BING: Binarized Normed Gradients for Objectness Estimation at 300fps

Ming-Ming Cheng[1] Ziming Zhang[2] Wen-Yan Lin[3] Philip Torr[1]
[1]The University of Oxford [2]Boston University [3]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 8×8 *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.* 8×8). 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**) 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 achieves better DR using a smaller set of proposals, is much simpler and 1000+ times faster, while being able to predict *unseen* categories. This fulfills afore mentioned 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[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. Some sizes (*e.g.* 10×500) are less likely than others to contain an object instance (*e.g.* 100×100). Thus we define the objectness score (*i.e.* calibrated filter score) as

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

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.

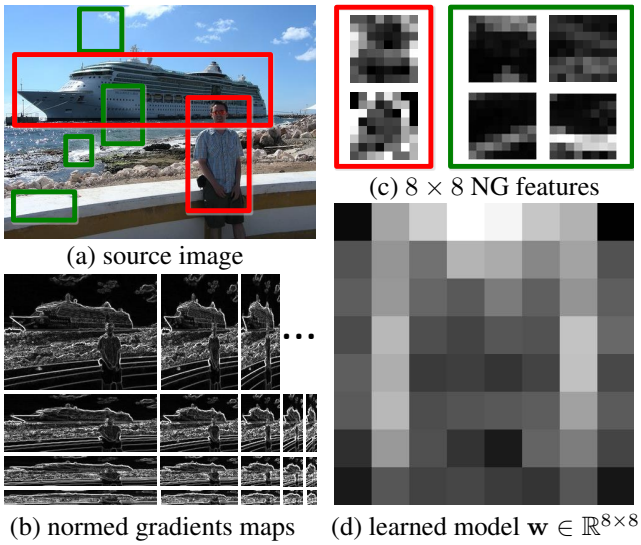
3.1. Normed gradients (NG) and objectness

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.* 8×8, 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 8×8 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 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.

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

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



(a) source image (b) normed gradients maps (c) 8×8 NG features (d) learned model $\mathbf{w} \in \mathbb{R}^{8\times8}$

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**
  $\mathbf{a}_j$ =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

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]),

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

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

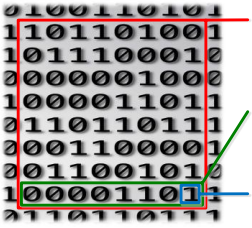


Figure 2. Illustration of variables: a BING feature $\mathbf{b}_{x,y}$, its last row $\mathbf{r}_{x,y}$ and last element $b_{x,y}$. Notice that the subscripts i, x, y, l, k, introduced in (2) and (5), are locations of the whole vector rather than index of vector element. We can use a single atomic variable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

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

Notice that these BING features have different weights according to its corresponding bit position in 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 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 8×8 fixed range.

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

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

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 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 [3] 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[4] 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

[3]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.

[4]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/.

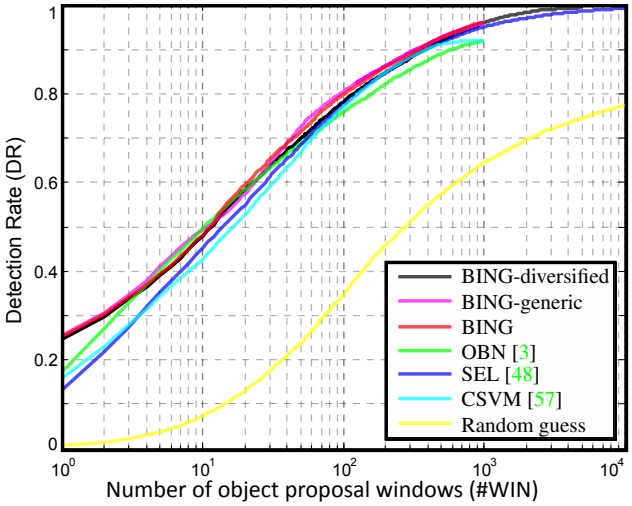


Figure 3. Tradeoff between #WIN and DR for different methods. Our method achieves 96.2% DR using 1,000 proposals, and 99.5% DR using 5,000 proposals. The 3 methods [3, 48, 57] have been evaluated on the same benchmark and shown to outperform other alternative proposal methods [6, 21, 25, 30, 50], saliency 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	**0.003**

Table 1. Average computational time on VOC2007.

