Focal Loss for Region Proposal Network

Chengpeng Chen, Xinhang Song, and Shuqiang Jiang*

[1] Key Laboratory of Intelligent Information Processing, Institute of Computing Technology, Chinese Academy of Sciences
[2] University of Chinese Academy of Sciences
{chengpeng.chen, xinhang.song}vipl.ict.ac.cn sqjiang@ict.ac.cn

Abstract. Currently, most state-of-the-art object detection models are based on a two-stage scheme pioneered by R-CNN and integrated with region proposal network (RPN), which is served as proposal generation. During the training of RPN, only a fixed number of samples with a fixed *object/not-object* ratio are sampled to avoid class imbalance problem. In contrast to the sampling strategies, *focal loss* is utilized to solve the class imbalance problem by down-weighting the losses of vast number of easy samples, which is encountered in one-stage detection methods. Inspired by this, we investigate the adaptation of focal loss to RPN in this paper, which allow us to train RPN free of the sampling process. Based on Faster R-CNN, we adapt focal loss to RPN and the experimental results on PASCAL VOC 2007 and COCO datasets outperform the baseline, which shows the efficiency of the proposed method and implies that focal loss can be applied to RPN directly.

Keywords: Object detection · Region proposal network · Focal loss.

1 Introduction

In this era of deep learning, most object detection models with state-of-the-art performance are based on a two-stage scheme [1, 7, 8, 5, 26, 25], where a sparse set of proposals are generated at the first stage, followed by regional object classification and coordinate regression at the second stage. The process of generating proposals has developed from off-line methods, such as Selective Search [15] and objectness [16], to integrated learning ones [18, 19, 1], in which Region Proposal Network (RPN) has become a standard component of these state-of-the-art two-stage methods. During the training of RPN, candidate proposals are first sampled among pre-located dense anchors, and then fed to the classifier of *object/not-object* and regressor. Within those dense anchors, the samples with class *object/not-object* are very imbalanced, particularly, the samples of *not-object* are much more than the ones of *object*, which make it difficult to train a classifier with regular policies. Thus, as a usual strategy, only a fixed number of anchors with a fixed *object/not-object* ratio, *e.g.,* 256 and 1:1 [1], are sampled for

* Corresponding author

training. Although such constraint in sampling progress can balance the samples, it also results in losing the diversity of proposals. Instead of constraining the sampling progress, we investigate this imbalance problem from the aspect of designing desired loss function during training.

The class imbalance problem is also encountered in one-stage detection models [11–13, 3], in which different types of example sampling strategies [14, 11, 23] are proposed to address this problem. However, Lin *et al.* [3] claims that it is the vast number of easy samples that overwhelms the detectors. Thus, they propose to take all pre-located dense anchors for training with a dynamically cross entropy loss, called *focal loss*, which prevents these easy samples from overwhelming the training process by down-weighting the losses of easy samples.

In this paper, we investigate the adaptation of focal loss to RPN (see Section 3.3), such that much more samples can be included for training while free of the training problems caused by class imbalance. By replacing standard cross entropy loss in RPN with focal loss, RPN can be trained directly with no need for specially designed sampling strategies. Besides, due to the full convolutional implementation of RPN, no extra computation cost is required. We take Faster R-CNN [1, 2] as our baseline model and conduct the experiments on PASCAL VOC 2007 [24] and COCO [10] detection benchmarks. The experimental results show the efficiency of the proposed method, implying that this sampling free strategy can be directly applied to RPN, so as to all the state-of-the-art two-stage detectors.

2 Related Work

Two-stage Detectors. With the fast development of deep leaning [9] over past few years, two-stage object detectors [6, 4, 7, 1, 5, 8, 26, 25] have become one of the fashion of object detection methods. In the two-stage methods, a sparse set of candidate proposals with high probabilities of containing objects are first generated [15, 18, 19, 1], followed by a second stage of object classification and coordinate regression. Empowered with deep neural networks [9, 20–22] and a series of improvements in both speed and accuracy [6, 4, 7, 1], the whole detection system is integrated into a single network, *i.e.*, the widely-used Faster R-CNN [1] framework. Many works to extend this framework have been conducted [8, 5, 26, 25]. We also utilize Faster R-CNN as our base model to investigate the adaptation of focal loss to RPN in this paper.

