Context Refinement for Object Detection

Zhe Chen, Shaoli Huang, and Dacheng Tao

UBTECH Sydney AI Centre, SIT, FEIT, University of Sydney, Australia
{zche4307}@uni.sydney.edu.au {shaoli.huang, dacheng.tao}@sydney.edu.au

Abstract. Current two-stage object detectors, which consists of a region proposal stage and a refinement stage, may produce unreliable results due to ill-localized proposed regions. To address this problem, we propose a context refinement algorithm that explores rich contextual information to better refine each proposed region. In particular, we first identify neighboring regions that may contain useful contexts and then perform refinement based on the extracted and unified contextual information. In practice, our method effectively improves the quality of the final detection results as well as region proposals. Empirical studies show that context refinement yields substantial and consistent improvements over different baseline detectors. Moreover, the proposed algorithm brings around 3% performance gain on PASCAL VOC benchmark and around 6% gain on MS COCO benchmark respectively.

Keywords: Object Detection · Context Analysis · Deep Convolutional Neural Network

1 Introduction

Recent top-performing object detectors, such as Faster RCNN [29] and Mask RCNN [16], are mostly based on a two-stage paradigm which first generates a sparse set of object proposals and then refines the proposals by adjusting their coordinates and predicting their categories. Despite great success, these methods tend to produce inaccurate bounding boxes and false labels after the refinement because of the poor-quality proposals generated in the first stage. As illustrated in Figure 1, if a proposed region has a partial overlap with a true object, existing methods would suffer refinement failures since this region does not contain sufficient information for holistic object perception. Although much effort such as [21] has been dedicated to enhance the quality of object proposals, it still cannot guarantee that the proposed regions can have a satisfactory overlap for each ground truth.

To tackle the aforementioned issue, we augment the representation for each proposed region by leveraging its surrounding regions. This is motivated by the fact that surrounding regions usually contain complementary information on object appearance and high-level characteristics, e.g., semantics and geometric relationships, for a proposed region. Different from related approaches [36, 12, 37, 26] that mainly include additional visual features from manually picked regions

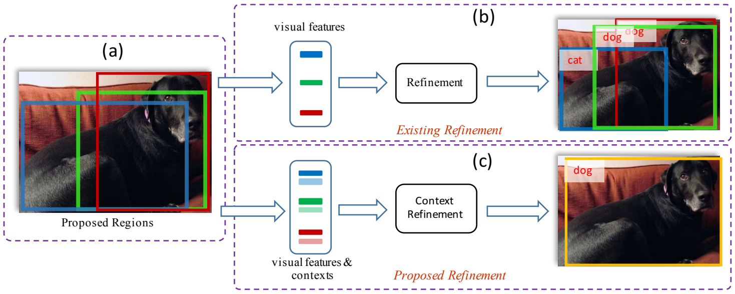


Fig. 1. Overview of the pipeline for the proposed context refinement algorithm comparing to existing refinement pipeline. Existing pipeline (b) refines each proposed region by performing classification and regression only based on visual features, while the proposed algorithm (c) can achieve a more reliable refinement by making use of both visual cues and contexts brought by surrounding regions.

to help refinement, our method is based on off-the-shelf proposals that are more natural and more reliable than hand-designed regions. Furthermore, by using a weighting strategy, our method can also take better advantage of contextual information comparing to other existing methods.

In this paper, we propose a learning-based context refinement algorithm to augment the existing refinement procedure. More specifically, our proposed method follows an iterative procedure which consists of three processing steps in each iteration. In the first processing step, we select a candidate region from the proposed regions and identify its surrounding regions. Next, we gather the contextual information from the surrounding regions and then aggregate these collected contexts into a unified contextual representation based on an adaptive weighting strategy. Lastly, we perform context refinement for the selected region based on both the visual features and the corresponding unified contextual representation. In practice, since the proposed method requires minor modification in detection pipeline, we can implement our algorithm by introducing additional small networks that can be directly embedded in existing two-stage detectors. With such simplicity of design and ease of implementation, our method can further improve the region proposal stage for two-stage detectors. Extensive experimental results show that the proposed method consistently boosts the performance for different baseline detectors, such as Faster RCNN [29], Deformable R-FCN [9], and Mask RCNN [16], with diversified backbone networks, such as VGG [32] and ResNet [17]. The proposed algorithm also achieves around 3% improvement on PASCAL VOC benchmark and around 6% improvement on MS COCO benchmark over baseline detectors.

2 Related Work

Object detection is the key task in many computer vision problems [7, 5, 18, 6]. Recently, researchers mainly adopt single-stage detectors or two-stage detectors to tackle detection problems. Compared with single-stage detectors [27, 24, 30], two-stage detectors are usually slower but with better detection performance [20]. With a refinement stage, two-stage detectors are shown to be powerful on COCO detection benchmark [23] that contains many small-sized objects and deformed objects. Over recent years, several typical algorithms [15, 29, 8, 9] have been proposed to improve the two-stage detectors. For example, [22] developed the feature pyramid network to address the challenge of small object detection. [16] proposes a novel feature warping method to improve the performance of the final refinement procedure. However, these methods are highly sensitive to the quality of object proposals and thereby may produce false labels and inaccurate bounding boxes on poor-quality object proposals.

