Scale-balanced loss for object detection

Scale-balanced loss for object detection

Kai Shuang[a,d], Zhiheng Lyu[a,*], Jonathan Loo[b], Wentao Zhang[c]

[a] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing 100876, PR China
[b] School of Computing and Engineering, University of West London, W5 5RF, UK
[c] Yituyishu(Beijing) Technology Company Ltd., Beijing, PR China
[d] Science and Technology on Communication Networks Laboratory, Shijiazhuang, PR China

ARTICLE INFO

Article history:
Received XXXX
Revised XXXX
Accepted XXXX
Available XXXX

Keywords:
Object detection
Neural network
Matching imbalance

ABSTRACT

Object detection is an important field in computer vision. Nevertheless, a research area that has so far not received much attention is the study into the effectiveness of anchor matching strategy and imbalance in anchor-based object detection, in particular small object detection. It is clear that the objects with larger sizes tend to match more anchors than smaller ones. This matching imbalance may result in poor performance in detecting small objects. It can be alleviated by paying more attention to the objects that match with fewer anchors. We propose an innovative flexible loss function for object detection, which is compatible with popular anchor-based detection methods. The proposed method, called the scale-balanced loss, does not add any extra computational cost to the original pipelines. By re-weighting strategy, the proposed method significantly improves the accuracy of multi-scale object detection, especially for small objects. Comprehensive experiments indicate that the scale-balanced loss achieved excellent generalization performance when separately applied to some popular detection methods. The scale-balanced loss attained up to 15% improvements on recall rates of small and medium objects in both the PASCAL VOC and MS COCO dataset. It is also beneficial to the AP result on MS COCO with an improvement of more than 1.5%.

1. Introduction

Object detection plays an important role in many applications of computer vision, such as face recognition [1], person re-identification [2], autonomous driving [3], and medical image analysis [4], etc. In recent years, a lot of detectors based on deep learning are proposed to improve the accuracy and efficiency of object detection models [5–9].

Object detection aims to identify all objects of interest in the image data. Due to the different sizes, shapes, and locations of objects, object detection is more challenging than image classification [10–13]. According to the steps of generating results, many popular detectors can be generally divided into two categories: the multi-stage methods and the one-stage methods. Multi-stage methods [14–16] first generate candidate prior boxes and then refine them in the following part. This strategy not only can alleviate the imbalance between background and foreground, but also give more accurate localization results. Many multi-stage methods have achieved the highest accuracy on some benchmarks, such as PASCAL VOC [17] and MS COCO [18], but they usually suffer from the high computational cost. In order to reduce the computational cost, some one-stage methods have been proposed [19,20]. They integrate the classification and localization results together in a more efficient pipeline, which is more feasible for real-time image analysis.

No matter which kind of pipelines for detection is used, most existing methods adopt a set of prior boxes called æanchorsg in their models to match the ground truth. In Faster-RCNN [14], positive samples refer to the anchors which have an IoU[1] overlap higher than 0.7 with any ground truth box while those have no IoU higher than 0.3 are considered as negative ones. This matching strategy determines which anchors are responsible for predicting the ground truth. Some slightly different strategies are applied in lots of other pipelines [21–24] to play a similar role. Due to the nonuniform distribution of anchors, different objects can match with different numbers of anchors. The objects with more matched anchors might be dominant in the optimization of loss. As shown in Fig. 1, for different objects in VOC 07train + 12trainval set, there exists an extreme imbalance of matching times with anchors in SSD300 settings (with data argument in SSD300). It is clear that

[1] Intersection over Union.

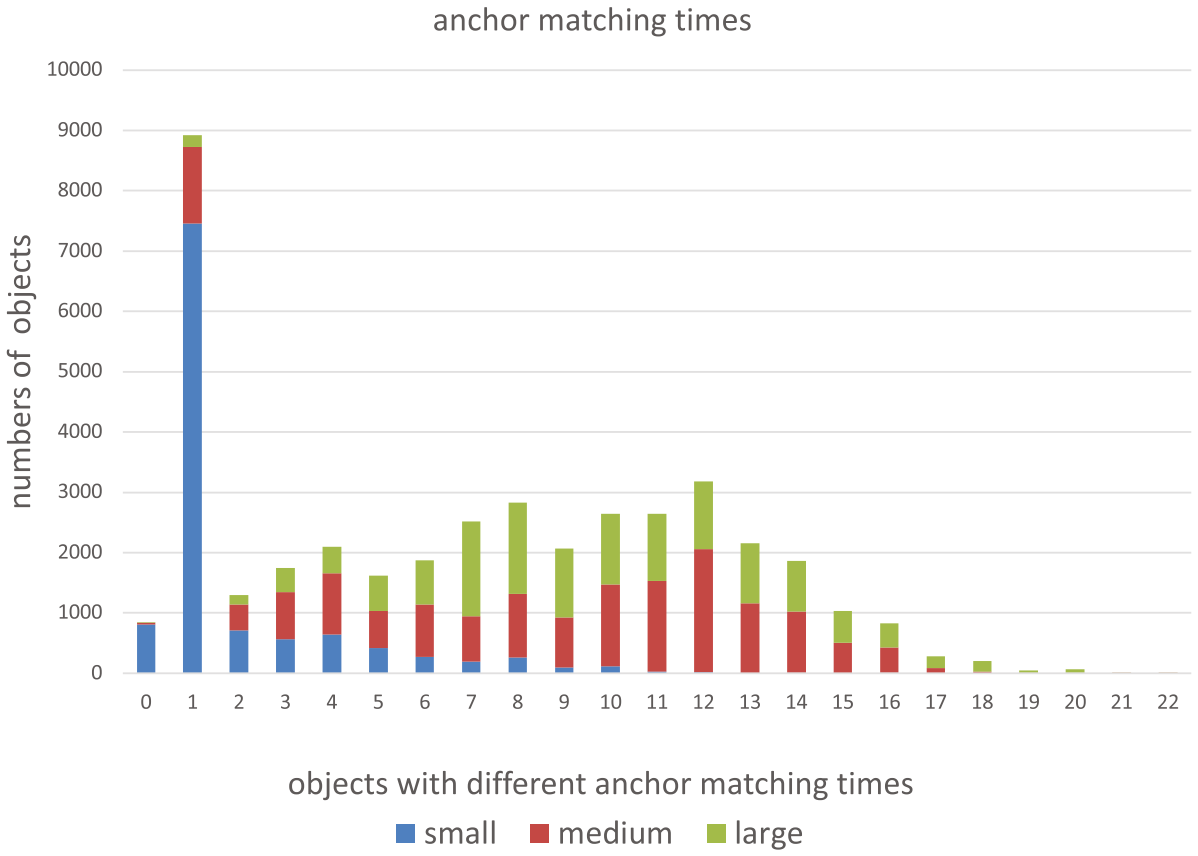

Fig. 1. The matching imbalance during anchors matching process. Taking SSD300 on VOC dataset as an example. The first bar with index 0 indicates the number of objects which match no anchor. The bar with index 1 indicates the number of objects which match one anchor, and so on. We can see that in the existing anchor matching strategy, nearly 9000 objects can only match one anchor, and most of them are small objects. On the other hand, there is about one third of objects can match more than 10 anchors. Statistics show that the existing anchor matching strategies is unbalanced, which is related to the size of objects.