Region Proposal Methods. As the first stage in the two-stage scheme, region proposal methods have been developed from pioneering off-line methods, such as Selective Search [15] and objectness [16], to integrated learning ones [18, 19, 1], in which RPN integrated this proposal process into the base networks by sharing their convolutional layers. During training, dense anchors are pre-located first, to which RPN applies *object/not-object* classification and class-agnostic regression, while for inference, it generates a sparse set of proposals for the second stage by applying coordinate refinements and non-maximum suppression (NMS) to the

dense anchors. RPN enables the end-to-end training of the two-stage detectors, and has become one of their components.

It is worth to note that not all the pre-located anchors are employed for training due to its class imbalance problem, that is, majority of the dense anchors are easy samples with class *not-object*. And if all these anchors are taken into account, they would overwhelm the detector during training. In this paper, we focus on this class imbalance problem in RPN.

Class Imbalance. As same as RPN in two-stage detectors, one-stage detectors also encounter class imbalance during training [11–13, 3], and some types of example sampling strategies are often the employed solutions [14, 11, 23]. In contrast, Lin *et al.* [3] propose a novel type of loss function, called focal loss, to down-weight the losses of easy samples, so as to include all samples for training and handle the class imbalance. Inspired by this work, we try to adapt focal loss to RPN such that we can also avoid the sampling process during the training of RPN.

Loss Function Design. There are two tasks, classification *(cls)* and bounding box regression (*reg*), in both first and second stage of these two-stage methods, which classifies the anchors/proposals to a specific class and regresses the bouncing boxes, respectively. The *cls* loss is taken as standard cross entropy loss, while for binary *cls*, it is shown as:

$$\begin{aligned} CE\left(p,y\right) &= \frac{1}{N_{cls}}\sum_{i} CE\left(p_i,y_i\right) \\ &= \frac{1}{N_{cls}}\sum_{i} y_i \log(p_i) + \left(1-y_i\right)\log\left(1-p_i\right) \end{aligned} \tag{1}$$

where y_i is the label, p_i is the estimated probability for each sample, and N_{cls} is the number of samples and taken as a normalization term. For multi-class *cls* task, the cross entropy loss can be extended straightforwardly.

For the *reg* task, smoothed L_1 loss [7] is applied as:

$$smooth_{L_1}\left(x\right) = \begin{cases} 0.5x^2 & if\ |x| \leq 1 \\ |x| - 0.5 & otherwise \end{cases} \tag{2}$$

where x is the difference between anchors/proposals and bounding boxes of ground true.

We note that the *cls* task is *object/not-object* binary classification in the first stage, *i.e.*, RPN, while in the second stage, it is taken as multi-class ones to classify foreground classes/background. For the *reg* tasks in both stages, the smooth L_1 loss is only computed on anchors/proposals belong to *object*/foreground classes.

We follow the literature and use these losses in our model except that we use focal loss in *cls* task of RPN instead of cross entropy loss, such that we can include much more anchors for training.

3 Focal Loss for RPN

As a Region Proposal Network (RPN) based detection model, Faster R-CNN [1] is taken as our base model for evaluating the adaption of focal loss to RPN. In the following of this section, we will briefly review RPN in Faster R-CNN (Section 3.1), focal loss [3] applied in detection models (Section 3.2), and finally introduce our focal loss equipped RPN (Section 3.3).

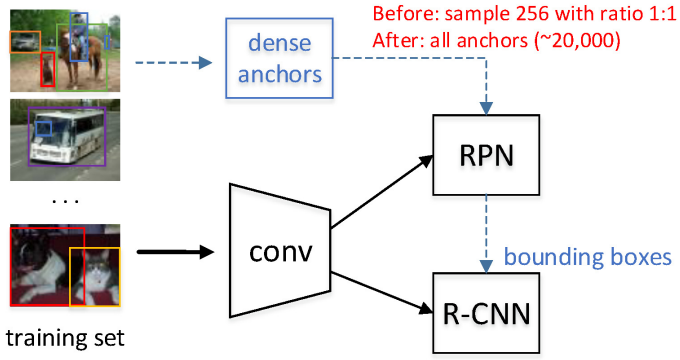

Fig. 1. The training process of Faster RCNN with focal loss. The blue/dashed lines indicate the generation/feeding of anchors or bounding boxes.

