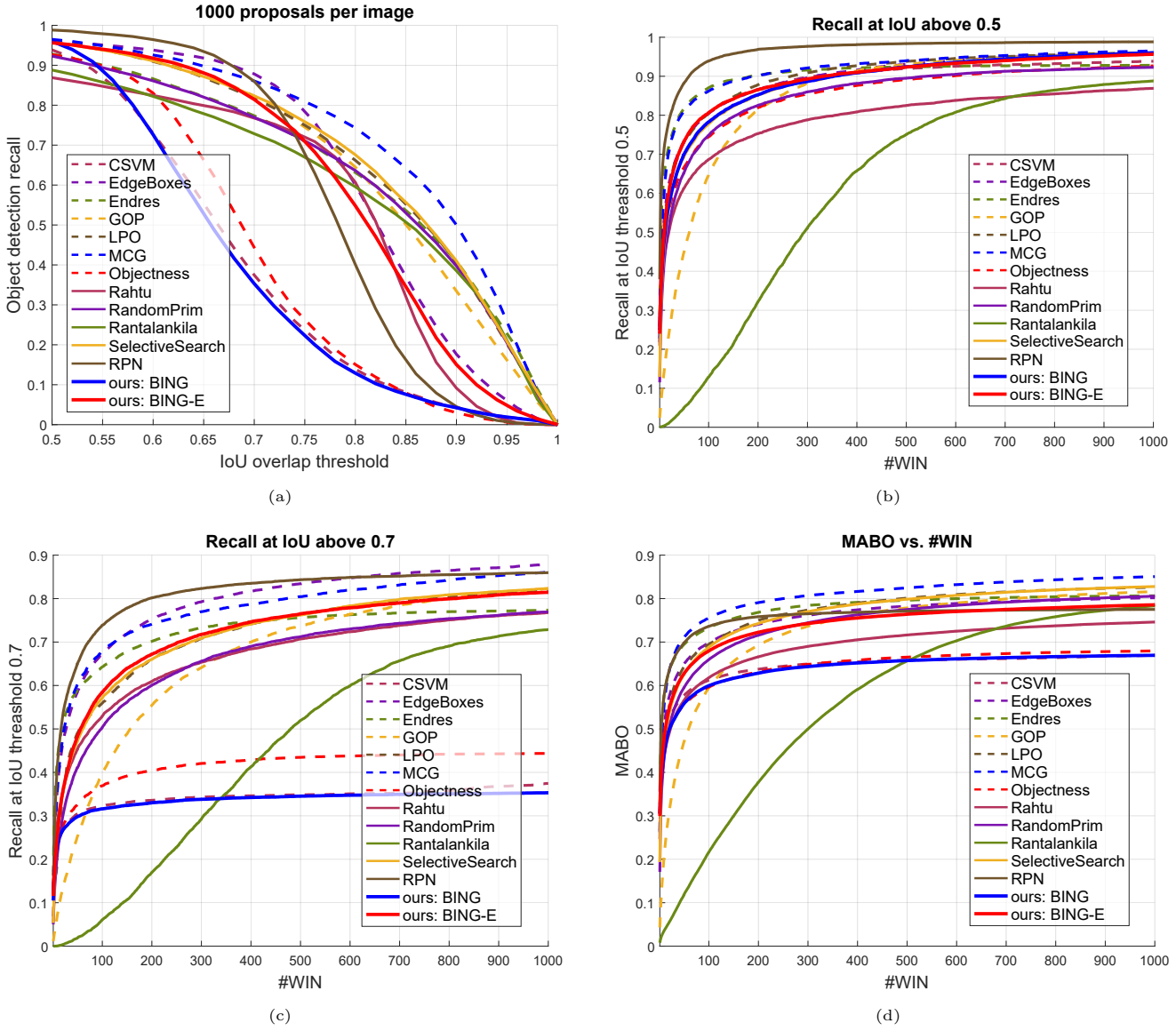


Fig. 3 Testing results on PASCAL VOC2007 test set: (a) object detection recall versus IoU overlap threshold; (b, c) recall versus the number of candidates at IoU threshold 0.5 and 0.7 respectively; (d) MABO versus the number of candidates using at most 1000 proposals.

