Neurocomputing xxx (xxxx) xxx

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

C-RPNs: Promoting object detection in real world via a cascade structure of Region Proposal Networks

Dongming Yang[a,b,1], YueXian Zou[a,b,1,*], Jian Zhang[c], Ge Li[a,b,d]

[a] ADSPLAB, School of ECE, Peking University, Shenzhen, China
[b] Peng Cheng Laboratory, Shenzhen, China
[c] School of Electrical and Data Engineering, University of Technology Sydney, Australia
[d] National Engineering Laboratory for Video Technology – Shenzhen Division, China

ARTICLE INFO

Article history:
Received 2 February 2019
Revised 23 June 2019
Accepted 5 August 2019
Available online xxx

Communicated by Dr. Zhen Lei

Keywords:
Object detection
Hard samples mining
Cascade network
Region Proposal Network

ABSTRACT

Recently, significant progresses have been made in object detection on common benchmarks (i.e., Pascal VOC). However, object detection in real world is still challenging due to the serious data imbalance. Images in real world are dominated by easy samples like the wide range of background and some easily recognizable objects, for example. Although two-stage detectors like Faster R-CNN achieved big successes in object detection due to the strategy of extracting region proposals by Region Proposal Network, they show their poor adaption in real-world object detection as a result of without considering mining hard samples during extracting region proposals. To address this issue, we propose a Cascade framework of Region Proposal Networks, referred to as C-RPNs, which adopts multiple stages to mine hard samples while extracting region proposals and learn stronger classifiers. Meanwhile, a feature chain and a score chain are proposed to help learning more discriminative representations for proposals. Moreover, a loss function of cascade stages is designed to train cascade classifiers through backpropagation. Our proposed method has been evaluated on Pascal VOC and several challenging datasets like BSBDV 2017, CityPersons, etc. Our method achieves competitive results compared with the current state-of-the-arts and attains all-sided improvements in error analysis, validating its efficacy for detection in real world.

1. Introduction

Object detection is a most fundamental step in visual understanding, which aims at identifying and localizing objects of certain categories in images. To promote the development of object detection, plenty of benchmarks have been developed, i.e., PASCAL VOC [1] and MS COCO [2]. Meanwhile, several frameworks for constructing image datasets from the internet have been proposed [3–5]. Most of object detection approaches are trained and tested on these common object detection benchmarks, which typically assume that objects in images are with good visibility and balance. Obviously, this assumption is usually not satisfied in real world.

Taking littoral bird images from developed benchmarks and wild scenes as examples, the former are usually collected with better visibility, while the latter are collected via monitoring cameras with different background and camera distance. Moreover, different illumination and weather conditions may appear in wild scenes. For more intuitive observation, several examples of littoral birds are illustrated in Fig. 1. The image from BSBDV 2017 [6] shows birds from wild scenes, while images from PASCAL VOC [1] show birds from common benchmarks. The image from BSBDV 2017 is with resolution of 4912*3264, in which the heights of birds vary from 80 to 300 pixels. Images from PASCAL VOC 2007 and 2012 are with average resolution of 400*400, where the heights of birds are from 150 to 480 pixels. Apparently, the easily recognizable background in wild scenes takes more prominent position compared with that in common benchmarks. Besides, bird objects obtained from the wild scenes are with smaller sizes and less texture information. For object detection techniques, such a distribution mismatch from common benchmarks to real world have been observed to lead to a significant performance degradation.

Although enriching training data could possibly alleviate the performance degradation, it is not favored since annotating data

* Corresponding author at: ADSPLAB, School of ECE, Peking University, Shenzhen, China.
E-mail addresses: yangdongming@pku.edu.cn (D. Yang), zouyx@pku.edu.cn (Y. Zou), Jian.Zhang@uts.edu.au (J. Zhang), gli@pkusz.edu.cn (G. Li).
URL: http://web.pkusz.edu.cn/adsp/ (Y. Zou)
[1] Present address: Peking University, Xili Road, Nanshan District, Shenzhen, PR China.

https://doi.org/10.1016/j.neucom.2019.08.016
0925-2312/

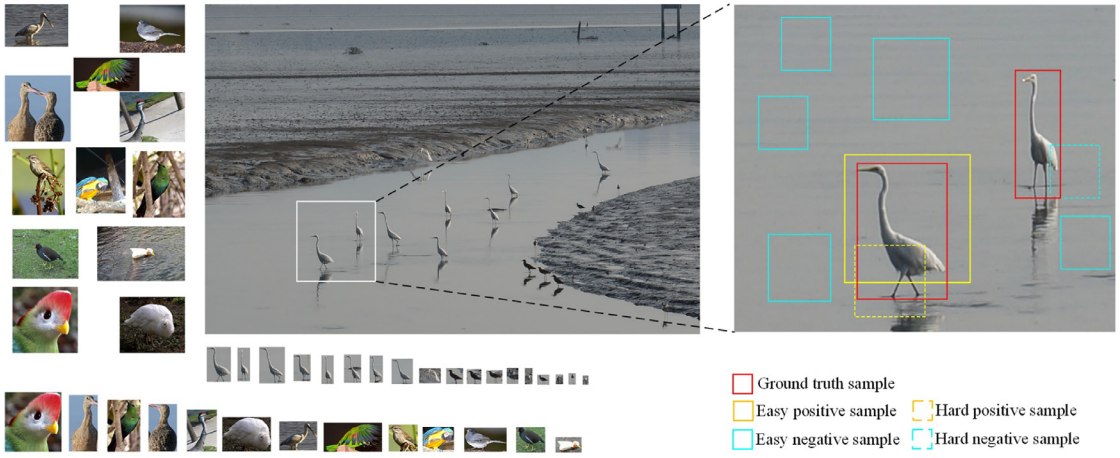

Fig. 1. Examples: (1) 12 images from Pascal VOC (left upper); (2) one littoral bird image from BSBDV 2017 (middle upper); (3) The bird objects drawn from these images (bottom) and (4) The easy and hard samples listed randomly from the real-world image (right). It is clearly that the image from realistic scenes is dominated by easy samples, especially easy negative samples. Besides, the scales and abundances of birds are mismatched from common benchmarks to realistic scenes, Best viewed in color.

is expensive and time consuming. Therefore, developing object detectors towards real world is desirable. To figure out the crucial elements of performance degradation in real-world object detection, plenty of experiments have been conducted. We list the conclusions as follows:

(1) Data imbalance frequently occurs in real world. From an image in real world, the number of negative samples (also called background samples) is much larger than that of positive samples (As shown in Fig. 1), and most of them are easy samples. Easy samples do not contribute useful learning information during training while hard samples benefit the convergence and the detection accuracy. Thus, the overwhelming number of easy samples during training leads to moronic classifiers and degenerate models.
(2) As mentioned above, because of the smaller size, poor shooting conditions and poor abundance of objects in real-world scenes, classifiers in detection algorithms are unable to learn discriminative features from ground truth.

In this work, we aim to improve the precision of object detection in real world. From the above observations, mining hard samples from abundant easy samples for training is a crucial route to address this issue. Based on the brilliant object detector Faster R-CNN [7], we firstly propose a Cascade framework of Region Proposal Networks, referred to as C-RPNs. While extracting region proposals, C-RPNs are adopted to mine hard samples and learn stronger classifiers. Multi-stage classifiers at early stages discard most of easy samples so that classifiers at latter stages focus on handling hard samples. Also, we design a feature chain and a score chain to generate more discriminative representations for proposals. Finally, a loss function of cascade stages is built to jointly learn cascade classifiers.

The contributions of this work are summarized as follows:

- Addressing data imbalance in real-world object detection, a Cascade structure of Region Proposal Networks is firstly proposed, referred to as C-RPNs.
- A feature chain and a score chain are designed in C-RPNs to further help mining hard samples and improve the classification capacity of multi-stage classifiers.
- A multi-stage loss function is constructed to jointly learn cascade classifiers.
- Integrating the above proposed components into the Faster R-CNN, our resulting model is trained end-to-end.