3.1 RPN in Faster R-CNN

Faster R-CNN is a widely-used two-stage detection model which integrates RPN to generate proposal regions, enabling an end-to-end detection model. Based on RPN, the two-stage detection approaches develop fast and achieve good performance in recent years [1, 5, 8, 25, 26].

As Figure 1 shown, RPN shares convolutional (conv) layers with base detection network, *e.g.,* first 5 conv layers in Zeiler and Fergus model (ZF net) [20], 13 in VGG16 [21] and first 4 blocks in ResNet [22]. On the top of these shared conv layers, RPN is included as external branch for *cls* and *reg*, consisting of an $3 * 3$ conv layer followed by two sibling fully-connected layers (or $1 * 1$ conv layers) for *cls* and *reg*, respectively. Note that, RPN only classifies *object/not-object* for each anchor, where we also apply sigmoid (1 for *object* and 0 for *not-object*) and softmax (as usual in two-stage detectors) for our focal loss adaptation, which will be introduced in Section 3.3. Besides, RPN regresses bounding boxes via

refining pre-fixed anchors, which are centered at each position of the top shared conv layer. k anchors at each position are taken according to different scales and aspect ratios, *e.g.,* 3 scales and 3 aspect ratios result in $k = 9$ anchors in [1]. Therefore, with a typical image scale $\sim 600 * 1000$ and feature stride 16 of the shared conv layers [20–22], $\sim 20,000$ anchors are obtained in total, in which the numbers of *object/not-object* are very imbalanced, *e.g.,* $\sim 1:1000$. However, only fixed number of anchors are sampled for training to ensure a relative balanced samples (in [1], 256 anchors with ratio $1:1$). The loss function of RPN is formulated as:

$$L_{RPN} = \frac{1}{N_{cls}} \sum_i CE\left(p_i, p_i^t\right) + \frac{1}{N_{reg}} \sum_i I\left(t_i^t\right) L_{reg}\left(t_i, t_i^t\right) \tag{3}$$

where N_{cls} and N_{reg} are the normalization terms, *e.g.,* 256 in [1], and p_i^t and t_i^t are the *cls* label and *reg* target, respectively. The first term of Eq. (3) stands for standard cross entropy loss, while the second stands for the *reg* loss, where standard smooth L_1 loss [7] is applied, and $I\left(t_i^t\right)$ is an indicator function. The loss here is only computed on the sampled anchors.

3.2 Focal Loss for Detection

Different to two-stage methods, one-stage detection models [11–13, 3] do not generate proposal first, but directly classify and regress the anchors (or priors) to the class and bounding boxes of ground true like RPN, respectively. The detection results are obtained in a single run, making them more efficient in the speed of detection. However, they also suffer the same imbalanced sample problem as RPN, and some types of examples sampling [14, 11, 23] are often the applied solutions. In [3], all pre-located anchors are used for training instead of a relative small number of sampled ones. The authors claim the affects of the imbalanced problem is that the accumulated loss from the vast number of easy samples overwhelms the detector [3]. Therefore, in order to address with this imbalanced problem, it proposed focal loss to down-weight the loss of the easy samples. Focal loss is a dynamically scaled cross entropy loss, which can be formulated as:

$$FL(p_t) = -\alpha_t (1 - p_t)^\gamma \log(p_t) \tag{4}$$

where for binary classification, $p_t \epsilon [0, 1]$ is the probability for the ground true class, $\alpha_t \epsilon [0, 1]$ the re-weighting factor to balance positive and negative samples, and $\gamma \geq 0$ a hyper-parameter. Note that, when $\alpha_t = 0.5, \gamma = 0$, focal loss deforms to standard cross entropy loss.

