The Visual Computer (2021) 37:245–259
https://doi.org/10.1007/s00371-020-01796-7

ORIGINAL ARTICLE

A multi-phase blending method with incremental intensity for training detection networks

Quan Quan[1] · Fazhi He[1] · Haoran Li[1]

Published online: 27 January 2020
© Springer-Verlag GmbH Germany, part of Springer Nature 2020

Abstract
Object detection is an important topic for visual data processing in the visual computing area. Although a number of approaches have been studied, it still remains a challenge. There is a suitable way to promote image classifiers by blending training with blended images and corresponding blended labels. However, our experiments show that directly moving existing blending methods from classification to object detection will cause the training process become harder and eventually will lead to a bad performance. Inspired by our discovery, this paper presents a multi-phase blending method with incremental blending intensity to improve the accuracy of object detectors and achieve remarkable improvements. Firstly, to adapt blending method to detection task, we propose a smoothly scheduled and incremental blending intensity to control the degree of multi-phase blending. Based on the above dynamic coefficient, we propose an incremental blending method, in which the blending intensity is smoothly increased from zero to full. Therefore, more complex and various data can be created to achieve the goal of regularizing the network. Secondly, we also design an incremental hybrid loss function to replace the original loss function. The blending intensity in our loss function increases smoothly, which is controlled by our scheduled coefficient. Thirdly, we further discard more negative examples in our multi-phase training process than other typical training methods and processes. By doing so, we can regularize the neural network to enhance generalization capability with data diversity and eventually to improve the accuracy in object detection. Another advantage is that there is no negative effect on evaluation because our method is just applied during the training process. Typical experiments show the proposed method improves the generalization of the detection networks. On PASCAL VOC and MS COCO, our method outperforms the state-of-the-art RFBNet of one-stage detectors for real-time processing.

Keywords Object detection · Data augmentation · Convolutional neural network

1 Introduction

Neural networks, especially deep neural networks, have fundamental advantages over traditional methods for visual computing [1–13]. For object detection task, since R-CNN [14] was proposed 4 years ago, the accuracy on VOC [15] dataset has gradually improved. Different from R-CNN and Fast R-CNN [16], Faster R-CNN is fully based on the convolutional network. Furthermore, the one-stage object detection approaches, such as SSD, combine two stages in Faster R-CNN to obtain the bounding boxes and the labels in the same output. Although the accuracy of one-stage detectors is a little lower than two-stage, it has the advantage of concise network architecture and high speed.

The above networks are used for many applications [17–21]. The typical network training rule is to train the networks by minimizing their average error over training data, which is known as the empirical risk minimization (ERM) principle [22]. The classical theory of machine learning tells us that the convergence of ERM can be guaranteed as long as the size of the learning machine does not increase with the number of training data [22,23].

However, a recent research [24] shows suspect opinion that ERM allows large neural networks to memorize (instead of generalizing from) the training data despite that the pre-

✉ Fazhi He
fzhe@whu.edu.cn

Quan Quan
quan_q@whu.edu.cn

Haoran Li
lhr@whu.edu.cn

[1] School of Computer Science and Technology, Wuhan University, Wuhan, Hubei, China

vious works conduct a lot of tricks, such as taking strong regularization and applying the random label for classification problem.

In many applications of neural networks of recent years [25,26], the performance can be easily impacted by the training and testing data. The neural networks being trained with ERM may give the opposite (error) predictions for the custom (testing) examples. Therefore, the generalization is still a challenge.

Typical data augmentation methods to address the above problems can be found in classification task [27] and can be formalized by the vicinal risk minimization (VRM) principle [28], which tries to train networks on similar but different examples. The basic methods include slightly image rotation, random crop, horizontal flip, mild scaling, etc. Other augmentation methods are noisy labeled data by adding noise to labels [29], label smoothing by softening the label from one-hot to no explicit ones and zeros in labels [30]. Blending methods try to blend the inputs and their targets across different classes [23,31,32] and achieve dramatic improvements in classification task.

However, the above data augmentation methods are oriented for classification task with the assumption that the examples in the vicinity share the same classes, and they are not suitable for being applied to the detection task directly.

For the classification task, the classifier only needs to produce a prediction for each image. However, for the detection task, the detector has to predict both locations and categories of all objects. So the complexity of detection is much higher than that of classification. Therefore, directly and simply moving above blending method from classification task to detection task will put more pressures on training and will make it difficult for the network to converge to the optimal state, eventually leading to performance degradation.

To solve the above problem, we present a multi-phase blending method to improve the accuracy of object detectors and achieve remarkable improvements.