30, hardly degrades the proposal quality of BING-E while reducing the runtime spent on BING process by half. When using gSLICr [69] to segment images into superpixels, we set the expected size of superpixels to 4×4. In the graph-based segmentation system [16, 30], we use the scale parameter $k = 120$, and the minimum count of superpixels in each produced segment is set to 6. We utilize the default multi-thresholds of MTSE, *i.e.* $\{0.1, 0.2, 0.3, 0.4, 0.5\}$. After refinement, non-maximal suppression (NMS) is performed to obtain the final boxes, where the IoU threshold of NMS is set to 0.8. All the following experiments use these settings.

5.2 PASCAL VOC2007

As demonstrated by [4, 75], 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 higher efficiency than traditional methods. Thus, we first compare our method with some competitors using detection recall metrics. Fig. 3 (a) show detection recall when varying the IoU overlap threshold using 1,000 proposals. EdgeBoxes and MCG outperform many other methods in all cases. RPN achieves

Methods \ #WIN	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	**87.8**	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	**93.9**	**98.4**	**98.8**	**73.9**	**84.3**	86.0	0.10
BING	78.3	92.4	96.2	31.6	34.5	35.3	**0.0033**
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

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

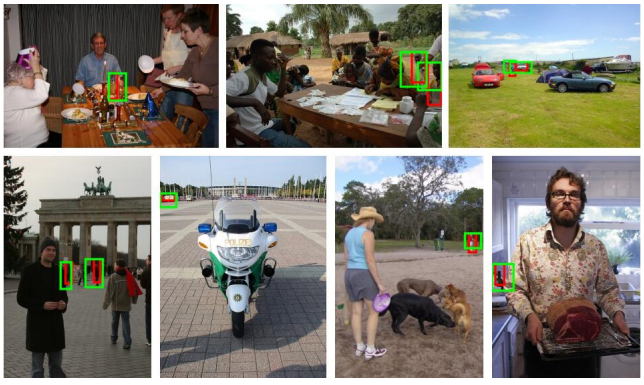


Fig. 4 Some failure examples of BING-E. Failure means that the overlap between the best detected box (green) and ground truth (red) is less than 0.5. All images are from the VOC2007 test set.

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 is close to the best performance. It should be emphasized that both BING and BING-E are **more than two orders of magnitude (*i.e.* 100+)** faster than most popular alternatives [26, 65, 75, 90] (see details in Tab. 3). The performance of BING and CSVM [85] 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.

Fig. 3 (b)-(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. Especially when the number of candidates is sufficient, BING and BING-E outperform all other methods. In the subfigure (e), the recall curve of BING drops a lot, and the same behavior appears in the MABO evaluation. This may be because the proposal localization quality of BING is poor. However, note that the performance of BING-E is consistently close to the best performance, indicating that BING's localization problem has been overcome.

We show numeric comparison of recall *vs.* #WIN in Tab. 3. BING-E always performs better than most of the competitors. Both 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. By 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 after applying some postprocessing steps. Tab. 4 shows the ABO & MABO comparison of these competitors. MCG always outperforms others with a big gap, and BING-E is competitive with all the methods except MCG.

Since proposal generation is usually a preprocessing step in vision tasks, we feed candidate boxes produced by objectness methods into the fast R-CNN [32] object detection framework to test the effectiveness of proposals in practical applications. The CNN model of fast R-CNN is retrained using boxes from the respective methods. Tab. 5 shows the evaluation results. In terms of mAP (mean average precision), the overall detection rates across all the methods are quite close to each other. RPN performs slightly better, and our BING-E method is 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. Synthesizing the effects of various factors, BING-E achieves a significantly high speed while obtaining state-of-the-art generic object proposals. Finally, we illustrate sample results with varied complexity for

Methods		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 **83.2** 79.3 76.2 70.0 **88.1** **81.6** **89.9** **79.6** **84.6** **88.6** 88.5 **84.4** **83.2** **78.2** **74.6** **82.8** **91.0** **86.6** **85.8**	**85.1**
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	**81.8** 82.4 **79.8** **77.5** 62.8 84.0 78.0 89.8 76.5 82.9 87.1 **89.1** 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 **70.7** 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
ours:BING	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
ours:BING-E	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