To relieve this issue, post-processing methods have been widely used in two-stage detection systems. One of the most popular among them is the iterative bounding box refinement method [12, 37, 17]. This method repeatedly refines the proposed regions and performs a voting and suppressing procedure to obtain the final results. Meanwhile, rather than using a manually designed iterative refinement method, some studies [13, 14] recursively perform regression to the proposed regions so that they can learn to gradually adapt the ground-truth boxes. Although better performance could be achieved with more iterations of processing, these methods are commonly computational costly. In addition, some other studies adopt a re-scoring strategy. For example, the paper [2] tends to progressively decrease the detection score of overlapped bounding boxes, lowering the risk of keeping false positive results rather than more reliable ones, while Hosang *et al.* [19] re-scores detection with a learning-based algorithm. However, the re-scoring methods do not consider contexts, thus only offering limited help in improving the performance.

More related studies refer visual contexts to improve object detection. Even without the powerful deep convolutional neural networks (DCNNs), the advantages of using contexts for object detection have already been demonstrated in [10, 35, 25]. In recent years, many studies [36, 12, 37, 26] attempt to further incorporate contexts in DCNN. In general, they propose to utilize additional visual features from context windows to facilitate detection. A context window is commonly selected based on a slightly larger or smaller region comparing to the corresponding proposed regions. The visual features inside each context window will be extracted and used as contextual information for the final refinement of each region. However, since context windows are commonly selected by hand, the considered regions still have a limited range and surrounding contexts may not be fully exploited. Instead of using context windows, some studies [1, 28, 4] attempt to employ recurrent neural networks to encode contextual information. For example, the ION detector [1] attempts to collect contexts by introducing multi-directional recurrent neural network, but the resulting network becomes much more complicated and it requires careful initialization for stable training.

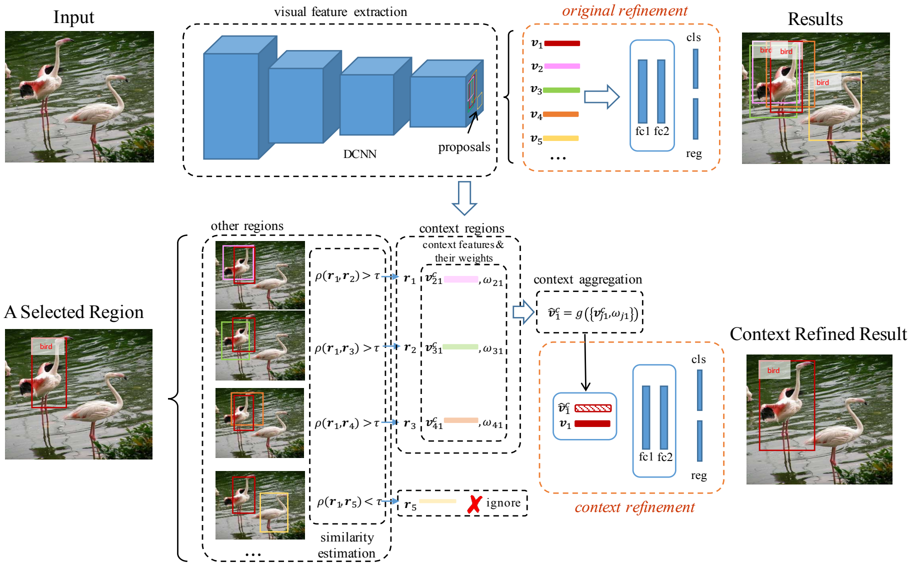


Fig. 2. The detailed working flow of the proposed context refinement algorithm for improving the original refinement (best view in color). Regarding each selected region, our algorithm first identifies its surrounding regions that may carry useful context based on a correlation estimation procedure. Afterwards, all the contextual information is aggregated to form a unified representation based on an adaptive weighting strategy. Using both the aggregated contexts and visual features extracted from DCNN, the proposed context refinement algorithm is able to improve the quality of detection results. The detailed definitions of the math symbols can be found in Section 3.

Nevertheless, most of the prevailing context-aware object detectors only consider contextual features extracted from DCNNs, lacking the consideration of higher-level surrounding contexts such as semantic information and geometric relationship.

3 Context Refinement for Object Detection

Different from existing studies that mainly extract visual contexts from manually picked regions or RNNs, we propose to extensively incorporate contextual information brought by surrounding regions to improve the original refinement.

Mathematically, we define that the status of a region $\boldsymbol{r}$ is described by its four coordinates $\boldsymbol{b} = (x_1, y_1, x_2, y_2)$ and a confidence score s. Suppose $\boldsymbol{v}_i$ represents visual features extracted from the region $\boldsymbol{r}_i$ bounded by $\boldsymbol{b}_i$, then original refinement procedure of existing two-stage detectors commonly perform the following operations to refine the region $\boldsymbol{r}_i$:

$$\begin{cases} s_i = f_{cls}(\boldsymbol{v}_i) \\ \boldsymbol{b}_i = f_{reg}(\boldsymbol{b}_i^0, \boldsymbol{v}_i) \end{cases} \tag{1}$$

where $\boldsymbol{b}_i^0$ is the original coordinates of the proposed region, f_{cls} and f_{reg} respectively represent the classification and regression operations. In a two-stage detector, f_{cls} is usually a soft-max operation and f_{reg} is generally a linear regression operation. Both operations perform refinement based on the inner product between the input vector and the weight vector. The classification operation actually assigns a pre-defined box (namely anchor box) with a foreground/background label and assigns a proposal with a category-aware label; the regression operation estimates the adjustment of the coordinates for the region. As mentioned previously, two-stage detectors which refine proposed regions based on Eq. 1 suffer from the issue that ill-localized proposed regions would result in unreliable refinement, if not considering context. Based on the observation that surrounding regions can deliver informative clues for describing the accurate status of an object, we introduce context refinement algorithm to tackle the partial detection issue and thus improve the original refinement.