Firstly, we propose a scheduled and incremental coefficient to control the blending intensity. We construct a sigmoid formulation to lead the multi-phase training process. (1) In the initial phase, the intensity starts from almost zero and increases slowly and smoothly. So the network has time to fit itself to the difficult object dataset and converge to a good state. (2) In the second phase, the intensity grows rapidly and reaches a high level of full intensity in a short time, so that the blending method starts to amplify the regularization effect on the detector. (3) In the last phase, the detector is trained with full intensity until the detection network converges. Based on the above dynamic coefficient, we propose an incremental blending method, in which the blending degree is controlled by this coefficient. In this way, more complex and various training data can be created to regularize the network. Meanwhile, the training process will not become too tough for the network.

Secondly, we also design a hybrid loss function with incremental intensity. The blending intensity of both increases smoothly at the beginning, which is controlled by our scheduled coefficient. Different from the original loss function, we propose a hybrid method for loss functions, in which the classification function and regression function can be blended separately.

Thirdly, the blending method will further increase the number of negative examples by creating hybrid categories with more backgrounds than objects, which generally belong to negative examples. For the detection task, too many negative examples have no advantage for detecting positive examples. On the contrary, they will make the training process more difficult. Therefore, we further discard more negative examples in our multi-phase training process than that in other typical training methods and processes.

Finally, our experiments indicate that we can achieve our purpose of regularizing the object detection networks and eventually improve the performance on complex detection tasks.

The proposed method is highly valuable for its improvement on the detector's performance without increasing its computational cost. The only price is more time spent in the training phase. Moreover, it is a compact and independent module that is easy to use.

The rest of this paper is organized as follows. Section 2 briefly reviews the related work in object detection. Section 3 presents our regularization method for one-stage object detectors. Section 4 conducts the experiments and discusses empirical results. Section 5 analyzes the highlights of the proposed network. Section 6 concludes this work and discusses future work.

2 Related work

2.1 Detection networks

Two-stage detector R-CNN [14] is a standard two-stage object detection framework. Girshick et al. [14] combines the steps of cropping box proposals like selective search [33] and classifying them through a CNN model, yielding a significant accuracy gain. For speeding up, Fast R-CNN [16] computes the entire image only once in a feature extractor and then puts it into a spatial pooling layer, called ROI pooling, thus allowing to reuse the features in classification.

Faster R-CNN [34] shows that the quality of object proposals can be optimized by deep neural networks and replaces the independent proposal generators in its predecessors by region proposal network (RPN). RPN has a set of boxes, named anchors, paved on the image at different locations,

scales, and aspect ratios, and it is trained to make class-agnostic predictions and regression predictions of offsets which fit the object locations for each anchor.

Faster R-CNN is later extended to many more advanced versions. A typical extension of it is Mask R-CNN [35], which uses a parallel branch to segment the object mask and presents a RoIAlign layer to fix misalignment to improve the detection accuracy.

One-stage detector The typical one-stage detectors are YOLO [36,37] and SSD[38]. YOLO predicts confidences and locations for multiple objects by using the whole feature map. YOLO runs very fast because of eliminating the stage of proposal generation. However, performance is limited. SSD [38] is another one-stage object detection approach and is widely used in pedestrian detection, car detection, and object tracking, etc. Different from two-stage detection, SSD produces the results of bounding boxes and class labels from the feature map at the same time through the location layer and classification layer, so this framework is faster than two-stage detector but less accurate.

RFBNet [39] improves basic SSD. It adds a module called Receptive Field Block (RFB), which consists of several convolutional kernels of different sizes in parallel. Compared with the inceptive block [30], RFB uses a different length of stride and a bigger kernel to ensure the feature map covered. So RFB block expands the receptive field of layers to have the ability to access more information.

Without special notation, our work is in the context of **one-stage** detection networks.

2.2 Data augmentation methods

Intuitive image operations Most existing data enhancement methods used in object detection are limited to the use of intuitive image operations (such as cropping, rotation, which are minor changes to the object). However, these operations do not obviously change the images.

Noisy label Learning with noisy labeled training data has been extensively studied in machine learning and computer vision literature. Limitations still exist. Experiments in [40] show that the classifiers inferred by label noise-robust algorithms are still affected by label noise. Many studies have shown that label noises can adversely impact the classification accuracy of induced classifiers [41]. Bartlett et al. [42] proved that most of the loss functions are not completely robust to label noise.

Label smoothing There exist several related label smoothing methods [23,30].

Szegedy et al. [30] tries to soften the label by adding additional labels of each class to enhance the regularization and get a small improvement. This method encourages the model to be less confident. It does regularize the model and makes it more adaptable by preventing the largest logit from becoming much larger than all others. Although it has a positive effect on generalization, this soft method is not explicit because label softening is random, and has little influence on some networks. By contrast to [30], we use the explicit image information to get the same effect of overfitting and avoid any wrong information.

Furthermore, [23,31,32] assume that the linear relationship between images and their labels also affect the generalization of models. They adopt another way to get the vicinity distribution: They mix the two original images by simply adding together with a random percentage, the label of each also needs to be added together with the same percentage, and thus the new images and labels are produced to train the neural networks.