As in Eq. (4), for those easy samples (p_t close to 1), the scale term $(1 - p_t)^\gamma$ down-weights the loss greatly; thus, it leads the model to focus more on hard samples. Through this dynamically scaled loss, the model can avoid the problem of the model being overwhelmed by much more easy samples, so as to include all the anchors for training.

3.3 Focal Loss for RPN

To investigate the application of focal loss to RPN, we re-formulate the loss of RPN with focal loss as:

$$L_{RPN-FL} = \frac{\lambda_{fl}}{N^{'}_{cls}} \sum_i FL\left(p^t_i\right) + \frac{1}{N^{'}_{reg}} \sum_i I\left(t^t_i\right) L_{reg}\left(t_i, t^t_i\right) \tag{5}$$

where we simply the replace the cross entropy loss with focal loss and use all anchors ($\sim 20,000$ per image) for training instead of those sampled. λ_{fl} is served as a balancing weight. Note that, in the first term of Eq. (5), we set $N^{'}_{cls} = |p^t_i \epsilon\, object|$, which means the *cls* loss is normalized with number of *object* samples in this dense anchor scenario, while in the second term, we set $N^{'}_{reg} = 2 * |p^t_i \epsilon\, object|$.

Figure 1 illustrates our adaptation of focal loss in RPN. In contrast to only training with a part of anchors as previous works [1, 8, 26, 25], all the generated dense anchors are taken for training with our adaptive focal loss. The focal loss equipped RPN is integrated into Faster R-CNN framework [1, 2] in the following form:

$$L = L_{RPN-FL} + L_{RCNN} \tag{6}$$

where L_{RCNN} includes multi-class *cls* loss and class-aware *reg* loss, and we do not modify it so as to verify the effect of focal loss applied in RPN on the whole detection system.

To get the probability p_t in Eq. (4), we utilize two output functions, softmax and sigmoid. For output with softmax, we get two scores $[p_p, p_n]$ implying *object* and *not-object*, respectively, and get $p_t = p_p$ if the anchor matches with *object* label, while $p_t = p_n$ if the anchor matches with *not-object* label. For output with sigmoid, only one score p_s is get and $p_t = p_s$ for *object* label, while for *not-object* label $p_t = 1 - p_s$. These two output function will be compared in the following experiments.

Implement Details. This work is based on the public TensorFlow implementation of Faster R-CNN[3][2], and we follow most of the parameter settings from the original implementation. We use stochastic gradient descent (SGD) for optimization and set momentum as 0.9 and weight decay as 0.0001. The model is trained with one image per iteration following [2], and the only data augmentation strategy is to randomly flip the training images. ImageNet [9] pre-trained VGG16 [21] is used as our base network, and the conv1 and conv2 layers are fixed.

We set the base learning rate as 0.001 for first 50k/350k iteration and decrease by 10 for next 20k/140k for PASCAL VOC 2007/COCO datasets. For the hyper-parameters, we set $\alpha_t = 0.25, \gamma = 2$ and $\lambda_{fl} = 0.1$ by default, and they will be evaluated in the following experiments.

[3] https://github.com/endernewton/tf-faster-rcnn

4 Experiments

We evaluate our model on PASCAL VOC 2007 [24] and COCO [10] detection benchmarks and follow the standard data splits. Average precision (AP) is reported following the literature. An image scale of 600 pixels is applied for both training and test [1, 2]. Note that, for fair comparison, we only modify the loss function and do not include any additional parameters in all our experiments, except the model of sigmoid output contains less parameters, where we reduce the output from two to one.

PASCAL VOC 2007. PASCAL VOC [24] has been a classical dataset for computer vision tasks, *e.g.,* classification and detection and segmentation. In the following experiments, we also utilize this dataset for evaluating our model. It contains 20 object categories for detection, and there are 2.47 objects in each image in average. We use the trainval split for training, and test split for evaluation. which consist of 5,011 and 4,952 images, respectively. Average precision (AP) is reported with the IOU threshold set as 0.5.