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

Methods		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	**73.4** 78.1 68.4 **55.7** 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 **70.6** 76.5 66.1	66.0
LPO	67.4 76.9 **68.8** 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 **69.8** 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 **74.5** 66.4 79.8 82.2 76.2 65.5 35.2 65.6 70.1 **77.4** 65.9	67.4
RPN	67.5 **78.5** 67.3 51.9 **51.5** 76.2 **79.8** **84.4** **50.2** 74.3 66.9 **83.2** 80.0 73.9 **76.5** **37.1** **69.4** 65.7 76.5 **74.2**	**69.2**
ours:BING	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
ours:BING-E	69.3 78.3 66.5 55.0 39.0 **81.7** 75.9 83.9 39.6 74.4 67.5 80.1 **83.7** **76.3** 67.0 35.2 67.2 68.8 75.8 61.7	67.4

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

VOC2007 test images using our improved BING-E method in Fig. 5 to better demonstrate our high quality proposals.

5.3 Discussion on PASCAL VOC2007

In order to perform further analysis, we divide the ground truths into different sets according to their window sizes, and test some of the most competitive methods on these sets. Tab. 6 shows the results. When the ground truth area is small, BING-E performs much worse than others. As the ground truth area increases, the gap between BING-E and other state-of-the-art methods is gradually narrowing, and BING-E outperforms all of them on the metric of recall when the area is larger than 2^{12}. Fig. 4 shows some failure examples of BING-E. Note that almost all the false detected objects are small. These small objects may have blurry boundaries that make them be hard to distinguish from the 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 applying MCG into the fast R-CNN framework. The fast R-CNN uses the VGG16 [72] model, in which the convolutional layers 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. So using MCG proposals to retrain the CNN model may confuse the network because of the detected small object proposals. Thus, MCG does not achieve the best performance in the object detection task although it

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

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

Fig. 5 Illustration of true positive object proposals for VOC2007 test images using our method (BING-E).

outperforms others on recall and MABO metrics.

5.4 Microsoft COCO

In order to test the generalization ability of these proposal methods, we extensively evaluate them on the COCO validation set using the same parameters as on the VOC2007 dataset without retraining. Since the dataset is too large, we only compare against some efficient methods.

Fig. 6 (a) show object detection recall versus IoU overlap threshold using different numbers of proposals. MCG always dominates the performance, but its low speed makes it impossible for many vision applications. EdgeBoxes performs well when the IoU threshold is