objects with larger sizes tend to match more anchors than smaller ones. This matching imbalance may result in poor performance in detecting small objects.

Some existing methods focus on the general characteristics of small objects. However, the matching imbalance mentioned above has not been paid enough attention. The characteristics include less information of their own, larger probabilities to be confused with background and higher precision requirements for localization [25]. But without the balanced anchor matching results, there still remains a giant gap between large objects and small ones. It causes such a serious conflict and competition that compared with large ones, small objects are so difficult to be detected, and they require special treatment in existing detection pipelines. In this circumstance, the information of small objects cannot be efficiently explored, and the image background can also easily overwhelm them.

In order to obtain a more reasonable anchor matching result, it is a natural way to improve the existing anchor matching strategies directly [22,26,27]. Different methods have been tried, but still cannot make a breakthrough. YOLO9000 [28] runs k-means clustering on the training set to automatically find better anchors. FaceBoxes [29] uses more small anchors to match faces, which improves the recall rate of small faces. MetaAnchor [30] makes use of a dynamical prior boxes generating method for robust bounding box distributions instead of manual selection. Although these data-based improved methods generate better anchors, the imbalance of anchor matching results remains as great as ever. Besides, the growing computational cost also limits the application of these methods.

Except for improving the original matching and generating strategies, designing unique network architectures is also an alternative method for small obejct detection. RFBNet [31] uses RFB blocks to focus on small anchors which are most affected by the unbalanced matching. Refinedet [32] and FPN [33] make top-down architectures and alleviate the imbalance from different perspectives.

Instead of making efforts on the network architecture, the loss function and weighting strategy [27,34] can directly alleviate the anchor matching imbalance. In this paper, we designed a new loss function called scale-balanced loss to replace the counterpart in previous approaches for maintaining the matching balance. The scale-balanced loss puts a weighted operation on the original one, which can reduce the proportion of objects with more matching times and enlarge the weight of objects with fewer matches. It is a compensation strategy for different sizes of objects. Experiments show that our proposed method achieved excellent generalization performance. It makes significant improvements on four popular models, SSD [20], FSSD [19], DSSD [35], and RefineDet [32]. We also explored the effects of different weighting forms to make a comparison. At last, we showed the specific impact on different sizes of objects to prove the effectiveness of the scale-balanced loss in detail.

The main contributions of this work are summarized as follows.

- We pointed out an imbalance among objects of different scales in existing anchor matching strategies, which may lead to poor performance in detecting multi-scale objects, especially for small objects. We found that this imbalance can be alle-

viated by paying more attention to the objects that matched fewer anchors in the loss function.
- We proposed an innovative flexible loss function called scale-balanced loss for object detection tasks to alleviate the matching imbalance. The proposed scale-balanced loss along with prevalent anchor-based methods achieved excellent generalization performance as compared to other prevalent models without the proposed loss.
- We explored the impact of scale-balanced loss on detecting objects with different scales. The recall rates for small and medium objects attained up to 15% improvements on both PASCAL VOC and MS COCO datasets.

2. Related work

Despite some major improvements in the object detection, detecting multi-scale objects, especially small objects is still a challenge for existing detectors. This is mainly due to the peculiarities of small objects including less information about themselves, larger probabilities to confuse with the background, higher requirements for localization [25], etc. For the recognition task, the 32×32 pixel is the minimal size for color images within the allowable range. Torralba et al. [36] In detection benchmark COCO, small objects refer to those occupying areas less than or equal to 32×32 pixels. For these small objects, researchers have made much effort in this area, which can be summarized from four aspects: (i) building detectors for images of different scales; (ii) using shallow networks directly for detection; (iii) combining context information with coarse features; (iv) getting super-resolution with GAN.

2.1. Building detectors for images of different scales

Considering that images have objects of different scales, some simple but effective methods are introduced to construct detectors for images of different sizes [37–39]. In paper [40], an image pyramid is designed for input, and results from different scales are integrated for output. It is effective for face targets, especially small face targets which are easier to detect in large-scale images. YOLO9000 [28] makes a multi-scale training strategy that allows different sizes of input as a data argument method. However, the applications of these approaches are still limited due to their poor ability to extract complex features. In the general object detection tasks, the characteristics of objects and their relationships are very complex, and even a small change can produce a huge difference. Therefore, simple scaling is unable to obtain suitable feature representation for detection.

2.2. Using shallow networks directly for detection

For convolutional neural networks, nodes have large receptive fields in the deep feature map of the network. It is beneficial to detecting large objects while leading to more information loss for detecting small objects. The nodes in shallow layers have smaller corresponding receptive fields, which is more suitable for detecting small objects [41]. SSD [20] and MSCNN [39] make predictions with feature maps in different layers separately, and then integrate all prediction results. Hypernet [42] takes a different way that multi-layer feature maps are resized to the same scale by upsampling or downsampling for detection. This method is widely used in the following researches [12,19,20,43]. Since shallow networks have a weak expression ability and are not enough to cope with complex scenes, adding extra features is one way to improve performance.

2.3. Combining context information with coarse features

Since the small object itself has fewer features, it is an effective means to use context information to assist judgment [24,44,45]. Experimental results show that for face detection, context information around the face can significantly improve the accuracy of classification and positioning by human observers, especially for small faces [40,43]. FPN [33] and RefineDet [32] build feature pyramids by a top-down module [46], which can generate more appropriate feature representation with context information than skip connection. DSSD [35] makes deconvolutions on coarse features and integrates them with fine-grained features for predictions. MDFN [47] makes use of the relationships of individual objects and local contexts. Besides the contextual information around objects, the relationship between multiple objects is also considered critical for judgment. Based on the proposal regions, RNN can also integrate the scene information and object information to adjust the prediction results [48,49]. Such approaches make full use of context information and do not depend on CNN's receptive field, which compensates for the defects of the fully convolutional network to some extent.

