Image Processing

Report of

" Edge Boxes: Locating Object Proposals from Edges （ECCV,2014）"

学号	班级	姓名	签名	打分	
3012216040	2 班	罗　琦			
3012216083	3 班	王雨朦			
3012216106	4 班	刘雨森			

2015 年 6 月 20 日

Paper Report of "Edge Boxes: Locating Object Proposals from Edges(ECCV, 2014)"

Abstract

In this report, first we introduce the main work and algorithm of the paper"Edge Boxes: Locating Object Proposals from Edges (ECCV, 2014, by Zitnick, C.L., Dollar, P.)". Since the source code of the system can be downloaded form GitHub, we then run it, adjust different parameters and try a huge amount of images for verification, comparison and deeply thinking. After discussion, several deficiency of the system and potential ways that can improve it have been proposed. We tried some of the feasible measures and the effect is quite desirable.

1 Background

The paper isn't the original paper we picked. What interests us is edge detection, however, when surfing on the internet for the PDF format of the paper "Structured Forests for Fast Edge Detection (ICCV,2013)". We surprisingly found this paper, which is based on the original one and mainly talked about how do they use edge boxes to locate object proposals, a new idea that no one did before. So we chose this one because of the latest date, the authority of the meeting, and most significantly, our curiosity.

2 Approach of the Paper

In this paper they propose Edge Boxes, a novel approach to generating object bounding box proposals directly from edges, because edges provide a simplified but informative representation of an image similar to segments.

They observe that: the number of contours wholly enclosed by a bounding box is indicative of the likelihood of the box containing an object. Therefore, it is demonstrate In the paper that scoring a box based on the number of contours it wholly encloses creates a surprisingly effective proposal measure. Moreover, rather than measuring the number of straddling contours, they instead remove such contours from consideration.

The process of the object proposal generator are summarized as follows:

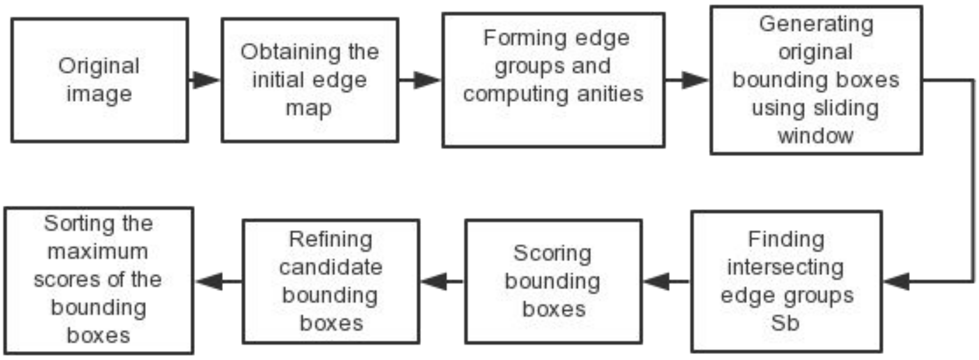

Fig.1The process of the object proposal generator

Edge Map Obtaining： Given an image, initially compute an edge response using Structured Edge detector (Recently proposed by the writer themselves in “Structured forests for fast edge detection” and “Fast edge detection using structured forests”). Then perform Non-Maximal Suppression (NMS) orthogonal to the edge response to find edge peaks, the result is a sparse edge map.

In the sparse edge map, each pixel p having an edge magnitude m_p and orientation θ_p. They define edges as pixels with $m_p > 0.1$ to threshold the edges.

Edge Groups and affinities： The edge groups is formed using a simple greedy approach that combines 8-connected edges until the sum of their orientation differences is above a threshold $(\pi/2)$, small groups are merged with neighboring groups.

Given a set of edge groups $s_i \in S$, for each pair of neighboring groups a pair of groups s_i and s_j, the affinity is computed based on their mean positions x_i and x_j and mean orientations θ_i and θ_j.

$$a(s_i, s_j) = \left|\cos(\theta_i - \theta_{ij})\cos(\theta_j - \theta_{ij})\right|^{\gamma}$$

Where θ_{ij} is the angle between x_i and x_j. The value of γ may be used to adjust the affinity's sensitivity to changes in orientation, with $\gamma = 2$ used in practice.

