OBJECT DETECTION REFINEMENT USING MARKOV RANDOM FIELD BASED PRUNING AND LEARNING BASED RESCORING

Naoto Inoue, Ryosuke Furuta, Toshihiko Yamasaki, Kiyoharu Aizawa

Department of Information and Communication Engineering, The University of Tokyo

ABSTRACT

Contextual information such as the co-occurrence of objects and the location of objects has played an important role in object detection. We present candidate pruning and object rescoring methods that leverage contextual information and that can improve the state-of-the-art CNN-based object detection methods such as Fast R-CNN and Faster R-CNN. In our pruning method, we formulate candidate reduction as a Markov random field optimization problem. In our rescoring method, we employ a machine learning technique to reconsider the detection scores of candidate windows. We experimentally demonstrate improvements in R-CNN-based object detection methods using two datasets. Moreover, we apply our model to the structured retrieval task to show the potential applications of our model.

Index Terms— object detection, contextual model, MRF

1. INTRODUCTION

Object detection is one of the most challenging problems in computer vision that has drawn a large amount of attention recently. Although state-of-the-art approaches such as [1, 2, 3] show high detection accuracy using the powerful recognition capabilities of convolutional neural networks (CNN), one important disadvantage is that in classifying each window, information outside the window, called the context, is not considered. It has been demonstrated that considering the co-occurrence of objects [4], the spatial relationship between objects [5], and semantic information [6] are useful. Based on these concepts, [7, 8] proposed improving the deformable part model (DPM) [9] by pruning candidate windows based on context. [7, 8] also show that explicitly modeling contextual information is useful for scene understanding tasks such as structured retrieval. Although their methods are effective when using the DPM, they cannot improve the detection accuracy of recent R-CNN-based methods [2, 3] because their accuracy is already quite high; there is little room for improvement based on previous pruning methods [7, 8] (discussed in Sec. 3.2). In addition, their methods only output a binary prediction, i.e., whether the window is correctly detected or not. There are other methods [10, 11] that consider context. However, [10, 11] cannot be applied to R-CNN-based methods because their technique is specific to DPM. More recently, Redmon *et al.* proposed YOLO [12], which removes background false positives for Fast R-CNN. However, YOLO cannot be directly applied to structured retrieval because YOLO does not explicitly consider contexts.

In this paper, we present candidate pruning and window rescoring methods based on context to achieve better object detection in terms of improving both mean average precision (mAP) and F1 with fewer candidate windows. Unlike those in [7, 8], our method rescores the candidate windows considering context and is effective, even for recent R-CNN-based methods. Our approach can be integrated into most object recognition frameworks as a post-processing step, which is an advantage as compared to [12], because in [12] the contextual information is implicitly included in the end-to-end detection pipeline. We approximate the distribution of the candidate windows and the spatial and scale relationship between candidate windows to calculate the likelihood. Candidate pruning is achieved by constructing a markov random field (MRF) model considering co-occurrence, spatial, and scale priors of the objects. The window rescoring is done with SVM using spatial, scale prior, co-occurrence, and global contexts as features. Our experimental results using Fast R-CNN [2] show that our approach can improve mAP from 66.9% to 67.3% and F1 from 3.5% to 26.2% for VOC2007 [13], and mAP from 32.3% to 33.0% and F1 from 8.4% to 11.0% for MSCOCO [14]. We also applied our method to Faster R-CNN [3]. Moreover, we apply our contextual model to structured retrieval to show the potential applications of our model.

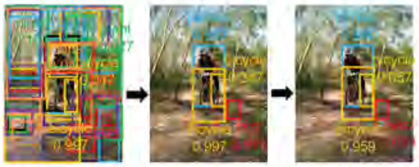

object detector output
pruning result
rescoring result

Fig. 1: Concept of our approach. These are the actual results.