2.4. Getting super-resolution with GAN

As proposed by PGAN [50], it is a novel way to use GAN to increase the resolution of small objects for detection. The generator learns to enhance the limited representations of small objects to super-resolved ones that are similar to real large objects to compete with a discriminator. This method is also adopted by Bai et al. [51] for the face detection task.

3. Our proposed method

In this section, we introduce our proposed method in detail. First, we show the matching imbalance in existing anchor matching strategies from different aspects. Second, a weighted loss function called scale-balanced loss is proposed for solving the imbalance. At last, we evaluate the impact of this imbalance on different pipelines.

3.1. The imbalance during anchors matching process

Currently, most of the state-of-the-art detection systems employ anchors in their methods, which play a key role in the head of pipelines. The anchors are designed as a set of default reference boxes with various sizes and aspect ratios to match the ground truth for a smaller searching space. Since the anchor gives proper prior knowledge for the network to determine which features are used to predict objects, the anchor-based methods far outperform anchor-free ones. As a well-known anchor-based method, RPN [14] makes use of anchors at each sliding window location for classification and localization. The anchor which has an obvious overlap with any ground truth will be regarded as a positive sample. Otherwise, it is ignored or considered as a negative one. However, this strategy is unfair for small objects as they naturally have smaller overlap with anchors. To make them not ignored, the anchors which have the highest IoU with small objects are also regarded as positive samples [14,20]. But it brings another problem that small objects may match fewer anchors than large objects. The matching process is shown in Fig. 2. The player matches more anchors than the ball. Since the matching strategy is only sensitive to objects' sizes, we can assume that the large objects match more anchors than small ones in most cases.

To give a fair comparison, we resize all images in the VOC2007 trainval set to scale 300. According to their size, these objects are divided into three types, just like what COCO [18] does on them. As

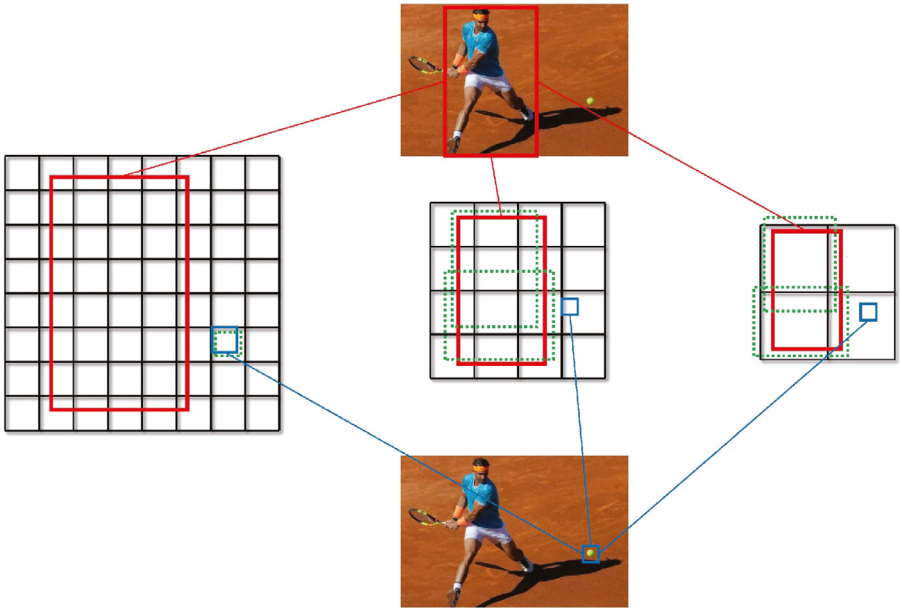

Fig. 2. The matching process performed on multiple feature maps is shown. The solid line boxes represent the objects and the dotted line boxes represent the matched anchors. The player can match more anchors in deep networks than the ball. It causes that the information of the ball cannot be paid enough attention to when the model tries to detect it.

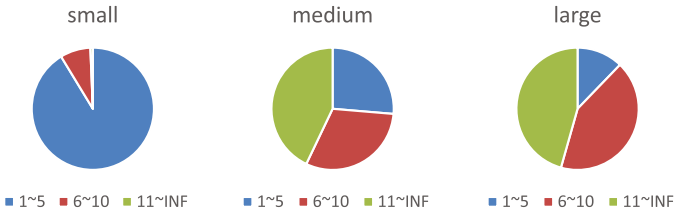

Fig. 3. The summary of matching times for objects of different sizes in SSD [20] is shown. The object in the blue part can only match anchors 1–5 times. For small objects, most of them belong to the blue part. When compared with medium and large objects, this unreasonable proportion does harm to the performance of detection. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

is shown in Fig. 3, after the matching process in SSD [20], most of the small objects can only match anchors 0–5 times and about 1/8 of large objects match less than 6 anchors. It is not in line with the original intention of designs for anchor matching strategies, which should give fair treatment on different sizes of objects.

As we all know, the small object is the hardest part of object detection tasks, and the matching imbalance mentioned above aggravates this problem. To alleviate this imbalance, a balanced anchor matching strategy is needed. However, due to the inflexible fully convolutional network framework in the existing detection pipelines, it seems to be difficult to design a proper strategy that can be applied to different designs. Our goal is to find out a practical method that is compatible with most existing anchor-based methods. Inspired by the focal loss [27] which is proposed to alleviate the extreme imbalance between foreground and background classes during training, we choose the loss function to make a breakthrough.

The focal loss is a dynamical scaled cross entropy loss and mainly relies on confidence, where the scaling factor decays to zero as confidence of the correct class increases. It does not work for anchor matching imbalance, because the imbalance among foreground classes is not so obvious compared to the imbalance between foreground and background [52]. So, the scaling factor in the focal loss has little difference in foreground classes and it cannot maintain the balance of anchor matching. Besides, the unbalanced anchor matching strategy leads to more outliers for large objects. In the focal loss, the confidence of outliers is small and will be given larger weights, which may reduce the stability of the model [53]. So, the confidence independent weighting method as this paper proposed is a more reliable solution.