The processing flow of the proposed algorithm can be described as an iterative three-stage procedure. In particular, the three processing stages for each iteration include: 1) selecting candidate region and identifying its context regions; 2) aggregating contextual features; and 3) conducting context refinement. Formally, we make $\boldsymbol{r}_i$ represent the selected region in current iteration and further define the surrounding regions of $\boldsymbol{r}_i$ that may carry useful contexts as its context region. In the first stage, we select a candidate region $\boldsymbol{r}_i$ and then the context regions of $\boldsymbol{r}_i$ can be properly obtained by collecting other regions that are in the neighbourhood and closely related to the selected region. We use the symbol R_i^c to represent the set of the obtained context regions for $\boldsymbol{r}_i$. Afterwards, in the second stage, we extract contextual features from R_i^c and fuse these contexts into a unified representation, $\hat{\boldsymbol{v}}_i^c$, based on an adaptive weighting strategy. For the last stage, based on both $\boldsymbol{v}_i$ and $\hat{\boldsymbol{v}}_i^c$, we perform context refinement using the following operations:

$$\begin{cases} s_i' = f_{cls}^c(\boldsymbol{s}_i, \boldsymbol{v}_i, \hat{\boldsymbol{v}}_i^c) \\ \boldsymbol{b}_i' = f_{reg}^c(\boldsymbol{b}_i, \boldsymbol{v}_i, \hat{\boldsymbol{v}}_i^c) \end{cases} \tag{2}$$

where $\boldsymbol{b}_i'$ and s_i' are the results of context refinement, and f_{cls}^c and f_{reg}^c are the context refinement functions for classification and regression respectively. The detailed workflow of context refinement for improving the detection is illustrated in Eq. 2.

3.1 Selecting Regions

In our proposed algorithm, the first step is to select a candidate region and identify its context regions for refinement. According to Eq. 2, we perform original refinement before the first step of our algorithm so that the regions can be first enriched with semantics and meaningful geometry information. This can also make the regions tend to cluster themselves around true objects and thus can convey helpful context.

After the original refinement, the estimated confidence score can indicate the quality of a region to some extents. In this study, we adopt a greedy strategy to select regions in each iteration, which means that regions of higher scores will be refined with contexts earlier. When a region is selected, we then identify its context regions for extracting contextual information. In our algorithm, the context regions represent closely related regions, considering that these regions could cover the same object with the selected region. In order to obtain an adequate set of context regions, we estimate the closeness between the selected region and the other regions. Therefore, the regions that are closer to the selected region can form an adequate set of context regions R_i^c.

We introduce the concept of correlation level to define the closeness between a selected region and other regions. The correlation level represents the strength of the relationship between any two regions. We use $\rho(\boldsymbol{r}_i, \boldsymbol{r}_j)$ to describe the correlation level between $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$. Using this notation, we describe the set of context regions for $\boldsymbol{r}_i$ as:

$$R_i^c = \{\boldsymbol{r}_j | \rho(\boldsymbol{r}_i, \boldsymbol{r}_j) > \tau\} \tag{3}$$

where τ is a threshold. In our implementation, we measure the correlation level between two regions based on their Intersect-over-Union (IoU) score, thus $\rho(\boldsymbol{r}_i, \boldsymbol{r}_j) = IoU(\boldsymbol{b}_i, \boldsymbol{b}_j)$. The detailed setting for τ is defined in Section 5.

3.2 Fusing Context

Context extracted from R_i^c can provide complementary information that could be beneficial for rectifying the coordinates and improving the estimated class probabilities for the selected candidate regions. However, a major issue of using the collected contextual information is that the number of context regions is not fixed and can range from zero to hundreds. Using an arbitrary amount of contextual information, it will be difficult for an algorithm to conduct appropriate refinement for $\boldsymbol{r}_i$. To tackle this issue, we introduce the aggregation function g to fuse all the collected contextual information into a unified representation based on an adaptive weighting strategy, thus facilitating the context refinement.

We use $\boldsymbol{v}_{ji}^c$ to denote the contextual information carried by $\boldsymbol{r}_j$ w.r.t $\boldsymbol{r}_i$. Then we can build a set of contextual representation V_i^c by collecting all the $\boldsymbol{v}_{ji}^c$ from R_i^c:

$$V_i^c = \{\boldsymbol{v}_{ji}^c | \boldsymbol{v}_{ji}^c \text{ for } \boldsymbol{r}_j \in R_i^c\}. \tag{4}$$

Since the size of V_i^c will vary according to different selected regions, we attempt to aggregate all the contexts in V_i^c into a unified representation. In order to properly realize the aggregation operation, we propose that the more related context regions should make major contributions to the unified contextual representation. This can further reduce the risk of distracting the refinement if surrounding regions are scattered. In particular, we adopt the use of an adaptive weighting strategy to help define the aggregation function g.

Mathematically, we refer ω_{ji} as the weight of $\boldsymbol{v}^c_{ji} \in V^c_i$ that can be adaptively computed according to different selected regions. Since we are assigning larger weights to more related context regions, we attempt to estimate the relation score between $\boldsymbol{r}_j$ and $\boldsymbol{r}_i$ and make ω_{ji} depend on the estimated score. Considering that we are using semantics (i.e. classification results) and geometry information to define regions, it is appropriate to describe ω_{ji} as a combination of semantic relation score ω^s_{ji} and geometric relation score ω^g_{ji}:

$$\omega_{ji} = \omega^s_{ji} \cdot \omega^g_{ji}. \tag{5}$$

We instantiate the semantic relation score ω^s_{ji} and geometry relation score ω^g_{ji} using the following settings:

$$\begin{cases} \omega^s_{ji} = \mathbb{1}(l_j = l_i) \cdot s_j \\ \omega^g_{ji} = IoU(\boldsymbol{b}_i, \boldsymbol{b}_j) \end{cases} \tag{6}$$

where $\mathbb{1}(\cdot)$ is a bool function and l_i, l_j represent the predicted labels for corresponding regions. Using this setting, the context regions with lower confidence and lower overlap scores w.r.t the selected region will make minor contributions to the unified contextual representation.