2. APPROACH

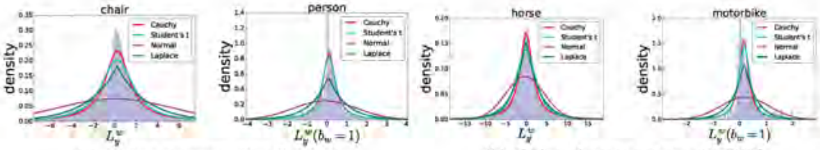
The flowchart of our approach is shown in Fig. 1. Our approach consists of two parts: pruning candidate windows and rescoring candidate windows using the SVM. The pruning technique is designed to remove contextually inconsistent candidate windows using MRF optimization so as to improve F1, although it results in a slight decline in mAP. The rescoring technique is designed to rescore each candidate window based on contextual information so as to improve mAP. We explain the conventional spatial location representation of windows, such as those in [7, 8], in Sec. 2.1. Our major contributions are presented in Secs. 2.2-2.4.

2.1. Spatial location representation of windows

For a given set of images, we run object detectors to obtain an initial set of candidate windows. For window w, c_w indicates the object class label of w. s_w indicates the corresponding object class score of w, thus $0 \leq s_w \leq 1$. l^x_w, l^y_w are the x, y coordinates of the center of

This work is partially financially supported by a Grant-in-Aid for Scientific Research (No. 26540078), the Japan Society for the Promotion of Science (JSPS), and the Support Center for Advanced Telecommunications Technology Research (SCAT), Japan.

978-1-5090-4117-6/17/$31.00 ©2017 IEEE



(a) Object detector: Fast R-CNN

(b) Object detector: Faster R-CNN

Fig. 2: Fitting examples on VOC2007.

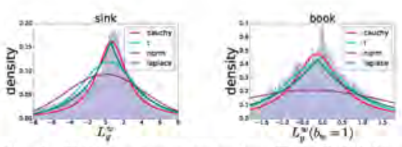


Fig. 3: Fitting examples using Fast R-CNN on MSCOCO.

Table 1: List of features used for the rescoring.

feature	dim
$p(b_w=1 \mid s_w, c_w)$	1
Spatial position (abspos) ($p(b_w=1 \mid L_y^w, c_w)$ in Eq. (2))	1
Spatial position likelihood between windows (relpos) ($l_{pos}(w)$) in Eq. (10))	1
Scale likelihood between windows (scale) ($l_{scale}(w)$) in Eq. (11))	1
$\boldsymbol{p}_{scene}(i)$ (global)	205

w, l_w^w, l_w^h are the width and height of w; we limit $-1 \leq l_w^y \leq 1, 0 \leq l_w^h \leq 2$. We transform these coordinates as in [8].

$$L_x^w = \frac{l_w^x}{l_w^h} H_{c_w}, L_y^w = \frac{l_w^y}{l_w^h} H_{c_w}, L_z^w = \frac{f}{l_w^h} H_{c_w}. \quad (1)$$

Here, f is the distance from the observer to the image plane and is fixed at 1. H_{c_w} is the physical height of an object, which is manually fixed (e.g., person=1.7 m, car=1.5 m, etc.). Because horizontal locations generally have weak contextual information, we ignore L_x^w and only consider L_y^w and L_z^w when capturing vertical location and scale relationships. We model $\log L_z^w$ because L_z^w is always positive and is more heavily distributed around small values. We assign a binary variable b_w to represent whether the classification label is expected to be correct ($b_w=1$) or incorrect ($b_w=0$).