3.2. Scale-balanced loss

For objects matching more than one anchor, all these anchors are expected to predict the same object correctly in the training stage. As all these anchors are treated equally by the training strategy in existing methods, the objects which match few anchors are more likely to be ignored. According to the statistics in Fig. 3, small objects match fewer anchors in existing pipelines. It makes small objects only have a slight impact on loss. Due to the key position of loss on deep learning, small objects have little effect on network parameters. As a result, the network cannot effectively extract the features of small objects.

If there is a compensation strategy for the anchor matching process, the impact of small objects on loss could be enhanced. In this condition, the network can extract features which are more meaningful for detecting small objects. At the same time, since the feature extraction of large objects is not a hard task, its detection accuracy will not decrease. The overall detection accuracy can be significantly improved. Based on the influence on the probabilities of successfully detecting objects, we design the scale-balanced loss to improve the previous training strategy. The proposed method re-weight the objects in the loss function according to the impact of the anchor matching process on the probability of detection.

Supposing that a ground truth G_i, it matches M_i anchors for a single input, the original total loss can be formulated as

$$loss = \sum_{i=1}^{N} \sum_{j=1}^{M_i} L(G_i, a2b_j(anchor_j)) \tag{1}$$

The $a2b$ represents the process of transforming an anchor into a bounding box based on the result of models, and L is the original loss function, which is designed to evaluate the difference between

the ground truth and the anchor. The total loss is the sum of all the $a2b$ matched anchor loss.

To reduce the negative effects caused by the anchor matching strategy on the prediction of objects, especially for multi-scale objects, we designed our scale-balanced loss as the following form.

$$loss^{scb} = \sum_{i=1}^{N} W_i \sum_{j=1}^{M_i} L(G_i, a2b_j(anchor_j)) \tag{2}$$

The difference between scale-balanced Loss and the original loss is that a weight related to the number of matched anchors is added for each object. W_i is used to balance the object prediction probability change of the anchor matching process, and objects that match more anchors have smaller W_i. For an object G_i, which matches M_i, anchors, its weight W_i, is the following form.

$$W_i = \frac{\beta(logM_i + \alpha)}{M_i} \tag{3}$$

There are two hyperparameters in the Eq. (3). The β here is to increase the loss of all positive samples. Without this parameter, the total loss of positive samples becomes too small, and negative samples will dominate the model training stage. The model may generate too many false-negative predictions without the β. The α here is a fixed weight to keep the balance between objects matched few anchors. Especially for the object which matches only one anchor, the α ensures their weights to be greater than zero.

By adding weights to the loss of different objects, we can alleviate the imbalance of positive samples in the anchor matching process. For negative samples, we take the existing sampling method(OHEM) to solve the imbalance problem and set their weights as 1. It ensures that the loss generated by positive samples matches with the one generated by nagetives samples proportionally.

In order to show the design purpose of the scale-balanced loss directly, we can transform W_i into a simpler form.

$$W_i = F(E_i, \alpha, \beta), E_i = \frac{logM_i}{M_i}, M_i \in N^* \tag{4}$$

In the scale-balanced loss, all its positive anchor losses are summed to generate the final loss for each object. As an object G_i matches M_i anchors, the $M_i * E_i$ can be considered as the approximate weight R_i of G_i in a training batch.

$$R_i = M_i * E_i = logM_i = -log(T_i), T_i = \frac{1}{M_i} \tag{5}$$

We can see from the above equation that the approximate weight of an object in scale-balanced loss is $logM_i$ instead of M_i. The objects matched plenty of anchors cannot dominate the loss in the training stage as before.

When we consider all the anchors match the same object as a cluster, every cluster should generate only one prediction. Otherwise, there will be a redundant false positive prediction. In this condition, for a single anchor in a cluster of size M_i, the probability to become the final prediction is $\frac{1}{M_i}$. Referring to the definition of information content in information theory, the physical meaning of R_i is the information content in the process of selecting an anchor as the final prediction among all the matched anchors.

It is not the first time to use information content as weights. For example, in the AdaBoost SAMME algorithm, in the process of using multiple weak classifiers to construct a strong classifier, the weight of the weak classifier also conforms to the definition of information content.

For a weak classifier C_k, its weight W_k in AdaBoost SAMME algorithm [54]:

$$W_k = \frac{1}{2}\log\left(\frac{1-e_k}{e_k}\right) + \log(R-1) \tag{6}$$

e_k is the error rate of the classifier, and R is the number of categories($R > 1$). As C_k should have a better performance than random classification, we can transform Eq. (6) into the following form:

$$W_k = \frac{1}{2}(-\log(T_k)) \tag{7}$$

$$T_k = \frac{e_k}{(1-e_k)A(R-1)^2}, e_k \in \left(0, \frac{R-1}{R}\right), T_k \in \left(0, \frac{1}{R-1}\right) \tag{8}$$

According to the range of e_k, we find that T_k can be expressed as a probability, and the weight of C_k is proportional to the logarithm of a probability, which is called information content in information theory.

Comparing the formulas (5) with (7), we can find that both of them have the same characteristics and roles of the information content, and the design of our scale-balanced loss is inspired by such laws.

Existing loss functions in object detection methods do not take into account the imbalance caused by the anchor matching process. The detection result of multi-scale objects especially small objects may become worse. Our proposed scale-balanced loss alleviates this imbalance by adding anchor matching process information to the loss function. When there are plenty of multi-scale objects in a single image, the scale-balanced loss not only can detect more small objects but also can produce less false positives.

3.3. Class imbalance with multi-stage detectors

Existing multi-stage methods mainly focus on the imbalance between the positive samples and negative samples and ignore the imbalance among the foreground classes. For those multi-stage detectors, more negative samples are generated by the anchor matching strategy during training time and most of them can be filtered out through the hard example mining strategy, etc. It is helpless for the matching imbalance mentioned in this paper because the positive samples are kept as many as possible in the hard example mining process. The imbalance among positive samples of different foreground classes remains unchanged. For multi-stage detectors, our proposed scale-balanced loss has an effect on different stages.