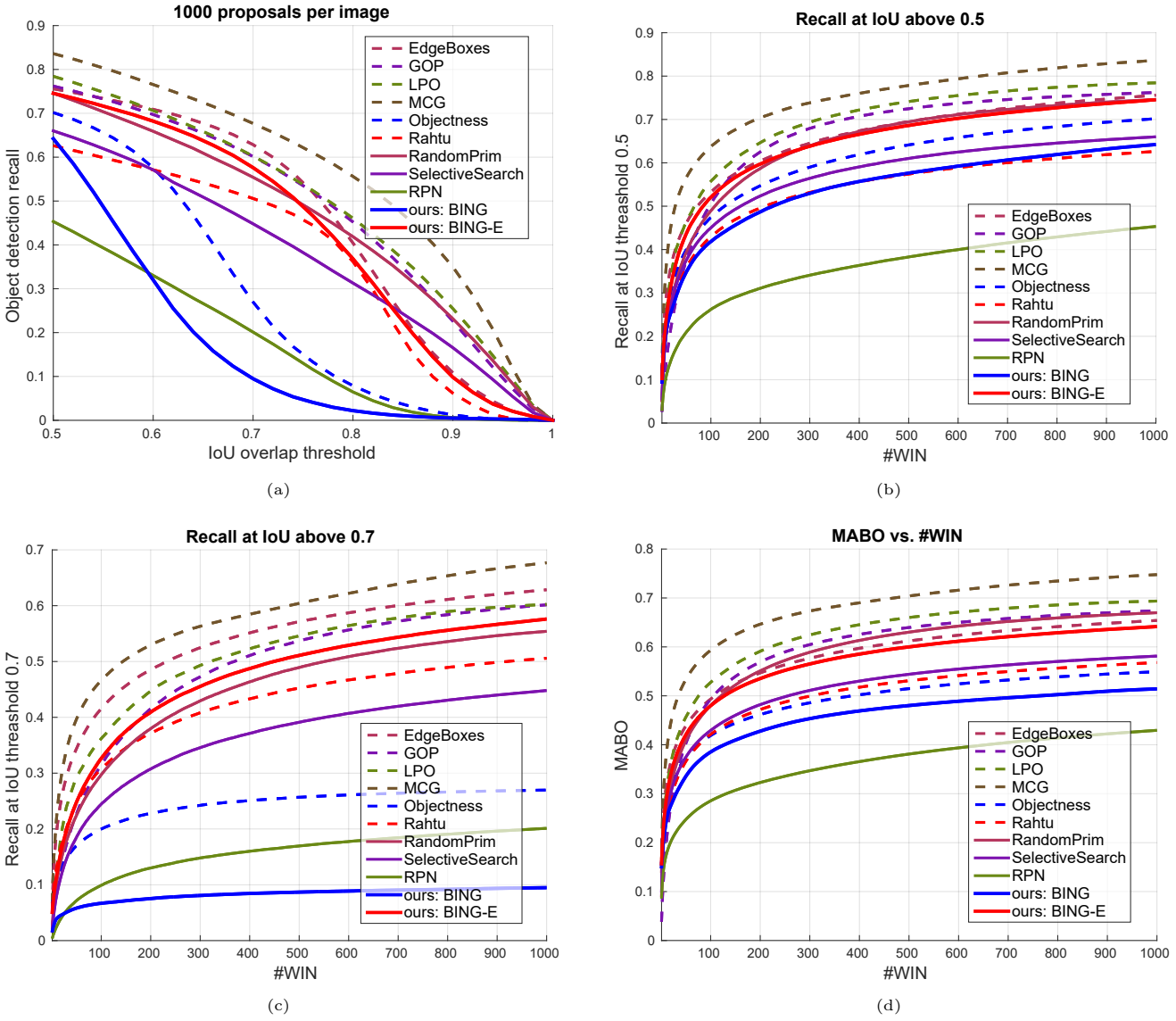


Fig. 6 Testing results on COCO validation dataset: (a) object detection recall versus IoU overlap threshold; (b, c) recall versus the number of candidates at IoU threshold 0.5 and 0.7 respectively; (d) MABO versus the number of candidates using at most 1000 proposals.

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. Note that RPN performs very poorly on COCO, which means it is highly dependent on the training data. As addressed in [11], 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.

Fig. 6 (b)-(d) show the recall/MABO when varying the number of proposals. The key observation is also that RPN suffers a big drop in performance over VOC2007. Its recall at IoU 0.5 and MABO are even worse than BING. In addition, our proposed BING and BING-E are very robust when transferring to different object classes. Tab. 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 if the increased computational load

Methods \ #WIN	IoU=0.5			IoU=0.7			MABO
	100	500	1000	100	500	1000	(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	**63.8**	**77.8**	**83.6**	**46.6**	**60.4**	**67.7**	**74.8**
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
ours:BING	41.8	57.6	64.2	6.7	8.7	9.5	51.4
ours:BING-E	52.1	68.6	74.6	32.6	51.1	57.6	64.2

Tab. 7 Detection recall (%) using different IoU threshold and #WIN on COCO validation set.

is acceptable.

6 Conclusion and Future Work

We present a surprisingly simple, fast, and high quality objectness measure by using 8×8 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. To improve the localization quality of BING, we further propose 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 with a significantly high speed. The evaluations also demonstrate that BING is a good basis for object proposal generation.

Limitations. BING and BING-E predict a small set of object bounding boxes. Thus, they share similar limitations as all other bounding box based objectness measure methods [4, 85] and classic sliding window based object detection methods [23, 29]. For some object categories (*e.g.* a snake, wires, *etc.*), a bounding box might not localize the object instances as accurately as a segmentation region [10, 26, 67].

Future work. The high quality and efficiency of our method make it suitable for many realtime vision applications and large scale image collections (e.g. ImageNet [24]). In particular, the binary operations and memory efficiency make our BING method suitable to run on low power devices [35, 88]. Our speed-up strategy by 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 computation bottleneck of proposal based vision tasks such as object detection methods [32, 39], enabling potential realtime high quality object detection.