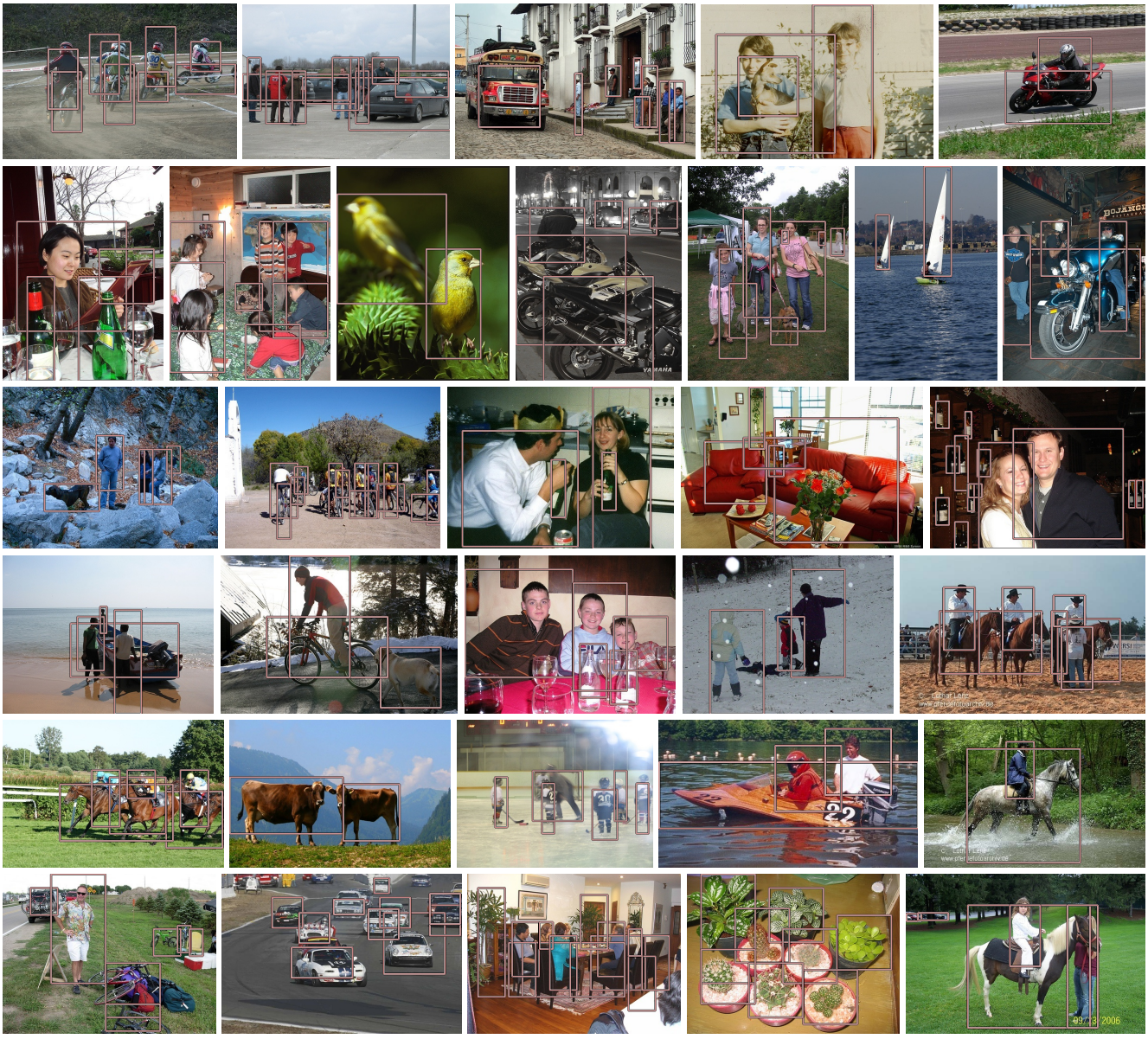


Figure 4. Illustration of the true positive object proposals for VOC2007 test images. See Fig. 3 for statistical results.

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 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

	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.

(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 8×8 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. Sturgess, 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 Neurbiology*, 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.