4. Experiment

The detectors chosen for comparison should meet the following 3 requirements: (1) The detector is representative and typical. (2) The anchor matching imbalance introduced above is obvious in this detector. (3) It has no additional operation to deal with the imbalance directly, which is not suitable for a fair comparison. Following these intuitions, we choose four popular detectors as our baselines. RefineDet [32] is a two-stage detector. SSD [20], FSSD [19] and DSSD [35] are one-stage detectors. All of them use anchors for matching and select similar matching strategies, which allows us to take a similar approach to them. We conduct experiments on Pascal VOC [17] and MS COCO [18] datasets, which have 20 and 80 object categories, respectively. In VOC 2007, a predicted bounding box that has IoU with the ground truth higher than 0.5 is considered as positives. In MS COCO, following the standard ways, different thresholds are used to get comprehensive results. The metric to evaluate detection performance is the Mean Average Precision(mAP). For a fair comparison, we follow all the training settings of the original experiments in the baselines [19,20,32] except for the loss function. All of our experiments are based on the PyTorch[2] version of model implementation. Code is available at: https://github.com/1243France/SCB_Loss

[2] https://pytorch.org

Table 1
Ablation study on SSD300 with PASCAL VOC. Different ways of adding weighting are performed. In order to be intuitive, some details are omitted and only the basic form is retained.

	VOC2007test
1	77.2
$1/(logM+1)$	77.8
$1/sqrtM$	78.1
$1/M$	76.8
(log M + 1)/M	**78.2**

4.1. Comprehensive results on PASCAL VOC

4.1.1. Ablation study on PASCAL VOC

VOC has 20 categories. VOC2007 dataset consists of about 5k trainval images and 5k test images. VOC2012 dataset includes about 11k trainval images and 11k test images. In our experiment, we trained models on the union of the 2007 trainval set and the 2012 trainval set. After applying the scale-balanced loss function, all of them achieve a better mAP. Furthermore, in order to illustrate that our proposed method can especially improve the performance of small object detection, we divide all objects in the VOC2007 test set into 3 types according to their scales and calculate the recall for them separately as a judgment.

Different weighting formulations Based on the class imbalance problem proposed in this article, it is intuitive to increase the weight of the objects with fewer matches. We try different weighting strategies to get a more comprehensive analysis. Since we hope to focus on the objects with fewer matches, the weights have a negative correlation with the number of matches which we represent with M. Following this idea, 4 ways are shown in Table 1, in which we select SSD300 to be the baseline because it has a more obvious class imbalance. We can observe from Table 1 that the method based on the weight relating to information content gives better performance than others.

Weighting strategies for mutlit-stage detectors The two-stage detectors will perform the matching process twice. Taking Faster-RCNN [14] as an example, all through the RPN process matches the target with the anchor and the following detectors make matching again. The variation of information content generated by such a process is difficult to estimate. It may make it easier for the model to fit the target but harder for us to understand. In our baseline RefineDet320 [32], the ARM and ODM take a similar strategy to refine their results. We can take a separate weighting approach to the two parts. After applying scale-balanced loss on ARM, the results are improved from 80.0 to 80.4 mAP. But when the ODM is applied with the same strategy, the result drops to 78.0, which is much worse than the original one. The matching process in ARM is different from the one in ODM. It causes that the scale-balanced loss cannot work well in ODM like in others.

4.1.2. mAP on PASCAL VOC

According to the ablation study in Section 4.1, we select SSD300, FSSD300, and RefineDet320 as our baselines. The weighting method is determined as proposed in Section 3. We use VOC 2007 trainval and VOC2012 trainval to train models following the original implements. We retain all settings of the original implementations except for the loss function. Our results on VOC2007 test set are shown in Table 2. The scale-balanced loss can improve mAP by about 1.0 points compared to our baselines, which is shown in bold font.

Table 2
Performance of scale-balanced loss with different detection pipelines on VOC2007 test set. All methods are trained on VOC 07 + 12 trainval.

	α	β	VOC2007test	Improvement
SSD300	–	–	77.2	–
FSSD300	–	–	78.8	–
RefineDet320	–	–	80.0	–
SSD300 w/ scb loss	**1**	**3**	**78.2**	**1.0**
FSSD300 w/ scb loss	**1**	**3**	**79.5**	**0.7**
RefineDet320 w/ scb loss(ARM)	**1**	**3**	**80.4**	**0.4**

4.1.3. Recall rates of different sized objects on PASCAL VOC

As MS COCO did, we divide the VOC2007 test set into 3 types, small, medium, and large according to their scales. The average precision and average recall are used in COCO. However, due to the limited number of small objects which is 516 in PASCAL VOC, such evaluation criteria are prone to fluctuations. In order to obtain a convincing evaluation result, we make a summary of the recall rate for models performing at inference time. By using different confidence thresholds such as 0.01, 0.1, and 0.3, most boxes can be filtered out. Then the NMS is applied with a Jaccard overlap of 0.45 per class and keeps the top 200 detections per image. The bounding box which have a 0.5 or higher IoU with any ground truth is considered as positives. Tables 3 and 4 show the results of this part. The improvements are shown in bold font. We can observe that the scale-balanced loss has a significant improvement on SSD, especially for small objects. The recall of small objects makes a big jump on all three confidence thresholds. When taking 0.3 as the confidence threshold, the recall rate of SSD300 on small objects is improved from 17.44% to 42.83%. Our proposed scale-balanced loss makes the original model from almost useless to available on small objects. The results on FSSD [19] also prove that our proposed method can effectively improve the ability to detect small objects.

4.2. Comprehensive results on MS COCO

4.2.1. AP and AR on MS COCO

MSCOCO dataset has 80 object categories. We use the COCO Challenge 2017 data split to prepare our dataset. The training is based on the trainval35k and we test on test-dev set about 20k images. The test results are shown in Table 5. The original SSD300 gets 25.1% on the test set. After applying the scale-balanced loss, it achieves 26.6% AP. It gets a more obvious improvement than on Pascal VOC. The chief reason for this result is that COCO has more small objects. The advanced evaluation further verifies our thoughts. The COCO dataset divides objects into 3 types according to their sizes. After applying our proposed scale-balanced loss, the performance on small objects gets much better than baselines. There is also a slight decline in the performance in detecting large objects. Experiments on FSSD and DSSD have similar conclusions with SSD, which verifies the effectiveness of the scale-balanced loss. Due to the imbalance of the anchor matching strategy, these models do not pay enough attention to small objects in the existing methods.

4.2.2. Recall rates of different sized objects on MS COCO

Although there are special evaluations of small objects in COCO, we create a summary of the recall rate for models performing at inference time for a fair comparison. All the models are trained with the COCO trainval35k dataset. As we do on PASCAL VOC, filtering with different thresholds and NMS are performed before calculating recall rates. The results in Tables 6–9 prove that our proposed method works well for small objects too. Significant improvements can be seen when we use a higher threshold. It is worth noting that using larger input images, models do not achieve

Table 3
The recall rate of SSD300 on VOC 2007 test set. The results which confidence are lower than the threshold is discarded for a better comparison.