We have demonstrated how to generate a small set (*e.g.* 1,000) 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 [51, 84], and explore more applications [14, 53, 64, 78, 80–82] using BING and BING-E. To encourage future works, we will continuously make the updated source code available at `http://mmcheng.net/bing`.

Acknowledgements

This research was supported by NSFC (NO. 61572264, 61620106008).

References

[1] R. Achanta, S. Hemami, F. Estrada, and S. Süsstrunk. Frequency-tuned salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2009.

[2] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk. SLIC superpixels compared to state-of-the-art superpixel methods. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 34(11):2274–2282, 2012.

[3] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 73–80, 2010.

[4] B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 34(11), 2012.

[5] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik. Semantic segmentation using regions and parts. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3378–3385. IEEE, 2012.

[6] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient object detection: A survey. *ArXiv e-prints*, 2014.

[7] A. Borji, M.-M. Cheng, H. Jiang, and J. Li. Salient

object detection: A benchmark. *IEEE Transactions on Image Processing*, 24(12):5706–5722, 2015.

[8] A. Borji, D. Sihite, and L. Itti. Quantitative analysis of human-model agreement in visual saliency modeling: A comparative study. *IEEE Transactions on Image Processing*, 2012.

[9] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu. Semantic segmentation with second-order pooling. In *European Conference on Computer Vision*, pages 430–443, 2012.

[10] J. Carreira and C. Sminchisescu. CPMC: Automatic object segmentation using constrained parametric min-cuts. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 34(7):1312–1328, 2012.

[11] N. Chavali, H. Agrawal, A. Mahendru, and D. Batra. Object-proposal evaluation protocol is 'gameable'. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 835–844, 2016.

[12] T. Chen, M.-M. Cheng, P. Tan, A. Shamir, and S.-M. Hu. Sketch2photo: Internet image montage. *ACM Transactions on Graphics*, 2009.

[13] 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 Transactions on Visualization and Computer Graphics*, (5), 2013.

[14] W. Chen, C. Xiong, and J. J. Corso. Actionness ranking with lattice conditional ordinal random fields. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[15] X. Chen, H. Ma, X. Wang, and Z. Zhao. Improving object proposals with multi-thresholding straddling expansion. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2587–2595, 2015.

[16] M.-M. Cheng, Y. Liu, Q. Hou, J. Bian, P. Torr, S.-M. Hu, and Z. Tu. Hfs: Hierarchical feature selection for efficient image segmentation. In *European Conference on Computer Vision*, pages 867–882, 2016.

[17] 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.

[18] M.-M. Cheng, N. J. Mitra, X. Huang, P. H. S. Torr, and S.-M. Hu. Global contrast based salient region detection. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 37(3):569–582, 2015.

[19] M.-M. Cheng, Z. Zhang, W.-Y. Lin, and P. H. S. Torr. BING: Binarized normed gradients for objectness estimation at 300fps. In *IEEE CVPR*, 2014.

[20] M.-M. Cheng, S. Zheng, W.-Y. Lin, V. Vineet, P. Sturgess, N. Crook, N. J. Mitra, and P. Torr. Imagespirit: Verbal guided image parsing. *ACM TOG*, 34(1):3, 2014.

[21] 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 Transactions on Graphics*, 30(6):156:1–156:8, 2011.

[22] M. Cho, S. Kwak, C. Schmid, and J. Ponce. Unsupervised object discovery and localization in the wild: Part-based matching with bottom-up region proposals. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1201–1210, 2015.

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

[24] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255, 2009.

[25] R. Desimone and J. Duncan. Neural mechanisms of selective visual attention. *Annual review of neuroscience*, 18(1):193–222, 1995.

[26] I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 36(2):222–234, 2014.

[27] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman. The PASCAL visual object classes (VOC) challenge. *International Journal on Computer Vision*, 88(2):303–338, 2010.

[28] 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.

[29] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, pages 1627–1645, 2010.

[30] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *International Journal on Computer Vision*, 59(2):167–181, 2004.

[31] 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. In *International Workshop on Object Representation in Computer Vision*, pages 335–360. Springer, 1996.