Extensive experiments have been conducted on several datasets, including PASCAL VOC [1], BSBDV 2017 [6], Caltech Pedestrian Benchmark [8] and CityPersons [9]. Our approach have provided competitive performance compared with the current state-of-the-arts. Besides, error analyses have shown that our approach achieved all-sided improvements compared with the baseline Faster R-CNN. The experimental results demonstrate the effectiveness of our proposed approach for object detection in real world.

2. Related work

2.1. Related work on object detection

We all have witnessed tremendous progresses in object detection using convolutional neural networks (CNNs) in recent years [7,10–15]. Region-based CNN approaches [7,10,11] are referred as two-stage detectors, which have received great attention due to their effectiveness. At the outset, R-CNN [10] was constrained by a selected search region. To reduce the computational complexity of R-CNN, Fast R-CNN [11] shared the convolutional feature maps among region of interest (RoI) and accelerated spatial pyramid pooling using RoI pooling layer. Renetal [7] introduced Region Proposal Network (RPN) to generate high-quality region proposals and then merged them with Fast R-CNN into a single network, referred to as Faster R-CNN. Besides, for faster detection, one-stage detectors such as YOLO [15] and SSD [14] were proposed to accomplish detection without region proposals, although this strategy reduced the detection performance. Researches showed that Faster R-CNN achieved a big success in object detection and laid the foundation for many follow-up works [13,16–18]. For example, feature pyramid and fusion operations were adopted [16] to enhance precision of detection. Deeper [19–21] or wider [22,23] networks also benefited the performance. Deformable CNN [18] and Receptive Field Block Net [24] enhanced the convolutional features using deformable convolutional operation and Receptive Field Block, respectively. In addition, using large batch size [25] during training provided improvement in detection. SIN [26] jointly used scene context and object relationships to promote detection performance.

Although reasonable detection performances have been achieved on benchmarks like PASCAL VOC [1] and MS COCO [2], object detection in real world still suffers from poor precision. Works mentioned above mostly focused on the conventional

Please cite this article as: D. Yang, Y. Zou and J. Zhang et al., C-RPNs: Promoting object detection in real world via a cascade structure of Region Proposal Networks, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.08.016

setting while rarely considered the adaptation issues for object detection in real world such as data imbalance.

2.2. Related work on hard example mining and cascade CNN

Gradually updating the set of background samples by selecting those from samples which are detected as false positives, bootstrapping [27] was the earliest solution to automatic employ hard samples for training. The strategy in bootstrapping led to an iterative process that alternates between updating the trained model and finding new false positives to add to the bootstrapped training set. Bootstrapping techniques were then successfully applied on detectors driven by CNN and SVMs for object detection [10,28], generally referred to as hard negative mining. After that, CNN detectors like Fast R-CNN [11] and its descendants were trained with stochastic gradient descent (SGD) on millions of samples, in which bootstrapping as an offline progress was no longer been adopted. To balance positive and negative training samples but without thinking of mining hard ones, Faster R-CNN [7] randomly used 256 samples in an image to compute the loss function of a mini-batch, where the positive and negative ones have a ratio of up to 1:1. A number of methods [29–31] then focused on mining hard samples online for training convolutional networks. Edgar [29] selected hard positive and negative samples from a larger set of random samples based on their loss independently. Ilya Loshchilov [30] and Shrivastava [31] focused on online hard sample selection strategies for mini-batch SGD methods and then OHEM [31] were introduced for region-based detectors which built mini-batches with the highest-loss samples. Recently, Focal Loss [32] has been proposed to address the extreme foreground-background class imbalance problem in object detection with one-stage detectors, which applied a modulating term to the cross entropy loss in order to focus on learning hard negative examples. Analyzing of previous works shows that inchoate bootstrapping techniques are inappropriate for CNN-based detectors. Some online hard example mining strategies selected hard examples based on their loss, which are innovative but time-consuming. Focal Loss focused on dealing with data imbalance with one-stage detectors, while our work pays more attention to two-stage detectors with region proposals.

From another perspective, cascade structure is a widely used technique to discard easy samples at early stages for learning better classification models. Before the prosperity of CNNs, cascade structure were applied to SVM [33] and boosted classifiers [34,35] with hand-crafted features. Multi-stage classifiers have been proved to be effective in generic object detection [33] and face detection [35,36], although these multiple classifiers were not trained jointly. It was showed that CNNs with cascade structure performed effectively on classification tasks [37–39] as well, in which multiple but separate CNNs were trained. After that, Qin et al. [40] proposed a method to jointly train a cascade CNNs. The recent method Cascade R-CNN [41] trained Faster R-CNN with cascade increasing IoU thresholds, which was innovative, but without considering the data imbalance issue. The newest cascade architecture Hybrid Task Cascade [42] for instance segmentation interweaved box and mask branches for a joint multi-stage processing. Based on the above observations, cascade structures are potential, but existing works either cannot be aggregated in the R-CNN based detection framework or have not considered building cascade structure on RPN to help extracting hard region proposals. Thus, confronting with the data imbalance problem in real-world object detection, the existing outcomes are very limited.

In this work, we propose C-RPNs to mine hard samples while extracting region proposals and learn more discriminative features for object detection in real world. Integrating with Faster R-CNN model, to the best of our knowledge, C-RPNs is the first cascade model of Region Proposal Networks for object detection.

3. Proposed method

3.1. Overview of C-RPNs

Faster R-CNN consists of a shared backbone convolutional network, a region proposal network (RPN) and a final classifier based on region-of-interest (RoI), in which the RPN is employed to extract region proposals. Without considering mining hard samples in the process of RPN, Faster R-CNN shows its limited capacity in detection in realistic scenes. Our novel C-RPNs are firstly proposed to address this problem.

For performance comparison fairness, VGG16 is taken as the backbone network [19]. Fig. 2 shows an overview of our proposed C-RPNs model. At first, several shared bottom convolutional layers are used for extracting convolutional features from the image (Conv1-Conv4_1). Then, C-RPNs is adopted upon four different convolutional layers, which are Conv4_2, Con4_3, Conv5_2 and Conv5_3. Since feature maps from Conv5 have the same channels but half size compared with those from Conv4, we employ an average pooling with size of 2*2 upon Conv4_2 and Conv4_3 to obtain feature maps of same resolutions for these four stages. In our C-RPNs, configuration of anchors is in keeping with Faster R-CNN, where 3 scales and 3 aspect ratios are adopted, and 9 anchors at each sliding position are generated.

At stage *1*, the feature map extracted from Conv4_2 are used for generating region proposals and obtaining binary classification scores by a softmax function. The binary classification scores estimate probabilities of a sample belonging to background and objects. With the classification scores and a reject threshold *r*, part of easy samples will be rejected at this stage, which are detailed in Section 3.3. At stage *2*, if a proposal has not been rejected at the former stage, then the feature map from Conv4_3 for this proposal is used for further binary classification. Similar processes are applied at stage *3* and stage *4*. Since there is no constrain that the rejected samples must be background, few easy positive samples might also be rejected at early stages during training. It is worth to point out that the stage *4* achieves not only binary classification but also bounding box regression. After these four stages, the proposals have not been rejected are sent to RoI pooling layer for final detection. In this study, we set batch of each stage as 1024, 768, 512 and 256, respectively, so that the stage *4* has the same batch size with RPN from Faster R-CNN. It is worth mentioning that the reason why we set only *4* stages not *5* or more is that employing the shallow and bigger feature maps from Conv3 contributes very limited performance gain but is time-consuming according to our experiments.

From Fig. 2, it can be seen that C-RPNs takes different convolutional features stage-by-stage which enables it to obtain different semantic information and receptive field. It is also noted that, in C-RPNs, the classifiers at shallow stages discard easier samples so that the classifiers at deeper stages focus on handling more difficult samples. The easy samples rejected by a classifier from shallow stage will not participate in the latter stages. With this design, abundant samples can be used but only hard samples been mined will go for final classification and bounding box regression, which benefits to alleviate the data imbalance problem.