By denoting Ω_i as the set of estimated ω_{ji} for $\boldsymbol{v}^c_{ji} \in V^c_i$, we introduce an averaging operation to consolidate all the weighted contextual information brought by a variable number of context regions. Recall that the unified contextual representation is $\hat{\boldsymbol{v}}^c_i$, we implement the aggregation operation g based on the following equation:

$$\hat{\boldsymbol{v}}^c_i = g(\{\boldsymbol{v}^c_{ji}, \omega_{ji} | \boldsymbol{v}^c_{ji} \in V^c_i,\ \omega_{ji} \in \Omega_i\}) \tag{7}$$

where:

$$g(\{\boldsymbol{v}^c_{ji}, \omega_{ji}\}) = \frac{\sum_j \omega_{ji} \cdot \boldsymbol{v}^c_{ji}}{\sum_j \omega_{ji}}. \tag{8}$$

3.3 Learning-based Refinement

After $\hat{\boldsymbol{v}}^c_i$ is computed by Eq. 7, we are then able to perform context refinement for each selected regions based on Eq. 2. In this paper, we introduce a learning-based scheme to fulfill the context refinement. More specifically, we employ fully connected neural network layers to realize the functions f^c_{cls} and f^c_{reg}. By concatenating together the $\boldsymbol{v}_i$ and $\hat{\boldsymbol{v}}^c_i$, the employed fully connected layers will learn to estimate a context refined classification score s'_i and coordinates $\boldsymbol{b}'_i$. These fully connected layers can be trained together with original refinement network.

Overall, Algorithm 1 describes the detailed processing flow of the proposed context refinement algorithm over an original refinement procedure. The proposed algorithm is further visualized by Figure. 2.

Algorithm 1 Context Refinement

```
Require: A set of regions, $R = \{\boldsymbol{r}_i = (l_i, \boldsymbol{b}_i)\}$, that has been first refined by Eq. 1;
Ensure: A set of context refined regions $R' = \{\boldsymbol{r}'_i = (l'_i, \boldsymbol{b}'_i)\}$;
$R' \leftarrow \{\}$
for each selected originally refined region $\boldsymbol{r}_i \in R$ do
    find the set of context regions $R^c_i$ based on Eq. 3;
    collect contextual representation $V^c_i$ based on Eq. 4;
    aggregate contexts and obtain the unified contextual representation $\hat{\boldsymbol{v}}^c_i$ based on
      Eq. 5 - Eq. 8;
    perform learning-based context refinement for $\boldsymbol{r}_i$ based on Eq. 2, obtaining $l'_i$
      and $\boldsymbol{b}'_i$;
    $R' \leftarrow R' \cup (l'_i, \boldsymbol{b}'_i)$;
end for
return $R'$
```

4 Embedded Architecture

Since the proposed method only alters refinement operations, such as classification and regression, of current two-stage detectors, it is straightforward to implement the proposed method by introducing an additional network that can be directly embedded into existing two-stage object detection pipelines. Such design is lightweight and can enable us to perform context refinement for both the final detection results and the region proposals because the proposals can be considered as the refined results of pre-defined anchor boxes.

As shown in Figure 3, we can directly attach the context refinement module to both the proposal generation stage and final refinement stage compatibly. As mentioned previously, we attach networks for context refinement after the original refinement operations. It is especially necessary to perform original refinement prior to our context refinement for proposal generation stage because pre-defined anchor map does not contain semantic or geometric information that can indicate the existence of objects. Moreover, such embedding design does not revise the form of a detection result, which means that it is still possible to use post-processing algorithms.

5 Implementation Details and Discussions

To embed context refinement network in different phases of a two-stage object detector, we apply the following implementation. In the first stage that produces region proposals, we attach the network of context refinement to the top-6k proposals without performing NMS. We re-use original visual features as useful information and also include relative geometry information (i.e. coordinates offsets) and semantics to enrich the instance-level contextual information for context refinement. The resulting context feature vector then has a length of $(C + 4 + K)$ where C is the channel dimension of visual feature and K is the number of categories. The threshold for defining context regions for proposals is

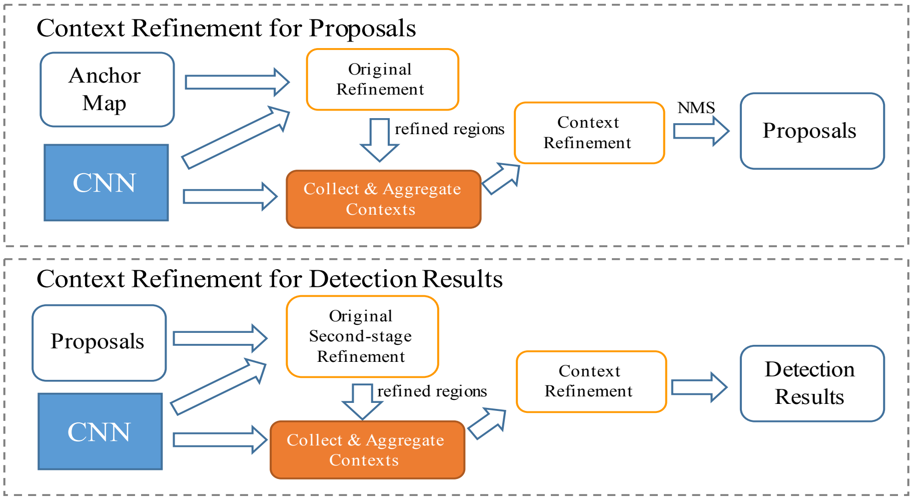

Context Refinement for Proposals
Anchor Map
CNN
Original Refinement
refined regions
Collect & Aggregate Contexts
Context Refinement
NMS
Proposals
Context Refinement for Detection Results
Proposals
CNN
Original Second-stage Refinement
refined regions
Collect & Aggregate Contexts
Context Refinement
Detection Results