[32] R. Girshick. Fast R-CNN. In *IEEE International Conference on Computer Vision*, pages 1440–1448, 2015.

[33] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 580–587, 2014.

[34] J. P. Gottlieb, M. Kusunoki, and M. E. Goldberg. The representation of visual salience in monkey parietal cortex. *Nature*, 391(6666):481–484, 1998.

[35] S. Hare, A. Saffari, and P. H. Torr. Efficient online structured output learning for keypoint-based object tracking. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1894–1901, 2012.

[36] J. Harel, C. Koch, and P. Perona. Graph-based visual saliency. In *Neural Information Processing Systems*, pages 545–552, 2006.

[37] 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 *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3005–3012, 2012.

[38] K. He, X. Zhang, S. Ren, and J. Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. In *European Conference on Computer Vision*, pages 346–361. Springer, 2014.

[39] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016.

[40] G. Heitz and D. Koller. Learning spatial context: Using stuff to find things. In *European Conference on Computer Vision*, pages 30–43. 2008.

[41] J. Hosang, R. Benenson, P. Dollr, and B. Schiele. What makes for effective detection proposals? *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 38(4):814–830, 2016.

[42] 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.

[43] H. Huang, L. Zhang, and H.-C. Zhang. Arcimboldo-like collage using internet images. *ACM Transactions on Graphics*, 30, 2011.

[44] A. Humayun, F. Li, and J. M. Rehg. RIGOR: Reusing inference in graph cuts for generating object regions. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 336–343, 2014.

[45] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 1998.

[46] T. Judd, F. Durand, and A. Torralba. A benchmark of computational models of saliency to predict human fixations. Technical report, MIT tech report, 2012.

[47] C. Kading, A. Freytag, E. Rodner, P. Bodesheim, and J. Denzler. Active learning and discovery of object categories in the presence of unnameable instances. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 4343–4352, 2015.

[48] C. Koch and S. Ullman. Shifts in selective visual attention: towards the underlying neural circuitry. *Human Neurbiology*, 4:219–227, 1985.

[49] P. Krähenbühl and V. Koltun. Geodesic object proposals. In *European Conference on Computer Vision*, pages 725–739, 2014.

[50] P. Krähenbühl and V. Koltun. Learning to propose objects. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1574–1582, 2015.

[51] W. Kuo, B. Hariharan, and J. Malik. Deepbox: Learning objectness with convolutional networks. In *IEEE International Conference on Computer Vision*, pages 2479–2487, 2015.

[52] S. Kwak, M. Cho, I. Laptev, J. Ponce, and C. Schmid. Unsupervised object discovery and tracking in video collections. In *IEEE International Conference on Computer Vision*, pages 3173–3181, 2015.

[53] Y. J. Lee and K. Grauman. Predicting important objects for egocentric video summarization. *International Journal on Computer Vision*, 114(1):38–55, 2015.

[54] K. Li, Y. Zhu, J. Yang, and J. Jiang. Video super-resolution using an adaptive superpixel-guided auto-regressive model. *Pattern Recognition*, 51:59–71, 2016.

[55] Y. Li, X. Hou, C. Koch, J. Rehg, and A. Yuille. The secrets of salient object segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[56] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755, 2014.

[57] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, T. X., and S. H.Y. Learning to detect a salient object. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 2011.

[58] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*, 2016.

[59] C. Lu, S. Liu, J. Jia, and C.-K. Tang. Contour box: rejecting object proposals without explicit closed contours. In *IEEE International Conference on Computer Vision*, pages 2021–2029, 2015.

[60] Y.-F. Ma and H.-J. Zhang. Contrast-based image attention analysis by using fuzzy growing. In *ACM Multimedia*, 2003.

[61] S. Manen, M. Guillaumin, and L. Van Gool. Prime object proposals with randomized Prim's algorithm. In *IEEE International Conference on Computer Vision*, pages 2536–2543, 2013.

[62] S. Paisitkriangkrai, C. Shen, and A. v. d. Hengel. Pedestrian detection with spatially pooled features and structured ensemble learning. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 38(6):1243–1257, 2016.

[63] F. Perazzi, P. Krähenbühl, Y. Pritch, and A. Hornung. Saliency filters: Contrast based filtering for salient region detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 733–740, 2012.