Original bounding boxes generating： Original bounding boxes is generated by using a sliding window search over position, scale and aspect ratio. The step size for each is determined using a single parameter α, indicating the *IoU* for neighboring boxes. *IoU* computes the intersection of two bounding box divided by the area of their union. After discussion, a value of $\alpha = 0.65$ is ideal. The scale values range from a minimum box area of $\sigma = 1000$ pixels to the full image. The aspect ratio varies from $\frac{1}{\tau}$ to τ, where $\tau = 3$ is used in practice.

Finding intersecting edge groups： Intersect edge groups is the set of edge groups S_b that overlap the box b's boundary. Creating the two data structures to be L_r and K_r to find S_b along a horizontal boundary from pixel (c_0, r) to (c_1, r) of a bounding box. The vertical boundaries can be handled in a similar manner.

L_r is an ordered list created by storing the order in which the edge groups occur along the row r. K_r with the same size as the width of the image is created that stores the corresponding index into L_r for each column c in row r. Thus, if pixel p at location (c, r) is a member of edge group s_i, $L_r(K_r(c)) = i$. Since most pixels do not belong to an edge group, using these two data structures we can efficiently find the list of overlapping edge groups by searching L_r from index $K_r(c_0)$ to

$K_r(c_1)$.

Scoring bounding boxes： For each group s_i we compute a continuous value $\omega_b(s_i) \in [0,1]$ that indicates whether s_i is wholly contained in b.

$$\omega_b(s_i) = \begin{cases} 0 & , \quad s_i \in S_b or \overline{x_i} \notin b \\ 1 - \max\limits_T \prod\limits_j^{|T|-1} a(t_j, t_{j+1}) & , \quad s_i \notin S_b and \overline{x_i} \in b \\ 1 & , \quad else \end{cases}$$

where T is an ordered path of edge groups with a length of $|T|$ that begins with some $t_1 \in S_b$ and ends at $t_{|T|} \in s_i$.

Using the computed values of ω_b, we define our score using:

$$h_b = \frac{\sum_i w_b(s_i) m_i}{2(b_w + b_h)^k}$$

Where b_w and b_h are the box's width and height, m_i is the sum of the magnitudes m_p for all edges p in the group s_i.

Finally, it has been observed that the edges in the center of the box are of less importance than those near the box's edges. We can subtract the edge magnitudes from a box b^{in} centered in b:

$$h_b^{in} = h_b - \frac{\sum_{p \in b^{in}} m_p}{2(b_w + b_h)^k}$$

where the width and height of bin is $b_w / 2$ and $b_h / 2$ respectively.

Refining candidate bounding boxes： Refinement is performed using a greedy iterative search to maximize h_b^{in} over position, scale and aspect ratio. After each iteration, the search step is reduced in half. The search is halted once the translation step size is less than 2 pixels. Once the candidate bounding boxes are refined, their maximum scores are recorded and sorted.

3 Code Implementation

The source code is published on GitHub (https://github.com/pdollar/edges). We run it mainly under the guide of "readme.txt".

4 Advantages & Disadvantages

After discussing the method, adjusting different parameters and trying a huge amount of images, the advantages and disadvantages of the system can be concluded as follows.

❖ Advantages

a. There are a lot of candidate boxes. they nearly include most objects. With a ground truth box, there is a high *IoU* matched bounding box among a lot of candidate bounding boxes. AS can be seen in Fig.2.

Fig.2 The result of matching ground truth box

b. The approach is significantly faster and more accurate than previous approaches. The author compare the run-time and summary statistics of the approach to other methods in Table 1. The run-time for Edge Boxes includes the 0.1 seconds needed.