Fig. 3. Embedded architecture of the proposed algorithm. This design makes the context refinement algorithm compatible for both region proposal generation stage and final refinement stage in existing two-stage detection pipeline.

set as 0.5. In addition, f^c_{cls} and f^c_{reg} are conducted on the output of two consecutive fully connected layers with ReLU non-linear activation for the first layer. In the second refinement stage, we additionally involve the semantics estimated in the first context refinement stage. The f^c_{cls} and f^c_{reg} of this stage are performed with one fully connected layer. Other settings are kept the same. When training the context refinement network, since we are using an embedded architecture, it is possible to fix the weights of other parts of a detector to achieve much higher training speed, which would not sacrifice much accuracy. The loss functions used for training are cross entropy loss for classification and smooth L1 loss for regression. Except that in the second stage, we additionally penalize the redundant detection results following the strategy proposed by [19] and thus can relieve the impacts of unnecessary detection results.

Model Complexity With the embedded design, the increase in model complexity brought by context refinement mainly comes from extracting and unifying contexts brought by context regions. Therefore, the required extra complexity would be at most $\mathcal{O}(M^2D)$ for using M candidate regions with the unified contextual feature of length D. In practice, our method will only use a small portion of proposals for context refinement. More specifically, based on Eq. 3, we can ignore a large number of proposals with a low correlation level when performing context refinement for each candidate region. In addition, we further conduct a thresholding procedure to eliminate the proposals with low confidence scores. As a result, our method only costs around 0.11s extra processing time when processing 2000 proposals.

Context Refinement for Single-stage Detectors Although it is possible to realize context refinement for single-stage detectors, we find that these detectors (e.g. SSD [24]) usually perform refinement on a smaller number of regions, meaning that there would not be sufficient surrounding contexts to access considerable improvements.

Failure Cases In general, our method brings limited improvements in two cases. The first one is that the context regions are inaccurate. In this case, the extracted contexts are not helping improve the performance. The second one is that the number of context regions is too small to provide sufficient contextual information for improvement.

6 Experiments

To evaluate the effectiveness of the proposed context refinement method for two-stage object detectors, we perform comprehensive evaluations on the well-known object detection benchmarks, including PASCAL VOC[11] and MS COCO[23]. We estimate the effects of our method on final detection results as well as region proposals, comparing to original refinement method and other state-of-the-art detection algorithms.

6.1 PASCAL VOC

PASCAL VOC benchmark [11] is a commonly used detection benchmark which contains 20 categories of objects for evaluating detectors. For all the following evaluation, we train models on both VOC 07 + 12 trainval datasets and perform the evaluation on VOC 07 test set, where mean Average Precision (mAP) will be majorly reported as detection performance.

In this section, we apply context refinement for both regional proposals and the second refinement stage in the Faster RCNN (FRCNN) detector [29]. Considering that this detector adopts region proposal network (RPN) to refine anchors, we abbreviate the context refined RPN as C-RPN. We further use C-FRCNN to represent the FRCNN whose RPN and the second refinement stage are both refined with contexts. We re-implement the FRCNN following the protocol of [3] and use † to represent this re-implemented FRCNN in following experiments.

Effects on Region Proposals We first evaluate the effectiveness of the proposed algorithm for region proposal network (RPN). The improvements in recall rates w.r.t the ground-truth objects will illustrate the efficacy of our method. In this part, recall rates will be reported based on different IoU thresholds and different number of proposals. It is worth noting that our method is not a novel region proposal algorithm, thus we do not compare with SelectiveSearch [34] and EdgeBox[38]. We only report the performance gain with respect to the original refinement performed by region proposal network.

Using different IoU thresholds as the criteria, we report the recall rates by fixing the number of proposals in each plot, as illustrated in Figure 4. From the

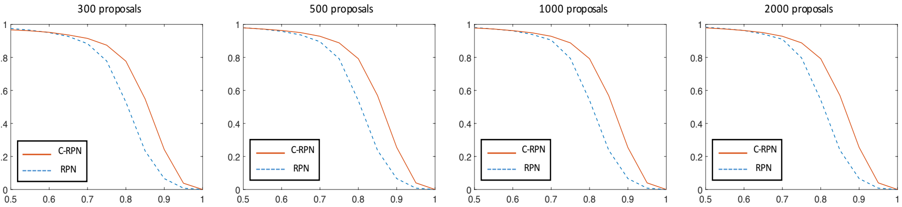


Fig. 4. Curves for the recall rates against IoU threshold on the PASCAL VOC07 test set for the original refined region proposal network and the context refined results. C-RPN refers to the region proposal network improved with contexts.