[64] P. O. Pinheiro and R. Collobert. From image-level to pixel-level labeling with convolutional networks. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1713–1721, 2015.

[65] J. Pont-Tuset, P. Arbelaez, J. T. Barron, F. Marques, and J. Malik. Multiscale combinatorial grouping for image segmentation and object proposal generation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 39(1):128–140, 2017.

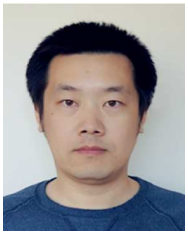
[66] E. Rahtu, J. Kannala, and M. B. Blaschko. Learning a category independent object detection cascade. In *IEEE International Conference on Computer Vision*, pages 1052–1059, 2011.

[67] P. Rantalankila, J. Kannala, and E. Rahtu. Generating object segmentation proposals using global and local search. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2417–2424, 2014.
[68] J. Redmon and A. Farhadi. Yolo9000: better, faster, stronger. *arXiv preprint arXiv:1612.08242*, 2016.
[69] C. Y. Ren, V. A. Prisacariu, and I. D. Reid. gSLICr: SLIC superpixels at over 250hz. *arXiv preprint arXiv:1509.04232*, 2015.
[70] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Neural Information Processing Systems*, pages 91–99, 2015.
[71] F. Sener, C. Bas, and N. Ikizler-Cinbis. On recognizing actions in still images via multiple features. In *European Conference on Computer Vision*, pages 263–272. Springer, 2012.
[72] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations(ICLR)*, 2015.
[73] J. Sun and H. Ling. Scale and object aware image retargeting for thumbnail browsing. In *IEEE International Conference on Computer Vision*, pages 1511–1518. IEEE, 2011.
[74] H. Teuber. Physiological psychology. *Annual Review of Psychology*, 6(1):267–296, 1955.
[75] J. R. Uijlings, K. E. van de Sande, T. Gevers, and A. W. Smeulders. Selective search for object recognition. *International Journal on Computer Vision*, 104(2):154–171, 2013.
[76] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2001.
[77] N. Wang, S. Li, A. Gupta, and D.-Y. Yeung. Transferring rich feature hierarchies for robust visual tracking. *arXiv preprint arXiv:1501.04587*, 2015.
[78] Y. Wei, W. Xia, M. Lin, J. Huang, B. Ni, J. Dong, Y. Zhao, and S. Yan. HCP: A flexible CNN framework for multi-label image classification. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 38(9):1901–1907, 2016.
[79] 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.
[80] J. Wu, Y. Yu, C. Huang, and K. Yu. Deep multiple instance learning for image classification and auto-annotation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3460–3469, 2015.
[81] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained CNN architectures for unconstrained video classification. In *British Machine Vision Conference*, 2015.
[82] D. Zhang, J. Han, C. Li, J. Wang, and X. Li. Detection of co-salient objects by looking deep and wide. *International Journal on Computer Vision*, pages 1–18, 2016.
[83] 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.
[84] Z. Zhang, Y. Liu, X. Chen, Y. Zhu, M.-M. Cheng, V. Saligrama, and P. H. Torr. Sequential optimization for efficient high-quality object proposal generation. *IEEE Transactions on Pattern Recognition and Machine Intelligence*, 2017.
[85] Z. Zhang, J. Warrell, and P. H. Torr. Proposal generation for object detection using cascaded ranking SVMs. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.
[86] Q. Zhao, Z. Liu, and B. Yin. Cracking BING and beyond. In *British Machine Vision Conference*, 2014.
[87] 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.
[88] 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.
[89] 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 Transactions on Graphics*, 2012.
[90] C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *European Conference on Computer Vision*, pages 391–405, 2014.

Ming-Ming Cheng received his PhD degree from Tsinghua University in 2012. Then he did 2 years 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 includes computer graphics, computer vision, and image processing.

Yun Liu is a Ph.D. Candidate with College of Computer Science and Control Engineering, Nankai University, under the supervision of Prof. Ming-Ming Cheng. His major research interest is computer vision and machine learning.

Wen-Yan Lin received his PhD 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 PhD 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 and 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 the PhD 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.