2.2. Spatial and scale prior

R-CNN-based object detection uses region proposal approaches considering "objectness" such as [15] and the distribution of the candidate windows are unknown, unlike in a sliding-window approach, such as that in [9]. We use different distributions to fit L_y^w for each object class. Fig. 2 shows the fitting results on VOC2007 using Fast R-CNN and Faster R-CNN as object detection methods. In addition, we consider MSCOCO [14]. Fig. 3 shows the fitting results for MSCOCO using Fast R-CNN. We observe that, regardless of the dataset and the region proposal based on considering the "objectiveness" approach, the Cauchy distribution has the best fitting result for both L_y^w and L_y^w ($b_w=1$). Considering these results, we can approximate the probability regarding w's vertical position in Eq. (2).

$$p(b_w=1 \mid L_y^w, c_w) = \frac{p(b_w=1 \mid c_w) p(L_y^w \mid b_w=1, c_w)}{p(L_y^w \mid c_w)}. \quad (2)$$

We use $d_{pos}(w, w') = L_y^w - L_y^{w'}$ to consider the likelihood of the relative position of two windows w and w'. We fit the distribution of $d_{pos}(w, w') = L_y^w - L_y^{w'}$ to each combination of $(c_w, c_{w'})$ and $(b_w, b_{w'})$ using the Cauchy distribution. Similarly, we use $d_{scale}(w, w') = \log L_z^w - \log L_z^{w'}$ to consider the likelihood of the relative scale of two windows w and w'. We fit $d_{scale}(w, w') = \log L_z^w - \log L_z^{w'}$ to each combination of $(c_w, c_{w'})$ and $(b_w, b_{w'})$ using the Cauchy distribution. Note that the finding that the objects' distribution can be modeled by the Cauchy distribution is one of our contributions.

2.3. Pruning candidate windows using MRF

Given the candidate windows $\mathbf{W} = \{w_1, w_2, ...\}$ in a single image, we construct a graph to represent the contextually consistent interactions. Second, for a window pair (w, w'), we do not connect the windows if $s_w \leq T \wedge s_{w'} \leq T$ (where T is a threshold constant). We aim to determine the optimal configuration $\boldsymbol{y}* = [y_{w_1}, y_{w_2}, ...]^T$ that minimizes the energy $E(\boldsymbol{y})$, where P indicates the set of connected window pairs, $y_w = 1$ indicates that we expect window w to be correct, and $y_w = 0$ indicates that we expect window w to be incorrect. We prune the candidate windows that have a y_w value of zero. $E(\boldsymbol{y})$ consists of unary and pairwise terms.

$$E(\boldsymbol{y}) = \sum_{w \in W} \Phi(y_w) + \beta \sum_{(w,w') \in P} \Psi(y_w, y_{w'}). \quad (3)$$

We define the unary potential as follows:

$$\Phi_w(y_w) = \begin{cases} 1 - \epsilon & (y_w = 0) \\ 1 - s_w & (y_w = 1). \end{cases} \quad (4)$$

When β is zero, it is equivalent to keeping a window w with a s_w value that is larger than ϵ, where ϵ is a constant. We define the pairwise term as follows:

$$p_{scale}(w, w') = p(b_w = 1, b_{w'} = 1 \mid d_{scale}(w, w'), c_w, c_{w'}), \quad (5)$$

$$p_{pos}(w, w') = p(b_w = 1, b_{w'} = 1 \mid d_{pos}(w, w'), c_w, c_{w'}), \quad (6)$$

$$p_{exist}(w, w') = max(p(b_w = 1 \mid s_w, c_w), p(b_{w'} = 1 \mid s_{w'}, c_{w'})). \quad (7)$$

$$g(w, w') = \sqrt[3]{p_{scale}(w, w') p_{pos}(w, w') p_{exist}(w, w')} \quad (8)$$

$$\Psi(y_w, y_{w'}) = \begin{cases} 0 & (y_w, y_{w'}) = (0, 0) \\ \frac{1 - g(w,w')}{adj(w')} & (y_w, y_{w'}) = (0, 1) \\ \frac{1 - g(w,w')}{adj(w)} & (y_w, y_{w'}) = (1, 0) \\ \frac{g(w,w')}{max(adj(w), adj(w'))} & (y_w, y_{w'}) = (1, 1), \end{cases} \quad (9)$$