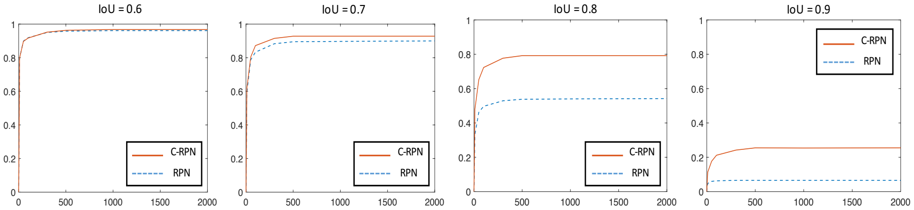


Fig. 5. Curves for the recall rates against the number of proposals on the PASCAL VOC07 test set for the original refined region proposal network and the context refined results. C-RPN refers to the region proposal network improved with contexts.

presented plots, we can find that although all the curves change slightly with the number of proposals increases, the proposed context refinement procedure can consistently boost recalls rates of original refinement. Especially, context refinement is able to improve the recall rates of original RPN with around 45% at an IoU threshold of 0.8 in each plot, which validates that the proposed algorithm is advantageous for improving the quality of region proposals.

In addition, we report the recall rates for adopting different numbers of proposals in Figure 5. In these plots, we can observe that the context refinement bring more improvements when using higher IoU thresholds as criteria. Starting from the IoU threshold of 0.8 for computing the recall rates, the improvements of the proposed method becomes obvious, out-performing the original refinement method in RPN with around 2 points for using more than 100 proposals. With a more strict IoU threshold (i.e. 0.9), the proposals refined with surrounding contexts can still capture 20% to 30% of ground-truth boxes, while original refinement only facilitates RPN to cover only around 7% ground-truth.

Effects on Detection With the help of context refinement, we not only can boost recall rates of proposals but also can promisingly promote the final detection performance. Table 1 briefly shows the ablation results of the context

Method	AP@0.5	AP@0.7	AP@0.8
FRCNN† with RPN	0.796	0.633	0.442
FRCNN† with C-RPN	0.804	0.650	0.469
C-FRCNN†	**0.822**	**0.685**	**0.485**

Table 1. Ablation study on VOC 07 test set for using context refinement to improve different refinement stages in Faster RCNN (FRCNN) detector. "C-RPN" refers to the context refinement improved region proposal network (RPN). "C-FRCNN" means the FRCNN whose both refinement stages are improved with contexts. †: the FRCNN implemented following the protocol suggested by [3].

Method	Network	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	motor	person	plant	sheep	sofa	train	tv	mAP
HyperNet [21]	VGG	84.2	78.5	73.6	55.6	53.7	78.7	79.8	87.7	49.6	74.9	52.1	86.0	81.7	83.3	81.8	48.6	73.5	59.4	79.9	65.7	71.4
ION [1]	VGG	79.2	83.1	77.6	65.6	54.9	85.4	85.1	87.0	54.4	80.6	73.8	85.3	82.2	82.2	74.4	47.1	75.8	72.7	84.2	80.4	75.6
CC [26]	BN-incep	80.9	84.8	83.0	75.9	**72.3**	**88.9**	88.4	**90.3**	66.2	87.6	74.0	89.5	89.3	83.6	79.6	55.2	83.4	81.0	**87.8**	80.7	81.1
R-FCN [8]	Res101	79.9	87.2	81.5	72.0	69.8	86.8	88.5	89.8	67.0	88.1	74.5	**89.8**	**90.6**	79.9	81.2	53.7	81.8	81.5	85.9	79.9	80.5
FRCNN† [3, 29]	VGG	76.1	82.5	75.3	65.3	65.6	84.8	87.5	87.5	57.7	82.4	67.7	83.3	85.3	77.1	78.4	44.1	76.9	70.1	82.6	77.0	75.3
C-FRCNN† (ours)	VGG	79.5	83.7	77.6	69.3	67.2	84.9	87.5	87.6	61.3	83.9	72.3	85.3	85.7	80.8	83.5	49.9	79.2	73.4	83.2	76.7	77.6
FRCNN† [3, 29]	Res101	83.1	86.0	79.7	74.2	68.3	87.7	88.0	88.4	62.3	86.8	70.4	88.5	87.3	82.9	82.9	52.8	81.0	77.7	84.5	79.3	79.6
C-FRCNN† (ours)	Res101	**84.7**	**88.2**	**83.1**	**76.2**	71.1	87.9	**88.7**	89.5	**68.7**	**88.6**	**78.2**	89.5	88.7	**84.8**	**86.2**	**55.4**	**84.7**	**82.0**	86.0	**81.7**	**82.2**

Table 2. Performance of context refinement improved Faster RCNN (C-FRCNN) detector compared to other cutting-edge detectors on VOC 07 test set. †: the Faster RCNN implemented following the protocol suggested by [3].

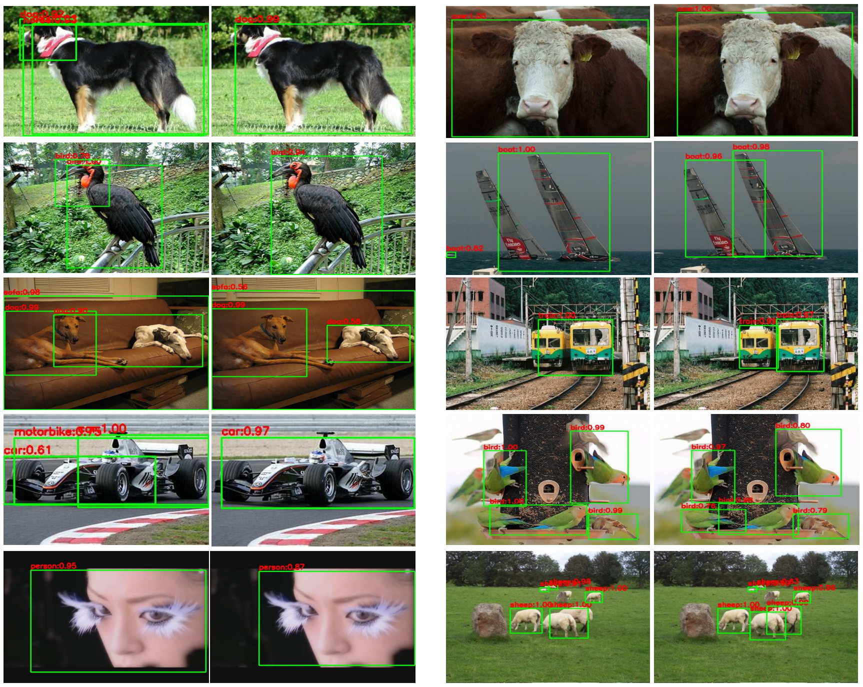
refinement algorithm for improving performance in different refinement stages. In particular, by improving the recall rates of generated proposals, context refinement brings 0.8 point's gain in final mAP using 0.5 as IoU threshold. When further employing the proposed refinement to the final refinement stage of FRCNN, there is another 1.6 points' improvement using the same metric. The presented statistics reveal that the proposed context refinement is effective in improving detection performance, especially for the final refinement stage in two-stage detectors.