To further enhance the classification capacity, a feature chain and a score chain are designed in C-RPNs, which are detailed in Section 3.2. In the end, the multi-stage classifications and bounding box regressions are learned in an end-to-end manner through backpropagation via a joint loss function, details are given in Section 3.3.

Please cite this article as: D. Yang, Y. Zou and J. Zhang et al., C-RPNs: Promoting object detection in real world via a cascade structure of Region Proposal Networks, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.08.016

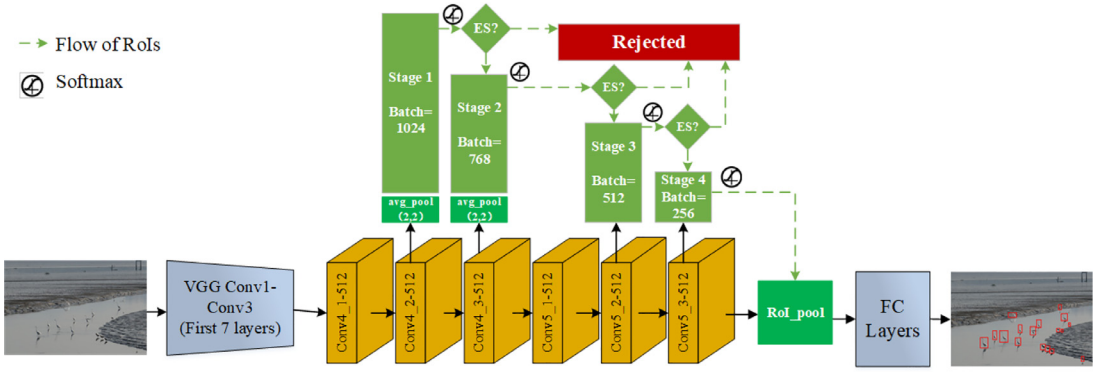

Fig. 2. An overall of our proposed C-RPNs model. After several shared bottom convolutional layers (Conv1-Conv4_1), C-RPNs are adopted upon four different convolutional layers (Conv4_2, Con4_3, Conv5_2 and Conv5_3). C-RPNs takes different convolutional features stage-by-stage to discard easy samples and learn stronger classifiers. ES refers to easy samples..

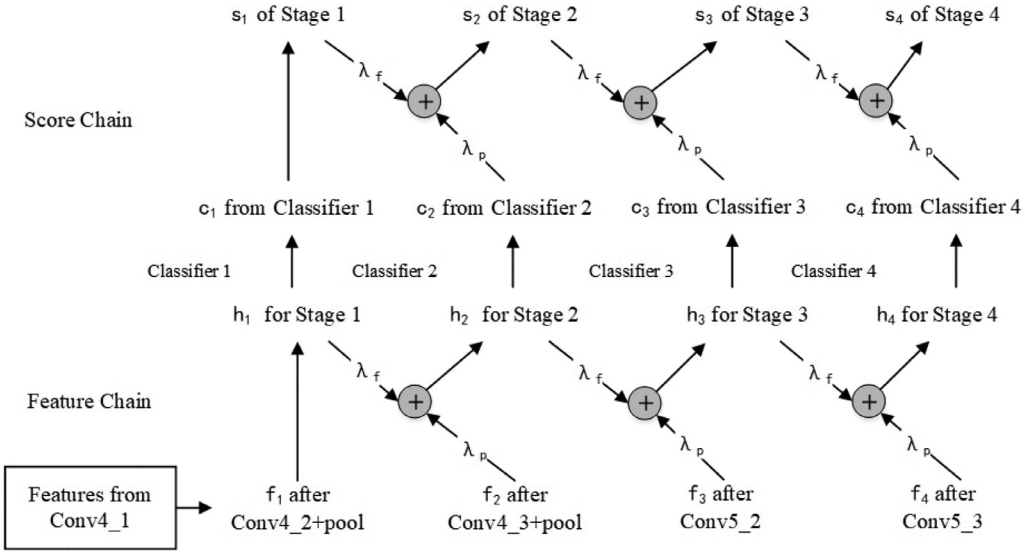

Fig. 3. The proposed feature chain and score chain in C-RPNs. Features and scores at current stage employ those from previous stages to optimize the classifier at current stage. The λ_f and λ_p are hyper parameters controlling both the feature fusion and score fusion.

3.2. Feature chain and score chain

Literature studies show that FPN [16] and DSSD [43] are effective for object detection using multiple convolutional layers. In this study, in order to capture the variation of features from different layers, a feature chain and a score chain at cascade stages are designed which are able to make use of features at previous stages as the prior knowledge for the classification at current stage. Differing from the top-down pathway with lateral connections in FPN, our feature fusion operation follows the bottom-up pathway which is the feed-forward computation of the VGG16. Moreover, FPN uses multi-scale features concurrently to predict objects, while our C-RPNs employs multi-scale features step by step to optimize the classifiers. The description of feature chain and score chain is shown in Fig. 3.

We define the number of stages as T and t is the stage index. At stage t, we denote the features from convolutional layer as f_t while features for classification as h_t. The feature chain is formulated as following:

$$h_t = \begin{cases} f_t & when \quad t = 1 \\ \lambda_f * h_{t-1} \bigoplus \lambda_p * f_t & when \quad t > 1 \end{cases} \tag{1}$$

where $\oplus$ denotes the summarized point to point. $\lambda = \{\lambda_f, \lambda_p\}$ are hyper parameters controlling the weight of features from former stage and present convolutional layer to generate fusional features for classification. λ_f and λ_p add up to 1. Considering features from present convolutional layer are more helpful for classification, we set λ_f as 0.1 and λ_p as 0.9 according to our empirical tests (detailed in Section 4.5). The fused features h_t are then used for classification.

At stage t, for each proposal have not been rejected at the t-1 stage, we denote the score from classifier t as c_t while the output score of this stage as s_t. The designed score chain has the following formulation.

$$s_t = \begin{cases} c_t & when \quad t = 1 \\ \lambda_f * s_{t-1} + \lambda_p * c_t & when \quad t > 1 \end{cases} \tag{2}$$

In this implementation, features and scores at current stage make use of those from previous stages to enhance the capacity of the classifier at current stage.

3.3. Cascade loss function with samples mining

In Faster R-CNN, training loss is composed of loss from RPN and ROIs. The former contains a binary classification loss and a regression loss. In our method, illustrated in Fig. 4, C-RPNs contains four binary classification losses and a regression loss.

Please cite this article as: D. Yang, Y. Zou and J. Zhang et al., C-RPNs: Promoting object detection in real world via a cascade structure of Region Proposal Networks, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.08.016

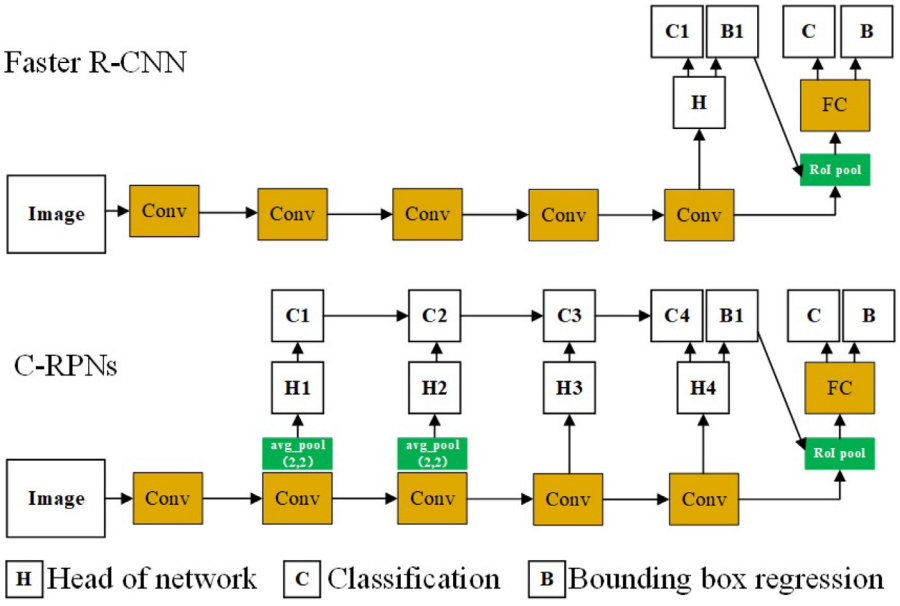

Fig. 4. Cascade Losses of our proposed C-RPNs. Faster R-CNN [7] is displayed as baseline to show our characteristics.