	AUC	N@25%	N@50%	N@75%	Recall	Time
BING [11]	.20	292	–	–	29%	**.2s**
Rantalankila [10]	.23	184	584	–	68%	10s
Objectness [4]	.27	27	–	–	39%	3s
Rand. Prim's [8]	.35	42	349	3023	80%	1s
Rahtu [7]	.37	29	307	–	70%	3s
Selective Search [5]	.40	28	199	1434	**87%**	10s
CPMC [6]	.41	15	111	–	65%	250s
Edge boxes 70	**.46**	**12**	**108**	**800**	**87%**	**.25s**

Table 1. The result of comparing

❖ Disadvantages

Still, the method has some drawbacks.

a. If the image is too garish to point out its object, the image will be totally selected as an object proposal. As shown in Fig.3, it is obviously that the biggest box gains a highest score. So in this kind of images, the algorithm is useless.

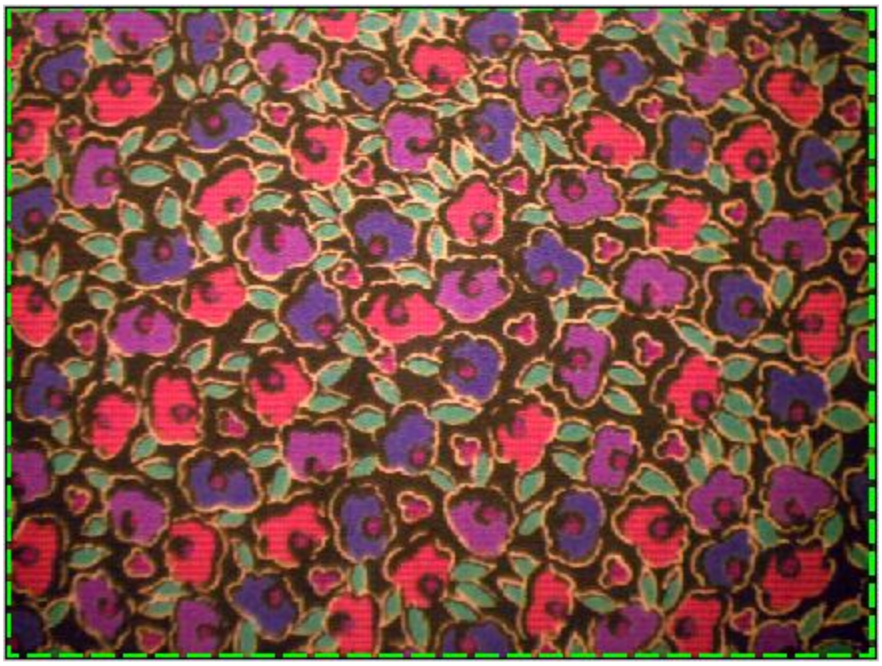

Fig.3 Bad result of garish image

b. As shown in the follow, the two biggest score of boxes are those which contain light, while the third scored box contains face. Because after structure edge detecting, face is formed of a little of edge groups, and the box must be large enough to enclose a face. So object which consists of several edge groups, can not gain high score. We can not devise an algorithm approach to face recognition based on the box scoring.

Fig4. Small object gains higher score.

c. In the paper, the authors measure the accuracy of a bounding box using the Intersection over Union (*IoU*) metric. They draw some ground truth bounding boxes which exactly contain objects. Then they select some bounding boxes, which have high *IoU* with the grounding truth bounding boxes, as object proposals from the candidate bounding boxes. Thus they can judge the accuracy from the matched rate of the ground truth bounding boxes. The result is quite good, shown as Table 2.

	IoU = 0.5		IoU = 0.7		IoU = 0.9				
	AUC	Recall	AUC	Recall	AUC	Recall	Runtime	α	β
Edge boxes 50	**.64**	**96%**	.36	55%	.04	5%	**.25s**	.65	.55
Edge boxes 70	.58	89%	**.45**	**76%**	.06	9%	**.25s**	.65	.75
Edge boxes 90	.38	59%	.28	46%	**.15**	**28%**	2.5s	.85	.95

Table 2 Accuracy of a bounding box

But they haven’t considered the errors in bounding box placement. In most cases, some candidate bounding boxes are far related with objects.