COCO 2014. As a more complicate dataset, COCO [10] has been a challenging benchmarks of object detection, and is most widely-used for evaluating various detection models. It contains 80 object categories for detection, and there are 7.58 objects in each image in average. We use COCO 2014 in our experiments, which contains of 82,783 images for training, 40,504 for validation and 40,775 for test. Due to the unavailable of the ground true of test split, we follow the literature [10, 2] to re-split the dataset to train+valminumsminival and minival. During test, COCO employs a more strict metric, where average precision (AP) is computed with different IOU thresholds, *i.e.*, [0.5 : 0.95] and report their average. Besides, the performances for different scales, *i.e.*, small/middle/large, are also reported.

Table 1. Parameters evaluation Detection average precision (%). All use Faster R-CNN on VGG16. For each column, we only change the corresponding parameter and keep others as default. The missing values mean that the model failed in those settings.

	γ			α			λ_{fl}		
	1.0	2.0	3.0	0.25	0.5	0.75	0.1	0.2	0.3
FL-sigmoid	**70.8**	70.7	70.7	70.7	70.5	70.5	70.7	70.5	70.5
FL-softmax	-	**71.2**	71.1	**71.2**	70.9	70.7	**71.2**	-	-

4.1 Parameters Evaluation

We evaluate the hyper-parameters in Table 1. FL-softmax and FL-sigmoid stand for Faster R-CNN with focal loss equipped RPN which output with softmax and

sigmoid, respectively, as introduced in Section 3.3. As the table shown, FL-softmax always gain a higher performance than FL-sigmoid, while the latter performs much more stable under different parameter settings. We assume that it is the saturation of sigmoid function that leads the model less sensitive to the hyper-parameters and also stuck the optimization process, which result in inferior performances. For those several failed scenarios in FL-softmax, the large scale of focal loss computed on all anchors may be the cause, *e.g.,* small exponent $\gamma \leq 1$ or large loss scale $\lambda_{fl} \geq 0.2$ could result in the exposure of loss and further hurt the optimization process. Thus, for FL-softmax, we should design the hyper-parameters more carefully to make the computed focal loss in a reasonable scale, so as to train the model correctly.

Table 2. VOC 2007 test Detection average precision (%). These models use the default hyper-parameters except that $\gamma = 1$ in FL-sigmoid. In baseline+FL, we combine focal loss with the original RPN. *Baseline we trained using the public implementation.

	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow
baseline*	70.9	71.0	78.8	69.5	55.5	56.0	79.5	84.9	81.3	50.2	79.8
FL-sigmoid	70.8	71.4	78.0	68.7	55.6	56.7	**80.5**	85.8	84.4	47.9	77.9
FL-softmax	**71.2**	**72.1**	**79.4**	**72.4**	55.1	**57.7**	78.0	84.5	**85.4**	48.5	**80.6**
baseline+FL	**71.2**	69.6	78.7	72.0	**56.4**	57.4	79.7	**85.1**	84.8	**51.2**	77.5
		table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
baseline*		**65.9**	80.6	**84.0**	74.6	77.3	44.2	73.0	65.7	72.7	**73.4**
FL-sigmoid		64.8	**81.3**	83.7	**77.3**	76.9	41.8	72.3	65.4	73.8	71.3
FL-softmax		63.0	**81.3**	83.0	74.4	77.5	**44.3**	**74.0**	63.5	**76.4**	73.2
baseline+FL		63.5	81.2	83.3	75.8	**77.7**	43.6	72.1	**67.0**	75.1	73.2

4.2 Performance Comparison

Table 2 shows the detection results of baseline and our models which are adapted with focal loss. The performances are comparable to the baseline, implying that focal loss can be modified to apply in RPN directly to replace the sampling mechanism, but only with a mirror impact on the performance.

As Table 2 shown, however, when slightly changing the mAP metric (0.1% lower in FL-sigmoid and 0.3% higher in FL-softmax), the performance of each class changes obviously, *e.g.,* obtaining 2.9% lower for 'table' class and 4.1% higher for 'cat' class in FL-softmax, which may indicate that focal loss is complementary to standard cross entropy loss. Inspired by this, we simply add focal loss to the original RPN, denoted as baseline+FL in Table 2, which obtains the same performance as FL-softmax and also mirror improvements over baseline. Specifically, in baseline+FL, focal loss is computed on all anchors as before while cross entropy loss is computed on sampled anchors, and these two losses are directly combined by average. Figure 3 displays some examples on PASCAL VOC 2007 detected by model baseline+FL, where we get the satisfactory results with a wide range of scales and aspect ratios.