In C-RPNs, the cascade classifiers assign probabilities of a sample to background and objects. k={0,1} is denoted to express these two class, respectively. At stages $t\epsilon${1, 2, 3, 4}, the set of class scores for a sample are denoted by $s = \{s_t | t = 1, \ldots, T\}$. $s_t = \{s_{(t,0)}, s_{(t,1)}\}$ are scores at stage t for background and objects respectively. Another layer at stage 4 outputs bounding box regression offsets $l = \{l^k | k = 1\}$, $l^k = (l^k_x, l^k_y, l^k_w, l^k_h)$ for objects. Our proposed loss function of C-RPNs has the following formulation:

$$L_{C-RPNs}(s, k^*, l, l^*) = L_{cls}(s, k^*) + L_{loc}(l, l^*, k^*) \tag{3}$$

$$L_{cls}(s, k^*) = -\sum_{t=1}^{T} \alpha_t \mu_t log(s_{t,k^*}) \tag{4}$$

where $L_{cls}(^*)$ is the loss for classification and L_{loc} is the loss for bounding box regression. For L_{loc}, we use the *smoothed* L_1 loss [11]. For $L_{cls}(^*)$, α_t and μ_t are defined as follows, where α_t is a parameter that controls the weight of loss from cascade classifiers and μ_t evaluates whether the sample is rejected at previous stages.

$$\alpha_t = \frac{\alpha_T}{10^{T-t}} \tag{5}$$

$$\mu_t = \begin{cases} \prod_{i=1}^{t-1} [s_{t,k^*} < r] & when \quad t > 1 \\ 1 & when \quad t = 1 \end{cases} \tag{6}$$

Here, we set $\alpha_T = 1$, where T=4 in C-RPNs. Since scores from deeper classifiers are more crucial for final classification than those from shallow classifiers, α_t from deeper classifiers has been distributed more weight with a tenfold increase based on our experience. For μ_t, we set the r as a threshold value at each stage. $[s_{(t,k^*)} < r]$ will output *1* if it is true or output *0* if it is false. If a sample has been rejected at previous stages, it will no longer be used for training the classifier at current stage. We set r as *0.99* according to our empirical tests (detailed in Section 4.5). If $\alpha_t = \mu_t = 1$ and $T = 1$, then $L_{cls}(^*)$ is a normal cross entropy loss.

For the object detection with the proposed model, the final training loss is designed to compose the loss from C-RPNs and the loss from ROIs:

$$L_{detection} = L_{C-RPNs} + L_{roi} \tag{7}$$

where L_{C-RPNs} and L_{roi} are both composed of classification loss and regression loss. The former contains four cascade binary classification losses while the latter contains a multi-class classification loss. With this loss function, multiple classifiers and bounding box regressions are learned jointly through backpropagation.

Integrating the above proposed components all together, our resulting model is qualified to confront the data imbalance problem in real-world object detection. We notice that there are several cascade-based methods constructed for different tasks very recently, e.g., Hybrid Task Cascade [42] and Siamese Cascaded Region Proposal Networks [44]. The Siamese Cascaded Region Proposal Networks [44] for visual tracking task, for example, also builds upon a multi-stage framework and stimulates hard negative sampling. Nevertheless, compared with our C-RPNs, the Siamese Cascaded Region Proposal Networks is constructed with a different structure and for the specific visual tracking task. To make the algorithm more efficient, we use multi-scale features following the bottom-up pathway to discard easy samples, while Siamese Cascaded Region Proposal Networks employs multi-scale features in a top-down way to capture fine-grained semantics for visual tracking. Besides, our feature chain and score chain are easier in terms of fusion operations, leading to lower computation burden in object detection.

4. Experiments and evaluations

4.1. Experimental setup

Datasets and evaluation metrics: We evaluated our approach on several public object detection datasets, including PASCAL VOC [1], BSBDV 2017 [6], Caltech Pedestrian Benchmark [8] and CityPersons [9]. For evaluation, we used the standard average precision (AP) and mean average precision (mAP) scores with IoU thresholds at *0.5*.

Pascal VOC: Pascal VOC involves 20 categories. VOC 2007 dataset consists of about 5k trainval images and 5k test images, while VOC 2012 dataset includes about 11k trainval images and 11k test images. Following the protocol in [11], we perform training on the union of VOC 2007 trainval and VOC 2012 trainval. The test is conducted on VOC 2007 test set.

Please cite this article as: D. Yang, Y. Zou and J. Zhang et al., C-RPNs: Promoting object detection in real world via a cascade structure of Region Proposal Networks, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.08.016

Table 1
Results on PASCAL VOC 2007 test set. 07+12: union of Pascal VOC07 trainval and VOC12 trainval.

Method	Trainset	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast R-CNN [11]	07+12	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	74.8	80.4	70.4
Faster R-CNN [7]	07+12	73.2	76.5	79.0	70.9	65.5	52.1	83.1	84.7	86.4	52.0	81.9	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
SSD500 [14]	07+12	75.1	79.8	79.5	74.5	63.4	51.9	84.9	85.6	87.2	56.6	80.1	70.0	85.4	84.9	80.9	78.2	49.0	**78.4**	72.4	84.6	75.5
ION [22]	07+12	75.6	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**
RON [51]	07+12	**77.6**	**86.0**	82.5	76.9	**69.1**	59.2	**86.2**	85.5	87.2	59.9	81.4	73.3	85.9	**86.8**	**82.2**	**79.6**	**52.4**	78.2	**76.0**	**86.2**	78.0
SIN [26]	07+12	76.0	77.5	80.1	75.0	67.1	62.2	83.2	86.9	**88.6**	57.7	**84.5**	70.5	**86.6**	85.6	77.7	78.3	46.6	77.6	74.7	82.3	77.1
C-RPNs (ours)	07+12	76.4	78.6	79.5	76.3	66.5	**63.2**	84.6	**87.8**	87.8	**60.2**	83.3	71.7	85.5	86.1	81.4	79.2	49.2	75.2	73.9	83.1	75.7

BSBDV2017: The Birds Dataset of Shenzhen Bay in Distant View [6] is a great challenging dataset in wild scenes, consisting of 1421 trainval images and 351 test images. BSBDV2017 contains three kinds of image resolutions, which are 2736*1824, 4288*2848 and 5472*3648 respectively. Size of birds varies greatly from 18*30 to 1274*632.

Caltech Pedestrian Benchmark: The Caltech Pedestrian Benchmark [8] includes a total of 350,000 bounding boxes of pedestrians. Approximately 2300 unique pedestrians were annotated in roughly 250,000 frames. Following the protocol in [45], one frame from every five frames of Caltech Benchmark and all frames of the ETH [46] and TUD-Brussels [47] are extracted as training data, which includes 27,021 images in total. 4024 images in the standard test set are used for evaluation.

CityPersons: The CityPersons dataset [9] consists of images recorded across 27 cities, 3 seasons, various weather conditions and more common crowds. It creates high quality bounding box annotations for pedestrians in 5000 images, which is a subset of the Cityscapes dataset [48]. 2975 images from train set and 500 images from val set are used for training and testing respectively.

Implementation details: Faster R-CNN is taken as our baseline, where all parameters are set according to the original publication [7] if not specified. We initialize the backbone network using a VGG16 pre-trained model on ImageNet [49] while all new layers are initialized by drawing weights from a zero-mean Gaussian distribution with standard deviation 0.01. For training on Pascal VOC, we use a learning rate of 0.001 for 80k iterations and 0.0001 for 30k iterations. For training on the other datasets, we use a learning rate of 0.001 for 50k iterations and 0.0001 for 20k iterations. We trained our model in the end-to-end manner with Stochastic Gradient Descent (SGD), where the momentum is 0.9, and the weight decay is 0.0005. Our program is implemented by Tensorflow [50] on a GPU of GeForce GTX TITAN X.