5 Improvement

a. The process of refining candidate bounding boxes

In refining candidate bounding boxes, the author try to broaden or compress the candidate bounding boxes to gain a higher box scoring. In each search step, he changes the height and the width of the box by changing the position of the box's boundaries. He changes one boundary at once, and score the changed box. If the new score is higher than the original score, the original box will be modified to the changed box. After traverse four edges, then search step is reduced in half. The search is halted once the search step is less than 2 pixels.

In this case, what if we alter the order of changing each boundary in every search step? To gain a better result, we alter the order in two form. The result is shown as follow.

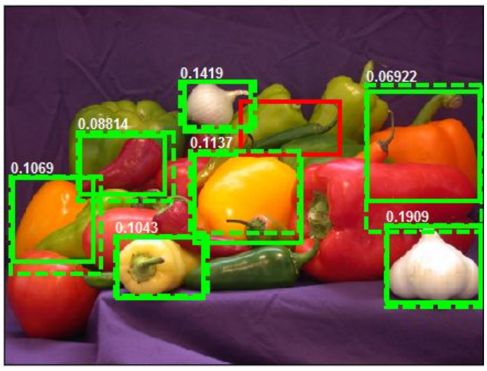

Fig.5 The result of object proposal with original order (up-down-left-right boundary)

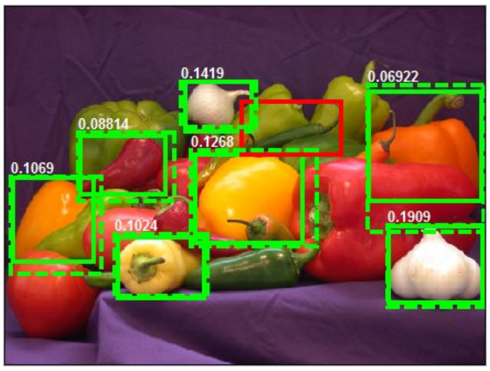

Fig.6 The result of object proposal with changed order (left-right-up-down boundary)

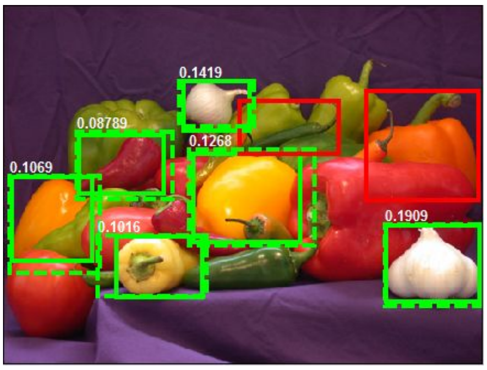

Fig.7 the result of object proposal with changed order (right-left-down-up boundary)

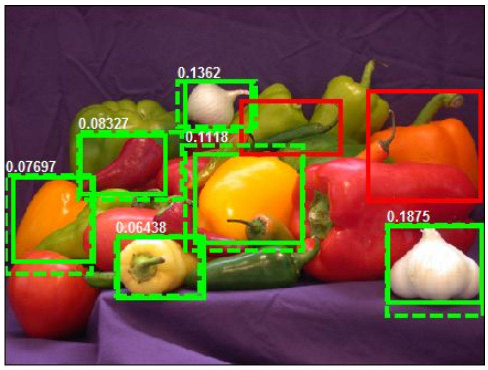

Fig.8 The result of object proposal with changed order (4 boundaries togather)

From the Fig. and the Fig., We can see that the order will effect the result. it is hard to evaluate the efficiency of the changes. So we choose a maximum score of the four changes. If the maximum score is higher than then original one, modify the original box to the maximum. Thus, we can gain Fig.. We can see that there are five boxes gain a better result, although losing one matched box.

b. Generating segmentation proposals

In the discussion, the author propose an direction for future work that is using the edges to help generate segmentation proposals in addition to the bounding box proposals for objects. We intended to complete the work. but the author has finished it in the latest code version. The result is shown as follow.

Fig.9 Segmentation proposals

Reference

[1]Dollar, P., Zitnick, C.L.: Fast edge detection using structured forests. CoRR abc / 1406.5549 (2014)

[2]Dollar, P., Zitnick, C.L.: Structured forests for fast edge detection. In: ICCV. (2013)