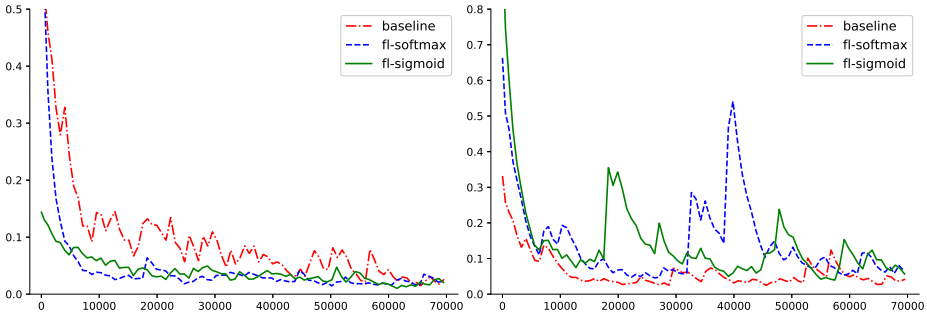

Fig. 2. Loss curves of RPN. **Left:** *cls* loss. **Right:** *reg* loss. The loss curves in baseline+FL display similar to baseline, so we omit them for simplicity.

4.3 Training Process in RPN

To further analyze the influence of focal loss on RPN, we plot the *cls* and *reg* losses during training in Figure 2. In the *cls* loss curve, the two focal loss equipped RPNs converge much faster and more stable than baseline. This effect is benefited from the intrinsic characteristic of focal loss that it is capable of training with much more anchors. For the *reg* loss curve, however, these two models perform worse than baseline; they are much unstable and have large scale. This may be the reason why focal loss can not boost RPN (and Faster R-CNN) greatly like one-stage detection model [3], *e.g.,* after we get the satisfied scores for all the anchors, these anchors can not be refined well to produce satisfied proposals for R-CNN, which may affect the performance of the whole detection system. This may implies that the training signals produced by focal loss is conflict to those from bounding box regression in some terms.

Besides, it is worth to note that, RetinaNet [3], the network first applied focal loss to detection, decouples the *cls* and *reg* tasks into two sub-networks, and thus avoids this conflict signals problem. In this work, we only follow the original design of RPN where these two tasks share the same networks except the task specific layers. Thus, decoupling *cls* and *reg* tasks like RetinaNet in our focal loss equipped RPN may further improves the model performance. Other ways to make focal loss more compatible with bounding box regression can also be taken into consideration, and this will be our future work.

Table 3. COCO 2014 minival object detection average precision (%). Legend same as table 2. *Baseline we trained using the public implementation.

	AP	AP-.5	AP-.75	AP-S	AP-M	AP-L
baseline*	26.5	46.7	27.2	11.8	30.4	37.5
FL-softmax	26.6	46.7	27.4	**12.0**	**30.9**	37.3
baseline+FL	**27.0**	**47.5**	**27.7**	**12.0**	30.8	**37.9**

4.4 More results on COCO

We also conduct experiments of the focal loss equipped RPN in COCO 2014 dataset [10], where we use train+valminumsminival and minival split following [10, 2]. As table 3 shown, FL-softmax performs comparable to baseline, while baseline+FL is superior in all the metrics. In terms of the performance difference of baseline+FL in these two datasets, we assume that it is the difference between the statistics of each dataset that counts; COCO contains much more objects in each image than PASCAL VOC 2007 (7.58 vs 2.47 in average), which may results in differences in the training process, *i.e.*, the anchors for computing focal loss in COCO is not such imbalanced like PASCAL VOC 2007. That is, in dataset with dense objects, such as COCO, focal loss combined with standard cross entropy loss may work better than either of them alone.

In other aspects, the original implementation [2] claims that the performance on COCO could continue to improve if we train with more iterations, *e.g.*, 900k/1190k; thus the reason why baseline+FL performs better than baseline and FL-softmax may be its fast convergence characteristic. So, whether the statistic difference or convergence characteristic contribute to performance difference is further to be explored.