4.2. Overall performance

Performance on Pascal VOC benchmark: We compare our approach with several state-of-the-arts in this subsection. Results in terms of mean average precision (mAP) are shown in Table 1. Our method achieves the second best performance among all methods, which is 1.2% lower than that of RON [51] but 3.2% higher than that of baseline Faster R-CNN. Besides, it is happy to see that our method outperforms ION [22] with the same backbone network which used features from Conv3_3, Conv4_3 and Conv5_3 to leverage context and multi-scale knowledge for object detection. From the table, we can see that although C-RPNs is designed aiming to improve detection in real world with unbalanced data, it gets competitive performance on Pascal VOC benchmark.

Performance on BSBDV 2017: Table 2 shows the comparisons of C-RPNs with state-of-the-arts on BSBDV 2017. As shown in Table 2, our method achieves the best performance and its average precision (AP) is 3.4% higher than the second best (FPN [16]). More specifically, the AP of C-RPNs is 70.3%, which obtains 11% performance gain compared with that of Faster R-CNN. It is noted that our C-RPNs gets slightly lower mAP than that of RON [51] on VOC 2007, but it outperforms RON by a margin of 12.3% on BSBDV 2017. Also, the AP of C-RPNs is 8.8% and 3.4% higher than that of R-FCN [12] and FPN [16], respectively. These results demonstrate that our C-RPNs is more competitive in real-world object detection.

Table 2
Performance comparison on BSBDV 2017.

Method	Backbone network	AP (%)
SSD500 [14]	VGG16 reduce	42.0
Faster R-CNN [7]	VGG16	59.3
RON [51]	ResNet-101	58.0
R-FCN [12]	ResNet-50	61.5
FPN [16]	ResNet-50	66.9
SIN [26]	VGG16	58.4
C-RPNs (ours)	VGG16	**70.3**

Table 3
Performance comparison on Caltech Pedestrian Benchmark and CityPersons.

Method	Caltech Pedestrian Benchmark	CityPersons
Faster R-CNN [7]	44.0	49.1
C-RPNs (ours)	**48.1**	**51.4**

Comparison with baseline Faster R-CNN on pedestrian datasets: Pedestrian datasets like Caltech Pedestrian Benchmark [8] and CityPersons [9] are more challenging then Pascal VOC, which are collected via monitoring cameras on realistic street scenes. Performances on these two datasets are helpful to verify the efficiency of our approach since the scales and occlusion of pedestrians are changed frequently. Table 3 shows the comparisons of our C-RPNs with the baseline Faster R-CNN on these pedestrian datasets. Our C-RPNs achieves average precision of 48.1% and 51.4% on Caltech Pedestrian Benchmark and CityPersons, bringing 4.1% and 2.3% performance gain upon baseline Faster R-CNN, respectively, which indicates its robustness in intricate realistic scenes.

4.3. Quantitive examples

Qualitative examples on wild bird detection: For visualization purpose, several examples of detection results on BSBDV 2017 are given in Fig. 5. The rows from the top to the bottom are respectively expressed as the results of Faster R-CNN and C-RPNs. Detection boxes from detectors are marked red. For better observation, we marked boxes of miss detection cases in yellow. According to the ground truth, there are 46 and 22 birds in the top and bottom images, respectively. From the results, we can see that our method shows significantly improved recall for object detection in wild scenes, where 40 and 17 birds have been detected, respectively. Compared with the results detected with Faster R-CNN, our method brings 16 and 2 more birds detected in two images respectively. Meanwhile, dotted boxes in blue show samples are detected with more than one boxes, three in the top images and none in the bottom images, which indicates that our method is able to generate more precise bounding boxes.

Please cite this article as: D. Yang, Y. Zou and J. Zhang et al., C-RPNs: Promoting object detection in real world via a cascade structure of Region Proposal Networks, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.08.016

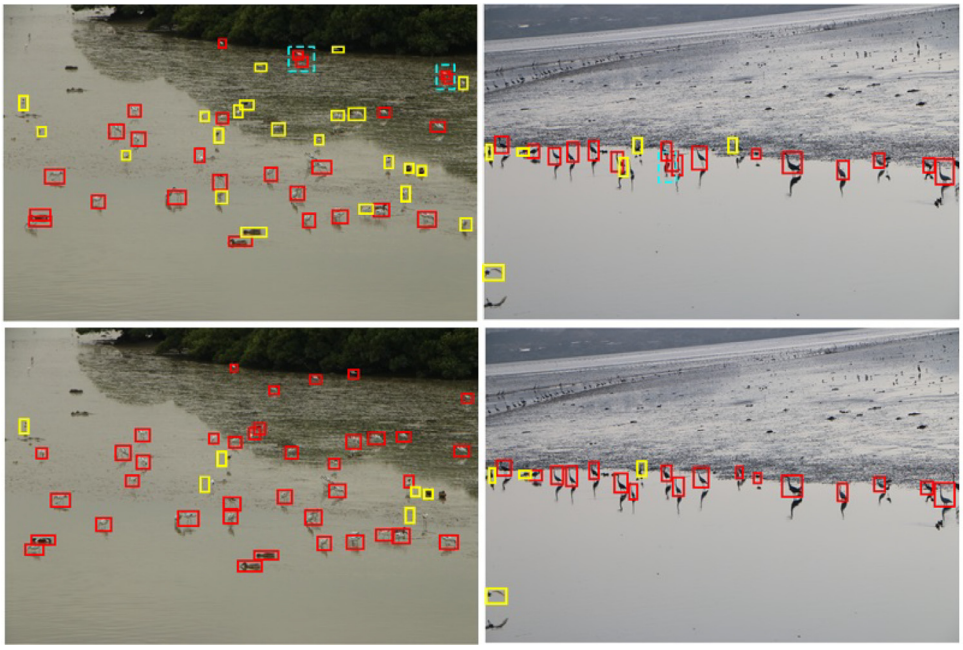

Fig. 5. Detection results of Faster R-CNN (row 1) and our proposed C-RPNs (row 2) on BSBDV 2017.

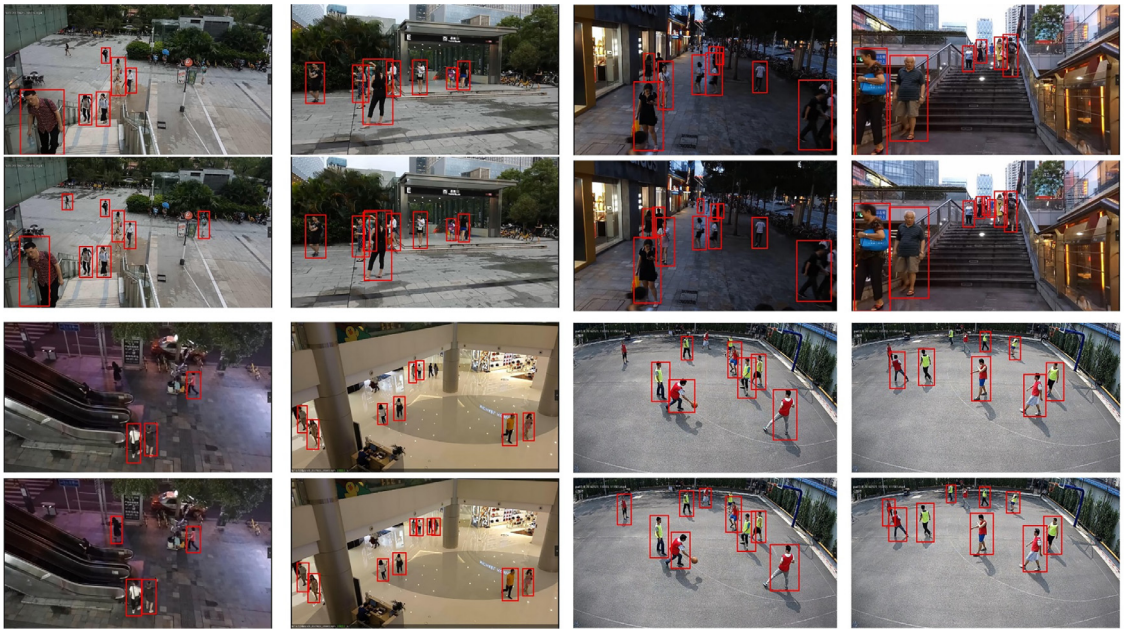

Fig. 6. Detection results of Faster R-CNN (row 1 and row 3) and our proposed C-RPNs (row 2 and row 4) on realistic pedestrian images.