Method & threshold	recall_small	recall_medium	recall_large
SSD300, 0.01	73.06(377/516)	92.29(3519/3813)	96.57(7439/7703)
SSD300 w/ scb loss, 0.01	**80.03(413/516)**	**92.76(3537/3813)**	**96.25(7414/7703)**
SSD300, 0.1	43.02(222/516)	78.94(3010/3813)	91.89(7078/7703)
SSD300 w/ scb loss, 0.1	**61.43(317/516)**	**83.11(3169/3813)**	**91.91(7080/7703)**
SSD300, 0.3	17.44(90/516)	62.65(2389/3813)	87.90(6771/7703)
SSD300 w/ scb loss, 0.3	**42.83(221/516)**	**69.81(2662/3813)**	**88.21(6795/7703)**

Table 4
The recall rate of FSSD300 on VOC 2007 test set.

Method & threshold	recall_small	recall_medium	recall_large
FSSD300, 0.01	69.57(359/516)	89.17(3400/3813)	95.55(7360/7703)
FSSD300 w/ scb loss, 0.01	**78.88(407/516)**	**90.24(3441/3813)**	**95.82(7381/7703)**
FSSD300, 0.1	41.08(212/516)	77.39(2951/3813)	90.83(6997/7703)
FSSD300 w/ scb loss, 0.1	**60.27(311/516)**	**79.67(3038/3813)**	**91.30(7033/7703)**
FSSD300, 0.3	18.99(98/516)	67.26(2565/3813)	87.74(6759/7703)
FSSD300 w/ scb loss, 0.3	**40.69(210/516)**	**71.02(2708/3813)**	**88.20(6794/7703)**

Table 5
Results on MS COCO test-dev 2015.

	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L	AR_1	AR_{10}	AR_{100}	AR_S	AR_M	AR_L
SSD300	25.1	43.1	25.8	6.6	25.9	41.4	23.7	35.1	37.2	11.2	40.4	58.4
SSD300 w/ scb loss	**26.6**	**45.9**	**27.5**	**8.3**	**28.1**	**40.8**	**24.4**	**36.3**	**38.1**	**13.7**	**41.5**	**56.9**
FSSD300	27.1	47.7	27.8	8.7	29.2	42.2	24.6	37.4	40.0	15.9	44.2	58.6
FSSD300 w/ scb loss	**29.3**	**50.5**	**30.1**	**11.5**	**31.1**	**42.5**	**25.9**	**39.5**	**41.8**	**19.2**	**45.9**	**57.9**
DSSD321	28.0	46.1	29.2	7.4	28.1	47.6	25.5	37.1	39.4	12.7	42.0	62.6
DSSD321 w/ scb loss	**29.5**	**49.1**	**31.2**	**10.5**	**30.0**	**47.6**	**26.3**	**39.7**	**41.0**	**16.2**	**43.1**	**61.9**
SSD512	28.8	48.5	30.3	10.9	31.8	43.5	26.1	39.5	42.0	16.5	46.6	60.8
SSD512 w/ scb loss	**30.4**	**50.6**	**32.0**	**13.0**	**34.5**	**42.3**	**26.6**	**40.6**	**42.6**	**17.7**	**48.6**	**58.6**
FSSD512	31.8	52.8	33.5	14.2	35.1	45.0	27.6	42.4	45.0	22.3	49.9	62.0
FSSD512 w/ scb loss	**33.3**	**56.4**	**34.9**	**17.9**	**35.2**	**43.8**	**28.2**	**44.0**	**46.6**	**27.7**	**49.7**	**59.6**
DSSD513	33.2	53.3	35.2	13.0	35.4	51.1	28.9	43.5	46.2	21.8	49.1	66.4
DSSD513 w/ scb loss	**34.4**	**57.0**	**36.1**	**16.5**	**35.7**	**49.9**	**29.4**	**44.5**	**47.3**	**26.6**	**49.2**	**64.7**

Table 6
The recall rate of SSD300 on COCO minival5k.

Method & threshold	recall_small	recall_medium	recall_large
SSD300, 0.01	38.93	83.4	95.2
SSD300 w/ scb loss, 0.01	**46.42**	**84.66**	**95.17**
SSD300, 0.1	27.32	71.75	88.9
SSD300 w/ scb loss, 0.1	**41.39**	**79.71**	**91.09**
SSD300, 0.3	8.4	45.27	78.34
SSD300 w/ scb loss, 0.3	**22.52**	**58.91**	**81.5**

Table 7
The recall rate of SSD512 on COCO minival5k.

Method & threshold	recall_small	recall_medium	recall_large
SSD512, 0.01	47.3	88.47	95.55
SSD512 w/ scb loss, 0.01	**56.25**	**89.17**	**95.36**
SSD512, 0.1	34.13	79.06	90.36
SSD512 w/ scb loss, 0.1	**47.5**	**82.64**	**90.35**
SSD512, 0.3	11.65	54.54	81.3
SSD512 w/ scb loss, 0.3	**22.68**	**61.12**	**79.51**

Table 8
The recall rate of FSSD300 on COCO minival5k.

Method & threshold	recall_small	recall_medium	recall_large
FSSD300, 0.01	46.52	85.85	94.59
FSSD300 w/ scb loss, 0.01	**51.69**	**86.67**	**95.02**
FSSD300, 0.1	35.26	75.26	88.05
FSSD300 w/ scb loss, 0.1	**47.47**	**81.36**	**90.44**
FSSD300, 0.3	14.14	52.42	78.42
FSSD300 w/ scb loss, 0.3	**30.95**	**65.03**	**82.07**

Table 9
The recall rate of FSSD512 on COCO minival5k.

Method & threshold	recall_small	recall_medium	recall_large
FSSD512, 0.01	62.05	88.22	95.36
FSSD512 w/ scb loss, 0.01	**66.20**	**88.76**	**95.31**
FSSD512, 0.1	49.55	79.01	90.49
FSSD512 w/ scb loss, 0.1	**58.16**	**81.6**	**89.82**
FSSD512, 0.3	23.44	58.18	82.12
FSSD512 w/ scb loss, 0.3	**35.77**	**65.1**	**81.84**

better results for large objects. It can be considered that in the case where the localization information is not required to be high, increasing the input size has little effect on detecting large objects. Another difference from results on VOC is that the recall rate of FSSD is higher than that of SSD. Due to the higher complexity of COCO, FSSD can produce more reasonable features and use them to generate better results.