We note that the training processes also display the same trends as in Section 4.3. And these experimental results show that focal loss is also adaptable to more complicate datasets.

5 Conclusion

In this work, we investigate how to adapt focal loss to train RPN without applying the sampling strategy. By down-weighting the losses of those vast numbers of easy samples, focal loss can intrinsically handle the class imbalance problem and prevent their losses from overwhelming the detector. Using focal loss is capable of including much more samples for training. Thus, RPN can also take all anchors into account for training via replacing standard cross entropy loss with focal loss or simply combining them. As the experiments conducted on PASCAL VOC 2007 and COCO shown, it is feasible to train RPN without particularly designed sampling. We also discuss the compatibility between focal loss and bounding box regression in RPN, and this is left as future work.

Acknowledge

This work was supported in part by the National Natural Science Foundation of China under Grant 61532018, in part by the Lenovo Outstanding Young Scientists Program, in part by National Program for Special Support of Eminent Professionals and National Program for Support of Top-notch Young Professionals, in part by the National Postdoctoral Program for Innovative Talents under Grant BX201700255.

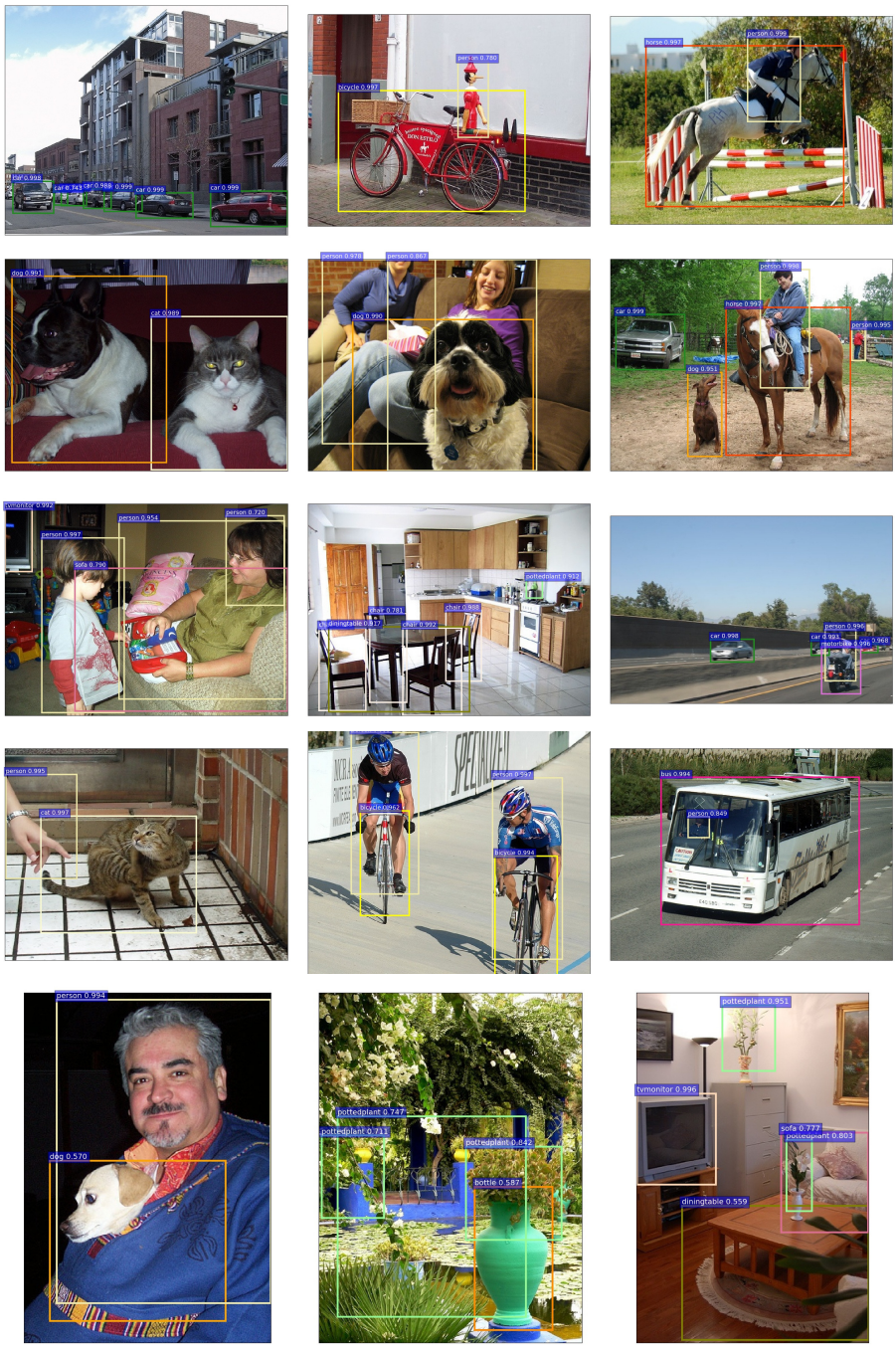

Fig. 3. Examples on PASCAL VOC 2007 detected by baseline+FL. The score threshold for display is set as 0.5

References

1. Ren, S., He, K., Girshick, R., Sun, J.: FasterR-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: NIPS. (2016)
2. Chen, X., Gupta, A.: An Implementation of Faster R-CNN with Study for Region Sampling. arXiv:1702.02138. (2017)
3. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Doll´ar, P.: Focal Loss for Dense Object Detection, In: ICCV. (2017)
4. He, K., Zhang, X., Ren, S., Sun, J.: Spatial pyramid pooling in deep convolutional networks for visual recognition. In: ECCV. (2014)
5. Dai, J., Li, Y., He, K., \Sun, J.: R-FCN: Object detection via region-based fully convolutional networks. In: NIPS. (2016)