Qualitative examples on practical pedestrian detection: In Fig. 6, our proposed approach is trained on Caltech Pedestrian Benchmark and tested in realistic environments with random pedestrian flows. We show some detection images with different shooting angles such as looking down and looking up or with poor illumination, which are collected in subway, park and campus. Compared with the results from Faster R-CNN, our method brings more true positive and less false positive detections in these images respectively. It is found that some hard samples are falsely detected as background from Faster R-CNN, while those are detected aright as pedestrians from C-RPNs. According to the results in Fig. 6, our proposed C-RPNs can adapt to harsh and complex environments to provide high quality object detection in real world.

4.4. Improvement analysis on false detections

To further examine the improvement of our C-RPNs upon baseline Faster R-CNN, the analysis tools [52] upon Pascal VOC are employed to produce a detailed error analysis. In Pascal VOC, animal categories include bird, cat, cow, dog, horse, sheep and person. Plane, bicycle, boat, bus, car, motorbike and train make up the vehicle categories. Fig. 7 takes animals and vehicles as examples to show the frequency and impact on the performance of each type of false positive. As shown, C-RPNs reduces detection errors compared with Faster R-CNN when detecting both animals and vehicles. It is found that C-RPNs has less BG errors as well as Loc errors compared with the baseline, indicating that C-RPNs can classify and localize objects better because it mined hard samples during

Please cite this article as: D. Yang, Y. Zou and J. Zhang et al., C-RPNs: Promoting object detection in real world via a cascade structure of Region Proposal Networks, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.08.016

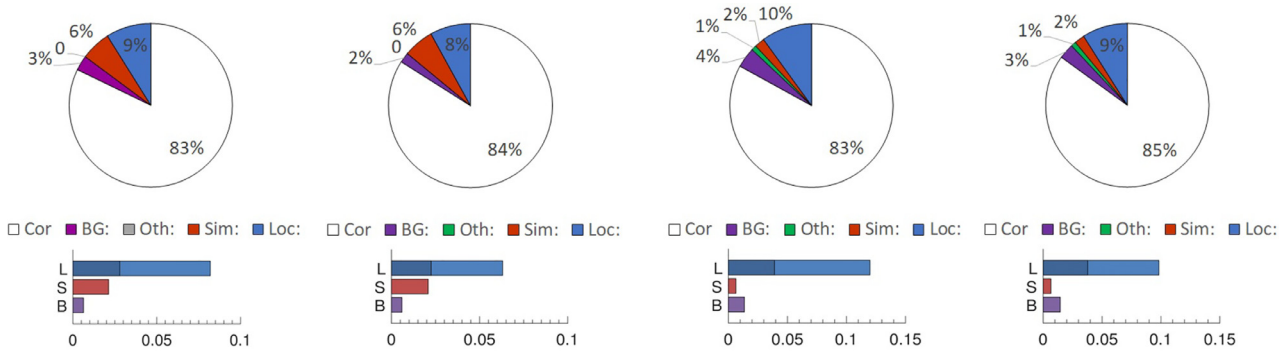

Fig. 7. Analysis of top-ranked false positives. Pie charts: fraction of detections that are correct (Cor) or false positive due to poor localization (Loc), confusion with similar objects (Sim), confusion with other VOC objects (Oth), or confusion with background (BG). Bar graphs: absolute AP improvement by removing all false positives of one type. L: first bar segment is improvement if duplicate or poor localizations are removed; second bar is improvement if localization errors are corrected so that the false positives become true positives. B: no confusion with background and non-similar objects. S: no confusion with similar objects. The first and second columns: results of the baseline Faster R-CNN and C-RPNs on detecting animals. The third and fourth columns: results of the baseline Faster R-CNN and C-RPNs on detecting vehicles.

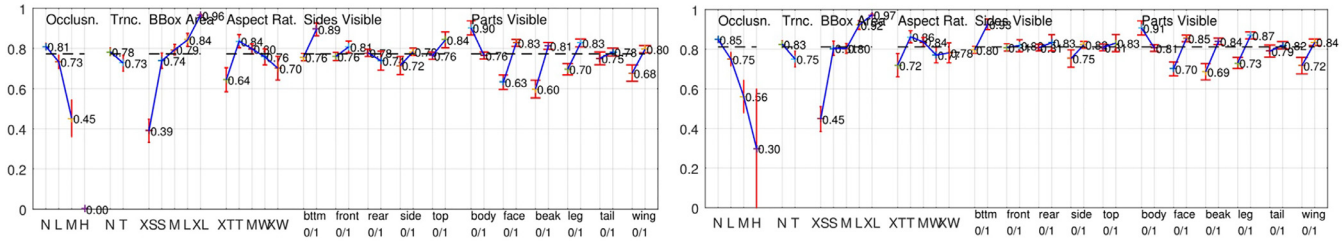

Fig. 8. Characteristics analysis of different bird characteristics on VOC2007 test set: Each plot shows Normalized AP (APN [52]) with standard error bars (red). Black dashed lines indicate overall APN. Key: Occlusion: N=none; L=low; M=medium; H=high. Truncation: N=not truncated; T=truncated Bounding Box Area: XS=extra-small; S=small; M=medium; L=large; XL =extra-large. Aspect Ratio: XT=extra-tall/narrow; T=tall; M=medium; W=wide; XW =extra-wide. Viewpoint / Part Visibility: 1=part/side is visible; 0=part/side is not visible. Left: results of the baseline Faster R-CNN. Right: results of C-RPNs.

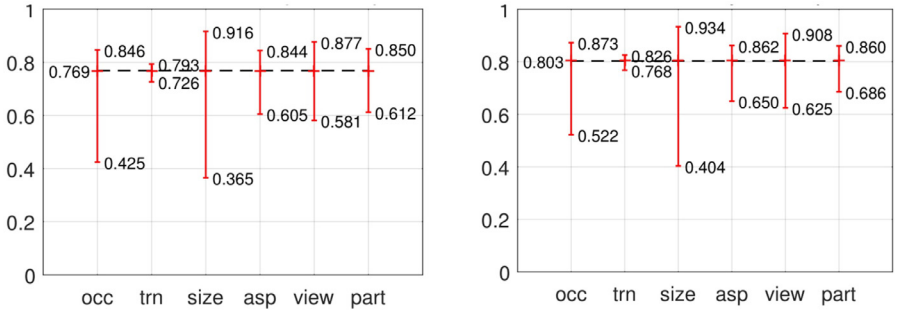

Fig. 9. Summary of sensitivity and impact of object characteristics. The APN are shown over 7 categories of the highest performing and lowest performing subsets within each characteristic (occlusion, truncation, bounding box area, aspect ratio, viewpoint, part visibility). Overall APN is indicated by the dashed line. The difference between max and min indicates sensitivity. The difference between max and overall indicates the impact. Left: results of the baseline Faster R-CNN. Right: results of C-RPNs.

training and learned stronger classifiers. However, just like Faster R-CNN, detection results from C-RPNs have same confusions with similar object categories, partly because binary classifiers in cascade RPN module only indicate samples to be background or object, which has limited promotion on distinguishing the categories of an object.