Moreover, by well-considering the contextual information carried with surrounding regions, the proposed method is supposed to greatly improve the detection results comparing to original detectors no matter what backbone network is used. To verify this, we evaluate the enhancement in detection performance of adopting the use of context refinement for using different backbone networks such as VGG and ResNet in FRCNN detector, comparing to other state-of-the-art two-stage detectors. All the other compared algorithms are processed as described in original papers, using VOC 07+12 dataset as training set.

Table 2 presents the detailed results of C-FRCNN based on different backbone networks, comparing to other state-of-the-art two-stage detectors based on similar backbone networks. According to the results, context refinement respectively achieves 2.3 points higher mAP for VGG-based FRCNN detector and 2.6 points higher mAP for ResNet101-based FRCNN detector. ResNet101-based C-FRCNN helps FRCNN surpass other state-of-the-art detectors, including the context-aware algorithms such as [1] and [26].

Method	Network	AP	mAP@0.5	mAP@0.7	mAP(small)	mAP(medium)	mAP(large)
TDM[31]	Inception-ResNet-v2	36.8	57.7	39.2	16.2	39.8	52.1
GRMI [20]	Inception-ResNet-v2	34.8	55.5	36.7	13.5	38.1	52.0
FPN [22]	Res101	36.2	59.1	39.0	18.2	39.0	48.2
FRCNN† [3, 29]	Res101	37.5	58.7	40.5	18.8	41.0	51.1
DRFCN [9]	Res101	37.1	58.9	39.8	17.1	40.3	51.3
Mask RCNN* [16]	Res101	40.2	62.0	43.9	22.8	43.0	51.1
C-FRCNN† (ours)	Res101	39.0	59.7	42.8	19.4	42.4	53.0
C-DRFCN (ours)	Res101	39.1	60.9	42.5	19.0	42.4	53.2
C-Mask RCNN * (ours) [16]	Res101	**42.0**	**62.9**	**46.4**	**23.4**	**44.7**	**53.8**

Table 3. Performance of context refinement improved Faster RCNN (C-FRCNN), Deformable RFCN (C-DRFCN), and Mask RCNN (C-MaskRCNN) detectors compared to other cutting-edge detectors on MS COCO test-dev results. †: the FRCNN implemented following the protocol suggested by [3]. ∗: Mask RCNN trained with an end-to-end scheme.



original refinement context refinement original refinement context refinement

Fig. 6. Qualitative Results. Context refinement has shown to improve the coordinates as well as the labels of originally refined results. Best illustrated in color.

6.2 MS COCO

We further evaluate our approach on MS COCO benchmark. The MS COCO benchmark contains 80 objects of various sizes and is more challenging than the

PASCAL VOC benchmark. This dataset has 80k images as *train* set. We report the performance gain brought by context refinement on the *test-dev* set with 20k images. In this part, besides FRCNN detector, we also embed the context refinement module to the compelling deformable RFCN (DRFCN) detector and Mask RCNN detector and report enhancement in their detection performance. We use C-DRFCN and C-Mask RCNN to respectively represent the relating detectors refined by our algorithm.

Table 3 illustrates the detailed performance of AP in different conditions for the evaluated methods. From it, we can find that the context refinement generally brings 1.5 to 2.0 points improvement over original detectors. It shows that the performance for detecting objects of all the scales can be boosted to a better score using our algorithm, proving the effectiveness of the proposed method. Furthermore, the C-FRCNN and C-DRFCN detectors have outperformed FPN, by around 3 points. By improving the state-of-the-art detector, Mask RCNN, C-Mask RCNN detector achieves the highest AP among all the evaluated methods even compared to the models with a more powerful backbone network, i.e. InceptionResNetv2 [33]. This result also suggests that the proposed context refinement is insensitive to different two-stage detection pipelines.

6.3 Qualitative Evaluation

Figure 6 presents qualitative results of the proposed context refinement algorithm. The illustrated images show that our algorithm is effective in reducing the false positive predictions based on the contexts carried by surrounding regions. The context refined results also provide better coverage about the objects.

7 Conclusion

In this study, we investigate the effects of contextual information brought by surrounding regions to improve the refinement of a specific region. In order to properly exploit the informative surrounding context, we propose the context refinement algorithm which attempts to identify context regions, extract and fuse context based on adaptive weighting strategy, and perform refinement. We implement the proposed algorithm with an embedded architecture in both proposal generation stage and final refinement stage of the two-stage detectors. Experiments illustrate the effectiveness of the proposed method. Notably, the two-stage detectors improved by context refinement achieve compelling performance on well-known detection benchmarks against other state-of-the-art detectors.