4.2.3. Visualization of performance improvement on MS COCO

Our proposed method performs better on small objects. For example, as illustrated in Fig. 4 column 1, the original SSD model cannot detect the bird on the bench but it detects successfully with the scale-balanced loss. In addition, as shown in Fig. 4 column 1 to column 4, SSD512 with scale-balanced loss detects small objects more accurately.

For FSSD512, our proposed scale-balanced loss also works well. As shown in Fig. 5, the performance of detecting small objects has been significantly improved. The optimized model can not only de-

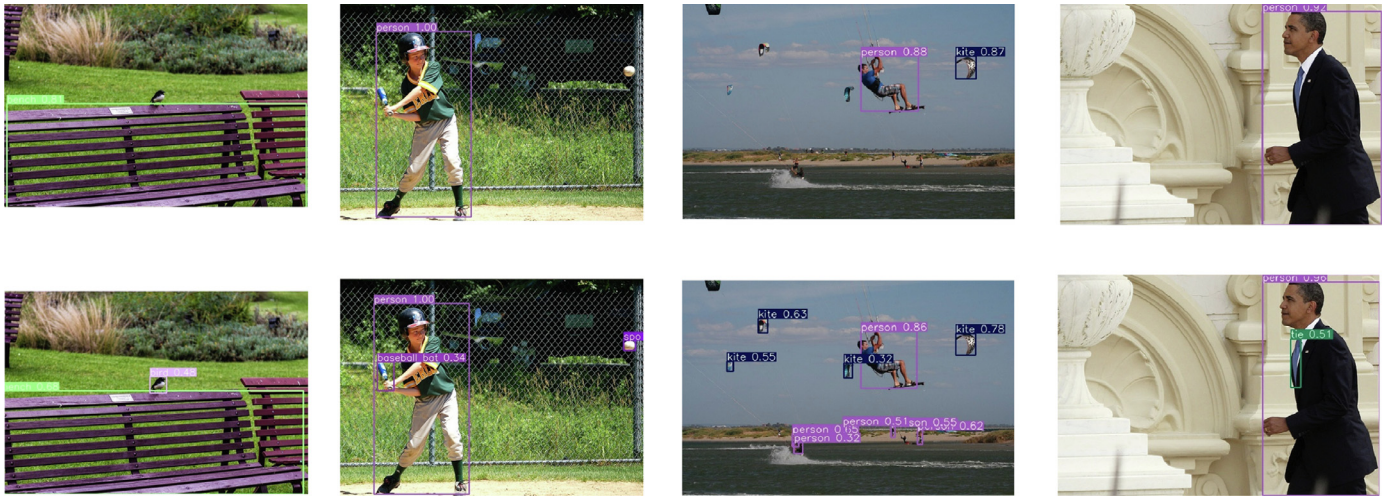

Fig. 4. SSD512 vs SSD512 w/ scb loss. Both models are trained with COCO trainval35k dataset. The top row contains the results from the conventional SSD512 and the bottom row is from SSD512 with scale-balanced loss. Bounding boxes with score of 0.3 or higher is drawn. Better viewed on screen.

Fig. 5. FSSD512 vs FSSD512 w/ scb loss. The top row contains the results from the conventional FSSD512 and the bottom row is from FSSD512 with scale-balanced loss. Settings in the inference time are same as in SSD512 above.

tect more small objects but also reduce false positives. This is significant for small object detection in practice.

4.3. Speed

The only part of the model we modified is the loss function. In the training process, the model needs to calculate the anchor matching times for each object. It is negligible when compared to the computation cost of the network. In the inference process, it does not cost any additional time in the whole process. Compared to the original SSD300, our proposed method has the same FPS[3] in our experiment(about 50 FPS with two 1080ti). Actually, the scale-balanced loss tries to make up the difference between the distribution of generated anchors and training data. It will not affect the speed of the original models.

5. Conclusion and future work

In this paper, we proposed the scale-balanced loss, which handles the class imbalance during the matching process in existing detection pipelines. As we all know, objects with small scales are harder to be detected correctly than others. Moreover, in existing detection pipelines, the unbalanced object matching strategies make them more difficult to be noticed. The scale-balanced loss manages to make each object's proportion in the loss function proportional to the information content generated during the matching process. Experiments on PASCAL VOC and MS COCO have proved that several popular methods make convincing improvements after applying the scale-balanced loss. The recall rates of small and medium objects get up to 15% improvements in both PASCAL VOC and MS COCO datasets, and it adds no computational cost in inference time for original models, which is meaningful to apply the proposed loss in practice.

In the future, some more complex models can be integrated with the scale-balanced loss for improvements, and it is worth studying a more suitable form of loss for multi-stage detectors. The performance of it can also be expected in other areas, such as face detection, pedestrian detection, and object tracking.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

[3] Frames per second.

Acknowledgments

The authors would like to thank the anonymous reviewers for the constructive comments. This work was supported in part by the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (Grant no. 61921003) and the open fund project of Science and Technology on Communication Networks Laboratory (Grant no. HHX21641X003).