where $adj(w)$ is the number of windows connected to w, $g(w, w')$ is a penalty term based on the spatial and scale relation between w and w'. $p(b_w = 1 \mid s_w, c_w)$ is computed using logistic regression, as in [7]. We use QPBO [16] to obtain the global optimal configuration $\boldsymbol{y}* = [y_{w_1}, y_{w_2}, ...]^T$. The proposed window w is rejected when y_w is zero, because it is contextually less probable to be correct. This process is useful for eliminating contextually unreasonable detection results and contributes to improving F1; however, it tends to eliminate some correct detections, resulting in mAP degradation. Therefore, the detection scores are further rescored as in Sec. 2.4.

2.4. Rescoring windows by SVM

The optimal configuration $\boldsymbol{y}* = [y_{w_1}, y_{w_2}, ...]^T$ is obtained for all images in the target dataset using candidate pruning in Sec. 2.3. Then, we rescore each window w where $y_w = 1$ (expected to be correct based on candidate pruning) by considering the context of the outside features of w. We integrate the context using an SVM. A list of additional features is shown in Table 1. Instead of using s_w, we

Table 2: AP [%] and F1 [%] in VOC2007 test using Fast R-CNN.

method	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP	F1
(a) Fast R-CNN [2]	73.8	**78.2**	68.7	54.4	37.0	76.4	78.2	**82.8**	41.4	71.7	67.7	**78.9**	**79.6**	**74.0**	66.9	33.0	63.4	68.9	74.0	68.4	66.9	3.5
(b) [2] + Tree-context [7]	53.0	60.7	63.0	49.3	28.7	37.9	75.8	62.6	40.6	45.1	53.4	68.3	2.4	59.3	65.1	27.8	42.4	51.5	55.6	58.8	50.1	3.5
(c) [2] + HOOD [8]	62.4	62.9	60.8	48.3	31.0	70.2	71.4	71.1	34.9	67.9	60.1	69.6	63.0	61.8	60.4	26.5	44.3	57.8	68.7	64.1	57.9	**67.2**
(d) [2] + threshold	73.8	77.7	68.7	54.4	36.8	75.9	78.2	**82.8**	41.0	71.7	67.7	76.9	78.1	**74.0**	66.9	33.0	62.4	68.9	74.0	67.9	66.5	24.4
(e) [2] + Ours (SVM(all context))	73.2	**78.2**	67.4	53.3	36.8	**76.7**	78.0	79.2	44.5	67.4	68.0	77.8	75.0	73.0	67.1	**37.0**	59.4	67.1	75.1	68.5	66.2	3.5
(f) [2] + Ours (MRF)	73.8	77.7	68.8	54.4	36.8	75.9	78.2	**82.8**	41.0	71.7	67.7	76.9	78.1	**74.0**	66.9	33.0	62.4	68.9	74.0	67.9	66.5	26.2
(g) [2] + Ours (MRF + SVM (scene))	74.7	77.8	68.1	**54.8**	36.9	75.9	**78.3**	79.3	41.7	71.6	69.0	77.0	78.1	**74.0**	66.8	34.8	62.2	**69.9**	74.2	68.1	66.7	26.2
(h) [2] + Ours (MRF + SVM (scene,abspos))	74.7	77.8	68.8	**54.8**	37.2	76.0	**78.3**	81.8	42.9	**71.8**	**70.0**	77.0	78.0	73.5	67.1	35.4	62.8	**69.9**	74.7	68.5	67.0	26.2
(i) [2] + Ours (MRF+SVM (scene,abspos,relpos))	**74.9**	77.8	**69.8**	**54.8**	37.9	76.0	78.2	82.0	44.2	71.5	69.1	77.1	78.1	73.4	67.0	36.4	63.1	69.6	75.1	68.8	67.2	26.2
(j) [2] + Ours (threshold + SVM (all context))	**74.9**	77.9	69.3	54.2	**38.3**	76.0	78.1	81.7	**44.8**	70.8	68.8	77.1	78	73.3	**67.2**	36.8	**63.8**	69.7	75.2	**69.1**	**67.3**	24.4
(k) [2] + Ours (MRF + SVM (all context))	**74.9**	77.8	69.3	54.3	**38.3**	76.0	78.1	81.4	**44.8**	70.7	69.9	77.1	78.1	73.7	67.1	**37.0**	63.4	69.7	**75.3**	69.0	**67.3**	26.2