6. Girshick, R., Donahue, J., Darrell, ,T. Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: CVPR. (2014)
7. Girshick, R.: Fast R-CNN. In: ICCV. (2015)
8. He, K., Gkioxari, G., Doll´ar, P., Girshick, R.: Mask RCNN. In: ICCV. (2017)
9. Krizhevsky, A., Sutskever, I., Hinton, G.: ImageNet classification with deep convolutional neural networks. In: NIPS. (2012)
10. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Doll´ar, P., Zitnick, C.L.: Microsoft COCO: Common objects in context. In: ECCV. (2014)
11. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S: SSD: Single shot multibox detector.: In: ECCV. (2016)
12. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: CVPR. (2016)
13. Redmon, J., Farhadi, A.: YOLO9000: Better, faster, stronger. In: CVPR. (2017)
14. Shrivastava, A., Gupta, A., Girshick, R.: Training region based object detectors with online hard example mining. In: CVPR. (2016)
15. Uijlings, J. R., Van de Sande, K. E., Gevers, T., Smeulders, A. W.: Selective search for object recognition. IJCV. (2013)
16. Alexe, B., Deselaers, T., Ferrari, V.: Measuring the objectness of image windows. IEEE TPAMI, 34(11). (2012)
17. Zitnick, C. L., Doll´ar, P.: Edge boxes: Locating object proposals from edges. In: ECCV. (2014)
18. Szegedy, C., Reed, S., Erhan, D., Anguelov, D.: Scalable, high-quality object detection. arXiv preprint arXiv:1412.1441v2 (2014)
19. Erhan, D., Szegedy, C., Toshev, A., Anguelov, D.: Scalable object detection using deep neural networks. In: CVPR. (2014)
20. Zeiler, M. D., Fergus, R.: Visualizing and understanding convolutional neural networks. In: ECCV. (2014)
21. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. In: ICLR. (2015)
22. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: CVPR. (2016)
23. Sung, K.-K., Poggio, T.: Learning and Example Selection for Object and Pattern Detection. In: MIT A.I. Memo No. 1521. (1994)
24. 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)
25. Singh, B., Davis, L. S.: An Analysis of Scale Invariance in Object Detection – SNIP. In: CVPR. (2018)
26. Hu, H., Gu, J., Zhang, Z., Dai, J., Wei, Y.: Relation Networks for Object Detection. In: CVPR. (2018)