Fig. 8 visualizes the analysis of different bird characteristics on VOC2007 test set. Performance improvements on these characteristics are explicit since C-RPNs achieves higher average precision on all characteristics of Occlusion, Truncation, Bounding Box Area, Aspect Ratio, Viewpoint and Part Visibility. It is worth mentioning that when occlusion is High, C-RPNs can still recognize some birds while baseline Faster R-CNN detects nothing. Furthermore, extra annotations of seven categories(airplane, bicycle, bird, boat, cat, chair, table) are created in [52] for evaluating robustness of detection approaches. Fig. 9 provides a compact summary of the sensitivity to each characteristic and the potential impact of improving robustness on seven categories. Overall, our C-RPNs achieves higher normalize average precision than Faster R-CNN against all characteristics, indicating its robustness in various scenes. Moreover, sensitivity against all these characteristics are decreased, which verifies that C-RPNs realizes an all-sided improvements upon Faster R-CNN. On the other side, we can see that C-RPNs is sensitive to the bounding box size just like Faster R-CNN and there is still some room to improve.

4.5. Ablation studies

In previous sections, we have shown the efficiency of C-RPNs on several datasets. To further evaluate the individual effect of components of our C-RPNs, we analyze the object detection performance affected by the cascade stages as well as feature chain and score chain. We use BSBDV 2017 in this section.

Please cite this article as: D. Yang, Y. Zou and J. Zhang et al., C-RPNs: Promoting object detection in real world via a cascade structure of Region Proposal Networks, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.08.016

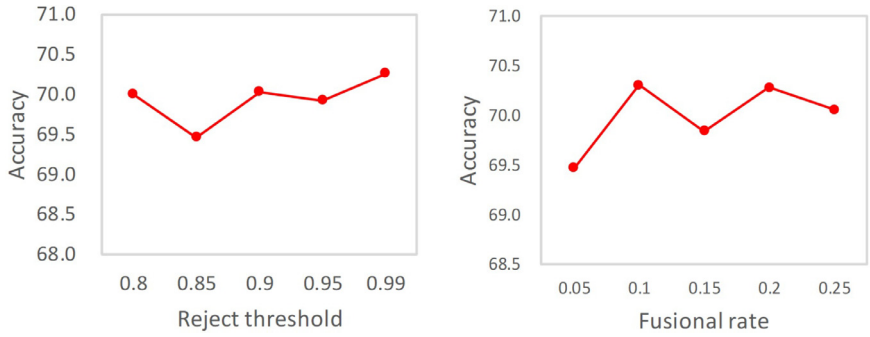

Fig. 10. Grid search for the best reject threshold and fusion rate. Left: accuracy vs reject threshold r; Right: accuracy vs fusion rate λ_f.

Table 4
The impact of cascade stages (BSBDV 2017).

AP of C-RPNs (%)	69.5	69.9	70.3
C-RPNs with Stage *4*	✓	✓	✓
C-RPNs with Stage *3*	✓	✓	✓
C-RPNs with Stage *2*		✓	✓
C-RPNs with Stage *1*			✓

Table 5
The impact of feature/score chain (BSBDV 2017).

AP of C-RPNs (%)	69.4	70.0	69.8	70.3
Feature chain		✓		✓
Score chain			✓	✓

Effects of cascade stages: To learn the efficiency of our C-RPNs with different number of cascade stages, results are summarized in Table 4. We remove different stages of C-RPNs to demonstrate their individual effect. It can be seen that, with stage *3* and stage *4*, C-RPNs achieves AP of 69.5% which already outperforms the baseline Faster R-CNN. Adding stage *2* and stage *1* yields AP of 69.9% and 70.3%, respectively, and it brings 0.4% and 0.4% performance gain, respectively. Finally, the 4-stage cascade RPNs achieves the best performance. These results validate that employing more cascade stages and classifiers in the C-RPNs benefits the detection performance.

Effects of feature chain and score chain: To learn the impact of feature chain and score chain more specifically, Table 5 shows the results of our C-RPNs with or without feature chain and score chain. We set the same parameters for C-RPNs with previous sections but control the usage of feature chain and score chain separately. As shown in Table 5, feature chain is found to be effective in C-RPNs, which brings 0.6% performance gain. When we adapt score chain but without feature chain, the AP is 0.4% higher, which illustrates the efficiency of using score chain as well. The adjustment boosts the performance by 0.9% while both feature chain and score chain are used. These results verify that using features and scores at previous stages as the prior knowledge for the latter stages promotes the final detection.

Selection of reject threshold and fusion rate: To find the best hyper parameters, empirical tests were conducted using different reject threshold r and fusion rate λ_f on BSBDV 2017 through one-dimensional grid search. Fig. 10 shows the impacts of these two factors. As shown, reject threshold r=*0.99* achieved the best AP of 70.31% when the fusion rate was fixed at *0.1*. We then fixed the reject threshold as *0.99* and applied a grid search by changing the fusion rate λ_f. From Fig. 10, the best λ_f is observed as *0.1* with the AP of 70.31%.

5. Conclusion

In this paper, we have constructed C-RPNs, an effective approach for object detection in real world. The essence of our C-RPNs lies in adopting a cascade structure of Region Proposal Networks to discard easy samples during training and learn stronger classifiers. Moreover, a feature chain and a score chain at multiple stages have been proposed to help generating more discriminative representations for mining hard proposals and optimizing cascade classifiers. Finally, a loss function of cascade stages is designed to jointly learn cascade classifiers. Extensive experiments have been conducted to evaluate our C-RPNs on a common benchmark (Pascal VOC) and several challenging datasets collected in wild scenes or realistic traffic scenes. Our C-RPNs achieves competitive results compared with the current state-of-the-arts and outperforms the baseline Faster R-CNN by an obvious margin, demonstrating its efficacy for object detection in real world.

Declaration of Competing Interest

We wish to confirm that there are no known conflicts of interest associated with this publication that could have influenced its outcome.

Acknowledgment

This paper was partially supported by National Engineering Laboratory for Video Technology - Shenzhen Division, Shenzhen Municipal Development and Reform Commission (Disciplinary Development Program for Data Science and Intelligent Computing). Special acknowledgements are given to AOTO-PKUSZ Joint Lab and Peng Cheng Laboratory for their support.

References

[1] M. Everingham, L. Van Gool, C.K. Williams, J. Winn, A. Zisserman, The pascal visual object classes (voc) challenge, Int. J. Comput. Vis. 88 (2) (2010) 303–338.
[2] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C.L. Zitnick, Microsoft coco: common objects in context, in: Proceedings of European Conference on Computer Vision, Springer, 2014, pp. 740–755.
[3] Y. Yao, J. Zhang, F. Shen, X. Hua, J. Xu, Z. Tang, Exploiting web images for dataset construction: a domain robust approach, IEEE Trans. Multimed. 19 (2017) 1771–1784.
[4] Y. Yao, F. Shen, J. Zhang, L. Liu, Z. Tang, L. Shao, Extracting privileged information for enhancing classifier learning, IEEE Trans. Image Process. 28 (2019) 436–450.
[5] Y. Yao, J. Zhang, F. Shen, L. Liu, F. Zhu, D. Zhang, H.T. Shen, Towards automatic construction of diverse, high-quality image dataset, IEEE Trans. Knowl. Data Eng. (2019), doi:10.1109/TKDE.2019.2903036.
[6] W. Guan, Y. Zou, X. Zhou, Multi-scale object detection with feature fusion and region objectness network, in: Proceedings of the 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), IEEE, 2018, pp. 2596–2600.
[7] S. Ren, K. He, R.B. Girshick, J. Sun, Faster r-cnn: towards real-time object detection with region proposal networks, IEEE Trans. Pattern Anal. Mach. Intell. 39 (6) (2017) 1137–1149.