use $p(b_w = 1|s_w, c_w)$ because the probability of correct detection of window w differs for the different object classes predicted. We use a posterior probability $p(b_w = 1|L_y^w, c_w)$ to consider the spatial position of a given window w ("abspos" in Table 1). Given candidate windows W from a single image, the spatial position likelihood of the window $w \in W$ ("relpos" in Table 1) and the scale likelihood of the window $w \in W$ ("scale" in Table 1) are defined as

$$l_{pos}(w) = \frac{\sum_{w' \in W \setminus w} p_{exist}(w') \cdot p(b_w = 1|d_{pos}(w, w'), c_w, c_{w'}, b_{w'} = 1)}{\sum_{w' \in W \setminus w} p_{exist}(w')}, \quad (10)$$

$$l_{scale}(w) = \frac{\sum_{w' \in W \setminus w} p_{exist}(w') \cdot p(b_w = 1|d_{scale}(w, w'), c_w, c_{w'}, b_{w'} = 1)}{\sum_{w' \in W \setminus w} p_{exist}(w')}. \quad (11)$$

Given an image i, we use the final layer output of Places-CNN [17] as a 205-dimensional probability vector $\boldsymbol{p}_{scene}(i)$ to represent the global context (labeled "global" in Table 1). The predicted candidate window scores are obtained using a probability estimation as in [18].

3. EXPERIMENTS

3.1. Implementation and evaluation metrics

We show the validity of our method on VOC2007 [13] and MSCOCO [14]. We use VOC2007 trainval for training the object detectors. Images in MSCOCO have much more ground truth objects per image than VOC2007 and is more challenging. We use MSCOCO train for training the object detectors. For the detector, we use Fast R-CNN [2] and Faster R-CNN [3]. In both detectors, mAP is slightly different from that reported in [2, 3] because we are unable to obtain the exact detection results; hence, we downloaded the code and trained the models again.

As main evaluation metrics, we use average precision (AP) and its mean (i.e., mAP). We also use F1 to show the effect of our pruning. F1 is a harmonic mean of precision and recall that considers the trade-off between precision and recall rates, without considering individual scores of detections. We compare our method with the state-of-the-art contextual models Tree-context [7] and HOOD [8]. The code in [7] was downloaded from the authors' homepage. The code in [8] was not available; therefore, we implemented it by ourselves. The code for our proposed approach is available at our website [1]. There are three parameters in our pruning method. The first is the threshold T in Sec. 2.3. To avoid considering the context of likely incorrect detections, T is fixed at 0.7 in all experiments. The second is ϵ in the unary term in Eq. (4), which is fixed at 0.04 for VOC2007 and at 0.10 for MSCOCO so as to avoid considering a large number of false positives. In the MSCOCO dataset, we use a linear SVM because the number of detection windows is extremely large. The third is β in Eq. (3), which is set such that it maximizes F1 without causing a decline in mAP for the training set. In all of our experiments, β is set to 1.0. There are two parameters in our rescoring using the SVM, C and γ. The optimal C and γ are determined using a grid search for each object class.

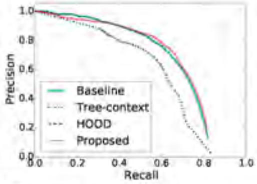


Fig. 4: VOC2007 precision-recall curves using Fast R-CNN.