8 Acknowledgement

This work was supported by Australian Research Council Projects FL-170100117, DP-180103424, and LP-150100671. We would like to thank Dr. Wanli Ouyang for his constructive suggestions on the response to review.

References

1. Bell, S., Lawrence Zitnick, C., Bala, K., Girshick, R.: Inside-outside net: Detecting objects in context with skip pooling and recurrent neural networks. In: CVPR. pp. 2874–2883 (2016)
2. Bodla, N., Singh, B., Chellappa, R., Davis, L.S.: Soft-nms–improving object detection with one line of code. In: ICCV (2017)
3. Chen, X., Gupta, A.: An implementation of faster rcnn with study for region sampling. arXiv preprint arXiv:1702.02138 (2017)
4. Chen, X., Gupta, A.: Spatial memory for context reasoning in object detection. ICCV (2017)
5. Chen, Z., Chen, Z.: Rbnet: A deep neural network for unified road and road boundary detection. In: ICONIP. pp. 677–687. Springer (2017)
6. Chen, Z., Hong, Z., Tao, D.: An experimental survey on correlation filter-based tracking. arXiv preprint arXiv:1509.05520 (2015)
7. Chen, Z., You, X., Zhong, B., Li, J., Tao, D.: Dynamically modulated mask sparse tracking. IEEE transactions on cybernetics **47**(11), 3706–3718 (2017)
8. Dai, J., Li, Y., He, K., Sun, J.: R-fcn: Object detection via region-based fully convolutional networks. In: NIPS. pp. 379–387 (2016)
9. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks (2017)
10. Divvala, S.K., Hoiem, D., Hays, J.H., Efros, A.A., Hebert, M.: An empirical study of context in object detection. In: CVPR. pp. 1271–1278. IEEE (2009)
11. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. IJCV **88**(2), 303–338 (2010)
12. Gidaris, S., Komodakis, N.: Object detection via a multi-region and semantic segmentation-aware cnn model. In: ICCV. pp. 1134–1142 (2015)
13. Gidaris, S., Komodakis, N.: Attend refine repeat: Active box proposal generation via in-out localization. BMVC (2016)
14. Gidaris, S., Komodakis, N.: Locnet: Improving localization accuracy for object detection. In: CVPR. pp. 789–798 (2016)
15. Girshick, R.: Fast r-cnn. In: ICCV. pp. 1440–1448 (2015)
16. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. ICCV (2017)
17. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. pp. 770–778 (2016)
18. Hong, Z., Chen, Z., Wang, C., Mei, X., Prokhorov, D., Tao, D.: Multi-store tracker (muster): A cognitive psychology inspired approach to object tracking. In: CVPR. pp. 749–758 (2015)
19. Hosang, J., Benenson, R., Schiele, B.: Learning non-maximum suppression. In: CVPR (2017)
20. Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al.: Speed/accuracy trade-offs for modern convolutional object detectors. CVPR (2017)
21. Kong, T., Yao, A., Chen, Y., Sun, F.: Hypernet: Towards accurate region proposal generation and joint object detection. In: CVPR. pp. 845–853 (2016)
22. Lin, T.Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: CVPR (2017)
23. Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: Common objects in context. In: ECCV. pp. 740–755. Springer (2014)

24. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., Berg, A.C.: Ssd: Single shot multibox detector. In: ECCV. pp. 21–37. Springer (2016)
25. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: CVPR. pp. 891–898 (2014)
26. Ouyang, W., Wang, K., Zhu, X., Wang, X.: Learning chained deep features and classifiers for cascade in object detection. ICCV (2017)
27. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. pp. 779–788 (2016)
28. Ren, J., Chen, X., Liu, J., Sun, W., Pang, J., Yan, Q., Tai, Y.W., Xu, L.: Accurate single stage detector using recurrent rolling convolution. CVPR (2017)
29. Ren, S., He, K., Girshick, R., Sun, J.: Faster r-cnn: Towards real-time object detection with region proposal networks. In: NIPS. pp. 91–99 (2015)
30. Shen, Z., Liu, Z., Li, J., Jiang, Y.G., Chen, Y., Xue, X.: Dsod: Learning deeply supervised object detectors from scratch. In: CVPR. pp. 1919–1927 (2017)
31. Shrivastava, A., Sukthankar, R., Malik, J., Gupta, A.: Beyond skip connections: Top-down modulation for object detection. arXiv preprint arXiv:1612.06851 (2016)
32. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2015)
33. Szegedy, C., Ioffe, S., Vanhoucke, V., Alemi, A.A.: Inception-v4, inception-resnet and the impact of residual connections on learning. In: AAAI. pp. 4278–4284 (2017)
34. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. IJCV **104**(2), 154–171 (2013)
35. Yu, R.R., Chen, X.S., Morariu, V.I., Davis, L.S., Redmond, W.: The role of context selection in object detection. T-PAMI **32**(9), 1627–1645 (2010)
36. Zagoruyko, S., Lerer, A., Lin, T.Y., Pinheiro, P.O., Gross, S., Chintala, S., Dollár, P.: A multipath network for object detection. arXiv preprint arXiv:1604.02135 (2016)
37. Zeng, X., Ouyang, W., Yan, J., Li, H., Xiao, T., Wang, K., Liu, Y., Zhou, Y., Yang, B., Wang, Z., et al.: Crafting gbd-net for object detection. T-PAMI (2017)
38. Zitnick, C.L., Dollár, P.: Edge boxes: Locating object proposals from edges. In: ECCV. pp. 391–405. Springer (2014)