References

[1] J. Xiang, G. Zhu, Joint face detection and facial expression recognition with MTCNN, in: 4th International Conference on Information Science and Control Engineering (ICISCE), 2017, pp. 424–427.
[2] W. Wu, D. Tao, H. Li, Z. Yang, J. Cheng, Deep features for person re-identification on metric learning, Pattern Recognit. 110 (2020) 107424.
[3] Y. Chen, D. Zhao, L. Lv, Q. Zhang, Multi-task learning for dangerous object detection in autonomous driving, Inf. Sci. 432 (2018) 559–571.
[4] Q. Dou, H. Chen, Y. Jin, H. Lin, J. Qin, P.-A. Heng, Automated pulmonary nodule detection via 3D ConvNets with online sample filtering and hybrid-loss residual learning, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2017, pp. 630–638.
[5] S. Ren, K. He, R. Girshick, X. Zhang, J. Sun, Object detection networks on convolutional feature maps, IEEE Trans. Pattern Anal. Mach. Intell. 39 (7) (2017) 1476–1481.
[6] 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.
[7] H. Rezatofighi, N. Tsoi, J. Gwak, A. Sadeghian, I. Reid, S. Savarese, Generalized intersection over union: a metric and a loss for bounding box regression, in: IEEE Conference on Computer Vision and Pattern Recognition 2019, IEEE, Institute of Electrical and Electronics Engineers, 2019, pp. 658–666.
[8] B. Jiang, R. Luo, J. Mao, T. Xiao, Y. Jiang, Acquisition of localization confidence for accurate object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 784–799.
[9] 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.
[10] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, Y. LeCun, Overfeat: integrated recognition, localization and detection using convolutional networks, in: 2nd International Conference on Learning Representations, ICLR, 2014.
[11] C. Szegedy, A. Toshev, D. Erhan, Deep neural networks for object detection, NIPS, 2013.
[12] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, J. Sun, DetNet: design backbone for object detection, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 334–350.
[13] J. Dai, Y. Li, K. He, J. Sun, R-FCN: object detection via region-based fully convolutional networks, NIPS, 2016.
[14] S. Ren, K. He, R. Girshick, J. Sun, Faster R-CNN: towards real-time object detection with region proposal networks, in: Advances in Neural Information Processing Systems, 2015, pp. 91–99.
[15] K. He, G. Gkioxari, P. Dollár, R. Girshick, Mask R-CNN, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 2961–2969.
[16] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.
[17] M. Everingham, A. Zisserman, C.K. Williams, L. Van Gool, M. Allan, C.M. Bishop, O. Chapelle, N. Dalal, T. Deselaers, G. Dorkó, et al., The 2005 pascal visual object classes challenge, in: Machine Learning Challenges Workshop, Springer, 2005, pp. 117–176.
[18] 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: European Conference on Computer Vision, Springer, 2014, pp. 740–755.
[19] Z. Li, F. Zhou, FSSD: feature fusion single shot multibox detector, arXiv preprint arXiv:1712.00960(2017).
[20] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, A.C. Berg, SSD: single shot multibox detector, in: European Conference on Computer Vision, Springer, 2016, pp. 21–37.
[21] M. Najibi, M. Rastegari, L.S. Davis, G-CNN: an iterative grid based object detector, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 2369–2377.
[22] Z. Tian, C. Shen, H. Chen, T. He, FCOS: fully convolutional one-stage object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9627–9636.
[23] 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.
[24] 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 the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 5936–5944.
[25] S. Agarwal, J.O.D. Terrail, F. Jurie, Recent advances in object detection in the age of deep convolutional neural networks, arXiv preprint arXiv:1809.03193(2018).
[26] H. Law, J. Deng, CornerNet: detecting objects as paired keypoints, in: Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 734–750.
[27] 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.
[28] A. Farhadi, J. Redmon, YOLO9000: better, faster, stronger, in: 2017 IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 6517–6525.
[29] S. Zhang, X. Zhu, Z. Lei, H. Shi, X. Wang, S.Z. Li, Faceboxes: a cpu real-time face detector with high accuracy, in: 2017 IEEE International Joint Conference on Biometrics (IJCB), IEEE, 2017, pp. 1–9.
[30] T. Yang, X. Zhang, Z. Li, W. Zhang, J. Sun, MetaAnchor: Learning to detect objects with customized anchors, in: Advances in Neural Information Processing Systems, 2018, pp. 320–330.
[31] 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.
[32] S. Zhang, L. Wen, X. Bian, Z. Lei, S.Z. Li, Single-shot refinement neural network for object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4203–4212.
[33] 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, 2017, pp. 2117–2125.
[34] D. Tao, J. Cheng, Z. Yu, K. Yue, L. Wang, Domain-weighted majority voting for crowdsourcing, IEEE Trans. Neural Netw. Learn. Syst. 30 (1) (2018) 163–174.
[35] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, A.C. Berg, DSSD: deconvolutional single shot detector, arXiv preprint arXiv:1701.06659(2017).
[36] A. Torralba, R. Fergus, W.T. Freeman, 80 million tiny images: a large data set for nonparametric object and scene recognition, IEEE Trans. Pattern Anal. Mach. Intell. 30 (11) (2008) 1958–1970.
[37] P.F. Felzenszwalb, R.B. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part-based models, IEEE Trans. Pattern Anal. Mach. Intell. 32 (9) (2010) 1627–1645.
[38] B. Singh, L.S. Davis, An analysis of scale invariance in object detection snip, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 3578–3587.
[39] Z. Cai, Q. Fan, R.S. Feris, N. Vasconcelos, A unified multi-scale deep convolutional neural network for fast object detection, in: European Conference on Computer Vision, Springer, 2016, pp. 354–370.
[40] P. Hu, D. Ramanan, Finding tiny faces, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 951–959.
[41] Y. Li, Y. Chen, N. Wang, Z. Zhang, Scale-aware trident networks for object detection, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 6054–6063.
[42] T. Kong, A. Yao, Y. Chen, F. Sun, Hypernet: towards accurate region proposal generation and joint object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 845–853.
[43] Y. Bai, B. Ghanem, Multi-branch fully convolutional network for face detection, arXiv preprint arXiv:1707.06330(2017).
[44] Y. Zhu, C. Zhao, J. Wang, X. Zhao, Y. Wu, H. Lu, CoupleNet: coupling global structure with local parts for object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4126–4134.
[45] J. Yuan, H.-C. Xiong, Y. Xiao, W. Guan, M. Wang, R. Hong, Z.-Y. Li, Gated CNN: integrating multi-scale feature layers for object detection, Pattern Recognition 105 (2019) 107131.
[46] A. Shrivastava, R. Sukthankar, J. Malik, A. Gupta, Beyond skip connections: top-down modulation for object detection, arXiv preprint arXiv:1612.06851(2016).
[47] W. Ma, Y. Wu, F. Cen, G. Wang, MDFN: multi-scale deep feature learning network for object detection, Pattern Recognition 100 (2020) 107149.
[48] 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.
[49] X. Chen, A. Gupta, Spatial memory for context reasoning in object detection, in: Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 4086–4096.
[50] J. Li, X. Liang, Y. Wei, T. Xu, J. Feng, S. Yan, Perceptual generative adversarial networks for small object detection, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 1222–1230.
[51] Y. Bai, Y. Zhang, M. Ding, B. Ghanem, Finding tiny faces in the wild with generative adversarial network, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 21–30.
[52] K. Oksuz, B.C. Cam, S. Kalkan, E. Akbas, Imbalance problems in object detection: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020) 1, doi:10.1109/TPAMI.2020.2981890.
[53] B. Li, Y. Liu, X. Wang, Gradient harmonized single-stage detector, in: Proceedings of the AAAI Conference on Artificial Intelligence, 33, 2019, pp. 8577–8584.
[54] J. Friedman, T. Hastie, R. Tibshirani, et al., Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors), Ann. Stat. 28 (2) (2000) 337–407.