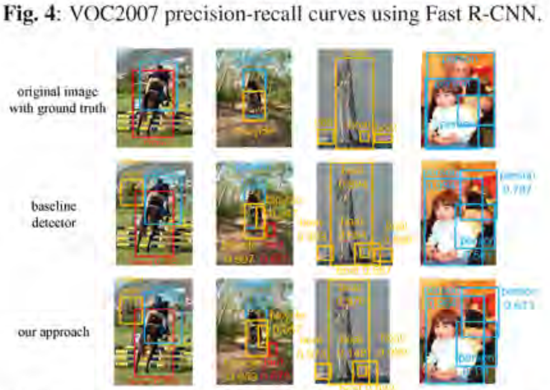


Fig. 5: Example outputs for our approach using Fast R-CNN on the VOC2007 test. We only show windows with scores after rescoring of over 0.05 so as to maintain visibility.

3.2. Results and discussions

Table 2 shows a comparison between the baseline detector Fast R-CNN [2] and our proposed method and its subsets in VOC2007. "threshold" in Table 2 implies the use of MRF-optimization by setting $\beta = 0$, which is equivalent to keeping a window w with an s_w value that is larger than ϵ. Our method, considering all contextual information [(k) in Table 2], achieved the best performance. The improvement relative to the baseline detector is 22.7% for F1 and 0.4% for mAP. The improvement relative to the baseline detector with thresholding (over 0.04) is 1.8% and 0.8% for F1 and mAP, respectively. mAP gradually increases as we introduce more contextual information. Specifically, our method shows significant improvement of AP on classes that are originally difficult to detect, such as plants (+4.0%, 33.0% → 37.0%), chairs (+3.4%, 41.4% → 44.8%), and tables (+2.2%, 67.7% → 69.9%). Table 2 also shows that the pruning candidate windows are necessary for

[1] http://www.hal.t.u-tokyo.ac.jp/~inoue

Table 3: AP [%] and F1 [%] in MSCOCO val using Fast R-CNN.

method	mAP	F1
(a) Fast R-CNN [2]	32.3	8.4
(b) [2] + Tree-context [7]	-	-
(c) [2] + HOOD [8]	21.0	**34.0**
(d) [2] + threshold	32.1	11.8
(e) [2] + Ours (SVM (all context))	32.4	8.4
(f) [2] + Ours (MRF)	32.2	11.0
(g) [2] + Ours (MRF + SVM (scene))	32.6	11.0
(h) [2] + Ours (MRF + SVM (scene,abspos))	32.8	11.0
(i) [2] + Ours (MRF + SVM (scene,abspos,relpos))	32.9	11.0
(j) [2] + Ours (threshold + SVM (all context))	32.4	8.4
(k) [2] + Ours (MRF + SVM (all context))	**33.0**	11.0

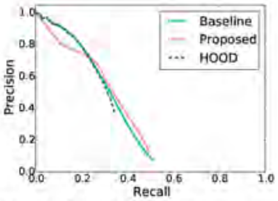


Fig. 6: MSCOCO val precision-recall curves using Fast R-CNN.

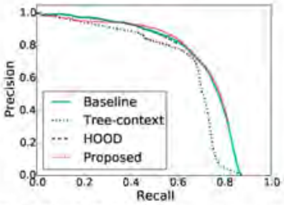


Fig. 7: VOC2007 test precision-recall curves using Faster R-CNN.

Table 4: AP [%] and F1 [%] in VOC2007 test using Faster R-CNN.

method	mAP	F1
(a) Faster R-CNN [3]	**69.7**	8.4
(b) [3] + Tree-context [7]	57.8	8.4
(c) [3] + HOOD [8]	60.2	**72.5**
(d) [3] + threshold	68.9	39.8
(e) [3] + Ours (MRF)	69.0	40.0
(f) [3] + Ours (MRF + SVM (scene))	69.0	40.0
(g) [3] + Ours (MRF + SVM (scene,abspos))	69.0	40.0
(h) [3] + Ours (MRF + SVM (scene,abspos,relpos))	69.0	40.0
(i) [3] + Ours (threshold + SVM (all context))	69.0	39.8
(j) [3] + Ours (MRF+SVM (all context))	69.1	40.0