[8] P. Dollar, C. Wojek, B. Schiele, P. Perona, Pedestrian detection: a benchmark, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 304–311.
[9] S. Zhang, R. Benenson, B. Schiele, Citypersons: a diverse dataset for pedestrian detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1, 2017, p. 3.
[10] R. Girshick, J. Donahue, T. Darrell, J. Malik, Rich feature hierarchies for accurate object detection and semantic segmentation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 580–587.
[11] R. Girshick, Fast r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1440–1448.
[12] J. Dai, Y. Li, K. He, J. Sun, R-fcn: object detection via region-based fully convolutional networks, in: Proceedings of Advances in Neural Information Processing Systems, 2016, pp. 379–387.
[13] S. Gidaris, N. Komodakis, Object detection via a multi-region and semantic segmentation-aware cnn model, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1134–1142.
[14] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, Ssd: single shot multibox detector, in: Proceedings of European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[15] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 779–788.
[16] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, S. Belongie, Feature pyramid networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 1, 2017, p. 4.
[17] L. Zhang, L. Lin, X. Liang, K. He, Is faster r-cnn doing well for pedestrian detection? in: Proceedings of European Conference on Computer Vision, Springer, 2016, pp. 443–457.
[18] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, Y. Wei, Deformable convolutional networks, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 764–773.
[19] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, in: Proceedings of Computer Science, 2014.
[20] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[21] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9.
[22] S. Bell, C. Lawrence Zitnick, K. Bala, R. Girshick, Inside-outside net: detecting objects in context with skip pooling and recurrent neural networks, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2874–2883.
[23] S. Zagoruyko, N. Komodakis, Wide residual networks, in: Proceedings of British Machine Vision Conference, 2016.
[24] S. Liu, D. Huang, et al., Receptive field block net for accurate and fast object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 385–400.
[25] C. Peng, T. Xiao, Z. Li, Y. Jiang, X. Zhang, K. Jia, G. Yu, J. Sun, Megdet: a large mini-batch object detector, in: Proceedings of Computer Vision and Pattern Recognition, 2018, pp. 6181–6189.
[26] Y. Liu, R. Wang, S. Shan, X. Chen, Structure inference net: object detection using scene-level context and instance-level relationships, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6985–6994.
[27] K.-K. Sung, Learning and Example Selection for Object and Pattern Detection, 1996.
[28] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, in: Proceedings of European Conference on Computer Vision, Springer, 2014, pp. 346–361.
[29] E. Simoserra, E. Trulls, L. Ferraz, I. Kokkinos, F. Morenonoguer, Fracking deep convolutional image descriptors, in: Proceedings of Computer Science, 2015.
[30] I. Loshchilov, F. Hutter, Online Batch Selection for Faster Training of Neural Networks, arXiv preprint arXiv:1511.06343 (2015).
[31] A. Shrivastava, A. Gupta, R. Girshick, Training region-based object detectors with online hard example mining, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 761–769.
[32] T.-Y. Lin, P. Goyal, R. Girshick, K. He, P. Dollár, Focal loss for dense object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2980–2988.
[33] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, Cascade object detection with deformable part models, in: Proceedings of the 2010 IEEE Conference on Computer vision and Pattern Recognition, IEEE, 2010, pp. 2241–2248.
[34] P. Dollár, R. Appel, S. Belongie, P. Perona, Fast feature pyramids for object detection, IEEE Trans. Pattern Anal. Mach. Intell. 36 (8) (2014) 1532–1545.
[35] R. Xiao, L. Zhu, H.-J. Zhang, Boosting chain learning for object detection, in: Proceedings of Ninth IEEE International Conference on Computer Vision, IEEE, 2003, pp. 709–715.
[36] L. Bourdev, J. Brandt, Robust object detection via soft cascade, in: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2, IEEE, 2005, pp. 236–243.
[37] W. Ouyang, X. Wang, X. Zeng, S. Qiu, P. Luo, Y. Tian, H. Li, S. Yang, Z. Wang, C.-C. Loy, et al., Deepid-net: deformable deep convolutional neural networks for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 2403–2412.
[38] H. Li, Z. Lin, X. Shen, J. Brandt, G. Hua, A convolutional neural network cascade for face detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5325–5334.
[39] B. Yang, J. Yan, Z. Lei, S.Z. Li, Craft objects from images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 6043–6051.
[40] H. Qin, J. Yan, X. Li, X. Hu, Joint training of cascaded cnn for face detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3456–3465.
[41] Z. Cai, N. Vasconcelos, Cascade r-cnn: delving into high quality object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 6154–6162.
[42] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, et al., Hybrid task cascade for instance segmentation, in: Proceedings of the 2019 IEEE Conference on Computer Vision and Pattern Recognition, 2019.
[43] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, Dssd: Deconvolutional Single Shot Detector, arXiv preprint arXiv:1701.06659 (2017).
[44] H. Fan, H. Ling, Siamese cascaded region proposal networks for real-time visual tracking, in: Proceedings of 2019 IEEE Conference on Computer vision and Pattern Recognition, 2019.
[45] D. Yang, J. Zhang, S. Xu, S. Ge, G.H. Kumar, X. Zhang, Real-time pedestrian detection via hierarchical convolutional feature, Multimed. Tools Appl. 77 (19) (2018) 25841–25860.
[46] A. Ess, B. Leibe, K. Schindler, L. Van Gool, A mobile vision system for robust multi-person tracking, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.
[47] C. Wojek, S. Walk, B. Schiele, Multi-cue onboard pedestrian detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 794–801.
[48] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, B. Schiele, The cityscapes dataset for semantic urban scene understanding, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 3213–3223.
[49] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: a large-scale hierarchical image database, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255.
[50] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, et al., Tensorflow: large-scale machine learning on heterogeneous distributed systems, in: Proceedings of Distributed, Parallel, and Cluster Computing, 2015.
[51] T. Kong, F. Sun, A. Yao, H. Liu, M. Lu, Y. Chen, Ron: reverse connection with objectness prior networks for object detection, in: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 1, 2017, p. 2.
[52] D. Hoiem, Y. Chodpathumwan, Q. Dai, Diagnosing error in object detectors, in: Proceedings of European Conference on Computer Vision, Springer, 2012, pp. 340–353.

Dongming Yang received the B.E. degree from Shan Xi University, in 2015, and the M.Sc. degree from University of Chinese Academy of Sciences, in 2018. He is currently pursuing the Ph.D. degree in computer science and engineering from Peking University. His research interests include computer vision and pattern recognition. He is now an intern in the Peng Cheng Laboratory.

YueXian Zou received the M.Sc. degree from the University of Electronic Science and Technology of China, in 1991, and the Ph.D. degree from The University of Hong Kong, in 2000. She is currently a Full Professor with Peking University and the Director of the Advanced Data and Signal Processing Laboratory, Peking University Shenzhen Graduate School. Her research interests mainly in machine learning for signal processing and deep learning and its applications. She was a recipient of the award Leading Figure for Science and Technology by Shenzhen Municipal Government in 2009.

Please cite this article as: D. Yang, Y. Zou and J. Zhang et al., C-RPNs: Promoting object detection in real world via a cascade structure of Region Proposal Networks, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.08.016

Jian Zhang received the B.Sc. degree from East China Normal University, Shanghai, China, in 1982, the M.Sc. degree in computer science from Flinders University, Adelaide, Australia, in 1994, and the Ph.D. degree in electrical engineering from the University of New South Wales, Sydney, Australia, in 1999. He is currently an Associate Professor with the Faculty of Engineering and Information Technology, University of Technology Sydney, Sydney. His current research interests include multimedia processing and communications, image and video processing, machine learning, pattern recognition, media and social media visual information retrieval and mining. He is an Associate Editor for the IEEE Transactions on Multimedia.

Ge Li received the B.E. degree from the Department of Computer Science and Engineering, Dalian University of Technology, Dalian, China, in July 1988 and the M.S. degree from the Institute of Acoustics, Chinese Academy of Science, Beijing, China, in 1991. He is currently is a Professor of Computer Science with Peking University, Shenzhen, China. From 1991 to 1992 he was a Lecturer with the Department of Electrical Engineering, Dalian Maritime University, Dalian, China. In 1995 he was a Summer Engineer at CSG Research Laboratories, Motorola Inc., Libertyville, IL. His general research interests include digital communications, statistical signal processing and machine learning.

Please cite this article as: D. Yang, Y. Zou and J. Zhang et al., C-RPNs: Promoting object detection in real world via a cascade structure of Region Proposal Networks, Neurocomputing, https://doi.org/10.1016/j.neucom.2019.08.016