improving F1 by comparing (d) and (f) in Table 2. By comparing (e), (j), and (k) in Table 2; we observe that candidate pruning is an essential preprocessing stage for learning-based rescoring. As described in Sec. 1, we observe that Tree-context [7] and HOOD [8] are not effective for Fast R-CNN. Although HOOD [8] improves F1, it significantly decreases main metrics, mAP from 66.9% to 57.9%. The precision-recall curve in Fig. 4 shows that our method is superior in terms of maximizing F1.

Fig. 5 shows the resultant images for our contextual model using VOC2007 and Fast R-CNN. The first and second columns of Fig. 5 show successful cases of application of our approach. In the first column, the detections of bicycle and bird are inconsistent with other windows in terms of scale relation, and thus, their scores become small. In the second column, pottedplant is often seen indoors, and thus, the detection of pottedplant is out of context in terms of the scene and its score drops sharply. The third and fourth columns of Fig. 5 show the failed cases of application of our approach. We assume that an object class has a fixed size. Although we attempt to consider this variation by approximating the distribution of the scale relation, it unable to compensate for the large variances caused by occlusion, as in the case in the third row, or scale differences, as in the case in the right-hand column.

The results on MSCOCO are shown in Table 3. For Tree-context [7], we could not obtain results because the training step of [7] failed. The improvement relative to the baseline detector is 0.7% and 2.6% for mAP and F1, respectively. The improvement relative to the baseline detector with thresholding (over 0.1) is 0.9% and −0.8% for mAP and F1, respectively. The mAP gradually increases as we use more contextual information. By comparing (e), (j), and (k) in Table 3, we observe that candidate pruning is an essential preprocessing stage for learning-based rescoring for this dataset. The precision-recall curve in Fig. 6 shows that our approach is better in terms of higher recall. The result shows that our contextual model is also effective for images that contain multiple objects.

We also used another detector, Faster R-CNN [3]. The result is shown in Table 4. The mAP obtained using our method is lower than the baseline. However, the precision-recall curve in Fig. 7 shows that the decline in mAP is due to thresholding, which results in missing correct detections whose scores are very small. Our approach still shows improvement in detections that have moderate scores, as seen when comparing (d) and (i), or (e) and (j).

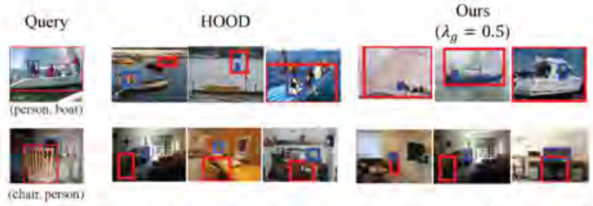


Fig. 8: Results of structured retrieval using the proposed contextual model and HOOD [8] (The three most similar images are shown).

3.3. Applications

Structured image retrieval systems, such as that in [19], aim to find images that have similar spatial and scale relationships as objects with a given query image. Our approach, which considers the context, can be easily applied to structured image retrieval when two objects are contained in the query, as in [8]. First, given a query q consisting of an image i_q and a window pair (w_q, w'_q), we retrieve the candidate images that contain the same pair of objects as the query image. Second, we define a score $dist(q,t)$ for each image t containing image i_t and window pair (w_t, w'_t) in the dataset:

$$dist(q,t)=(1-\lambda_g)\left\|\begin{pmatrix} p_{pos}(w_q,w'_q) \\ p_{scale}(w_q,w'_q) \end{pmatrix}-\begin{pmatrix} p_{pos}(w_t,w'_t) \\ p_{scale}(w_t,w'_t) \end{pmatrix}\right\|_2 +\lambda_g \left\|\boldsymbol{p}_{scene}(i_q)-\boldsymbol{p}_{scene}(i_t)\right\|_2, \quad (12)$$

where λ_g $(0 \leq \lambda_g \leq 1)$ is a parameter used to balance the likelihood of spatial and scale relationships and the global context. Smaller $dist(q,t)$ values imply that i_q and i_t are more similar. If i_t has multiple window pairs of the target object, we treat the window pair that is most similar by considering Eq. (12) as the candidate. Third, we sort the candidate images according to Eq. (12). Fig. 8 shows the example results. We show more examples at our website. Images in the VOC2007 test are used as the query; similar images are retrieved from VOC2007 trainval. Fig. 8 shows that our approach more precisely retrieves images with consistent spatial, scale, and semantic relations to the query, as compared to HOOD.

4. CONCLUSIONS

In this paper, we presented an MRF-based candidate reduction technique and learning-based object rescoring using context information. Additionally, we confirmed that previous approaches do not work well when used in state-of-the-art object detection methods such as Fast R-CNN or Faster R-CNN because their detection performance in terms of mAP is already high. Experimental results showed that our approach can improve both mAP and F1 and that all of our proposed features contribute to better object detection when using different metrics. Therefore the best performance is achieved when all of the features are combined. Applications to structured image retrieval were also presented.

5. REFERENCES

[1] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *IEEE CVPR*, 2014.

[2] Ross Girshick, "Fast R-CNN," in *IEEE ICCV*, 2015.

[3] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *NIPS*, 2015.

[4] Andrew Rabinovich, Andrea Vedaldi, Carolina Galleguillos, Eric Wiewiora, and Serge Belongie, "Objects in context," in *IEEE ICCV*, 2007.

[5] Carolina Galleguillos, Andrew Rabinovich, and Serge Belongie, "Object categorization using co-occurrence, location and appearance," in *IEEE CVPR*, 2008.

[6] Carolina Galleguillos and Serge Belongie, "Context based object categorization: A critical survey," *Comput. Vis. Image Underst.*, vol. 114, no. 6, 2010.

[7] Myung Jin Choi, Antonio Torralba, and Alan S Willsky, "A tree-based context model for object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 2, pp. 240–252, 2012.

[8] Xiaochun Cao, Xingxing Wei, Yahong Han, and Xiaowu Chen, "An object-level high-order contextual descriptor based on semantic, spatial, and scale cues," *IEEE Trans. Cybern.*, vol. 45, no. 7, pp. 1327–1339, 2015.

[9] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan, "Object detection with discriminatively trained part-based models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 9, pp. 1627–1645, 2010.

[10] Guang Chen, Yuanyuan Ding, Jing Xiao, and Tony Han, "Detection evolution with multi-order contextual co-occurrence," in *IEEE CVPR*, 2013.

[11] Davide Modolo, Alexander Vezhnevets, and Vittorio Ferrari, "Context forest for efficient object detection with large mixture models," *arXiv preprint arXiv:1503.00787*, 2015.

[12] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi, "You only look once: Unified, real-time object detection," in *IEEE CVPR*, 2016.

[13] Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman, "The pascal visual object classes (voc) challenge," *Int. J. Comput. Vis.*, vol. 88, no. 2, pp. 303–338, 2010.

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick, "Microsoft coco: Common objects in context," in *IEEE ECCV*, pp. 740–755. 2014.

[15] Jasper RR Uijlings, Koen EA van de Sande, Theo Gevers, and Arnold WM Smeulders, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, 2013.

[16] Vladimir Kolmogorov and Carsten Rother, "Minimizing non-submodular functions with graph cuts-a review," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 7, pp. 1274–1279, 2007.

[17] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva, "Learning deep features for scene recognition using places database," in *NIPS*, 2014.

[18] Chih-Chung Chang and Chih-Jen Lin, "Libsvm: a library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27, 2011.

[19] Tian Lan, Weilong Yang, Yang Wang, and Greg Mori, "Image retrieval with structured object queries using latent ranking svm," in *IEEE ECCV*. 2012.