Our work differs from the above literature [23,30–32] as follows: (1) It is aimed at object detection, including both regression problems and classification problems, while the above methods are only for classification problems. (2) In addition to one type of blended loss function for the labels, our method constructs two types of hybrid loss functions for both labels and locations, containing hybrid classification loss function and hybrid regression loss function. (3) In order to alleviate the difficulty of training the complex data caused by blending operations, we propose a scheduled and incremental blending parameter to smoothly control blending intensity and discard more negative examples.

2.3 Contribution

As a brief summary of this section, our contributions lie as follows: (1) We design a smooth, scheduled and incremental coefficient with mathematical sigmoid formulation to control the blending intensity among the multi-phase and propose a blending method based on dynamic and incremental intensity. (2) We propose two incremental hybrid loss functions containing hybrid classification loss function and hybrid regression loss function, in addition to the original loss function. (3) We further enhance the hard negative mining method by discarding more negative examples (Fig. 1).

3 The proposed method

3.1 The principle of the proposed method

Firstly, the widely used data augmentation methods of the intuitive image operations increase the number of true images that are stable and concise for training both classification network and object detection networks. Blending method creates data of blended class which is closer to one of the two classes. The blended data expand the training space, and

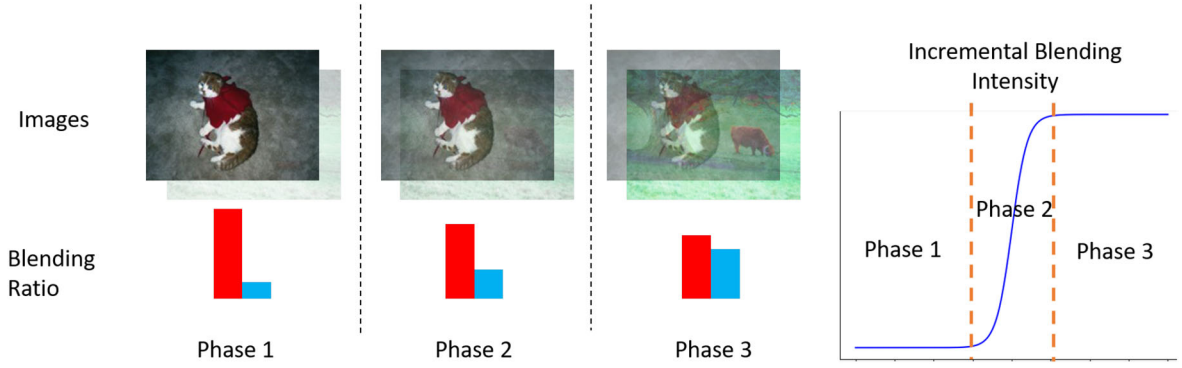


Fig. 1 The overview of multi-phase training. The blending intensity smoothly increases according to our scheduled blending intensity. In the initial phase, the intensity starts from almost zero and increases slowly and smoothly. In the second phase, the intensity grows quickly and reaches a high level of full intensity in a short time. In the last phase, the detector is trained with full intensity until the detection network converges

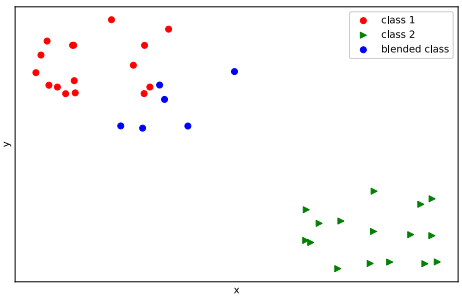


Fig. 2 The red dots are data of a class in the natural distribution, and the green dots are another class. Blended data are created in the vicinal space of the red dots, to expand the training space and make the feature space smoother

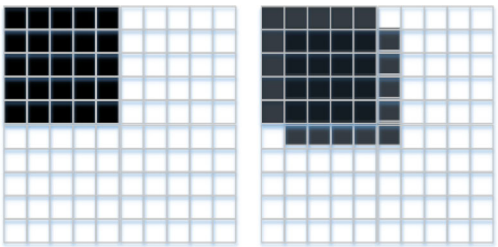


Fig. 3 Left is the original image (black box) and right is the blended image (blended black box)

the soft labels of blended data make the nearby feature space smoother (Fig. 2). However, this blending method creates inexact data, which is acceptable for classification network, but hard for the detection network. Therefore, we propose a new multi-phase method to smoothly control the blending intensity among the multi-phase, in which the network can adapt gradually.

Secondly, in predicting the position of bounding boxes, the coordinates of bounding boxes are continuous values. The softened labels is also continuous values, which match the object detection task very well. Therefore, we propose two incremental hybrid loss functions containing hybrid classification loss function and hybrid regression loss function, in addition to the original loss function.

The basic idea of the proposed method is illustrated in Fig. 1.

3.2 Gaps between classification and detection

Gaps always exist between classification and detection tasks. To initially test the performance of the blending method on regression problems, we conduct a fundamental experiment to show the effect for regression problem in object detection.

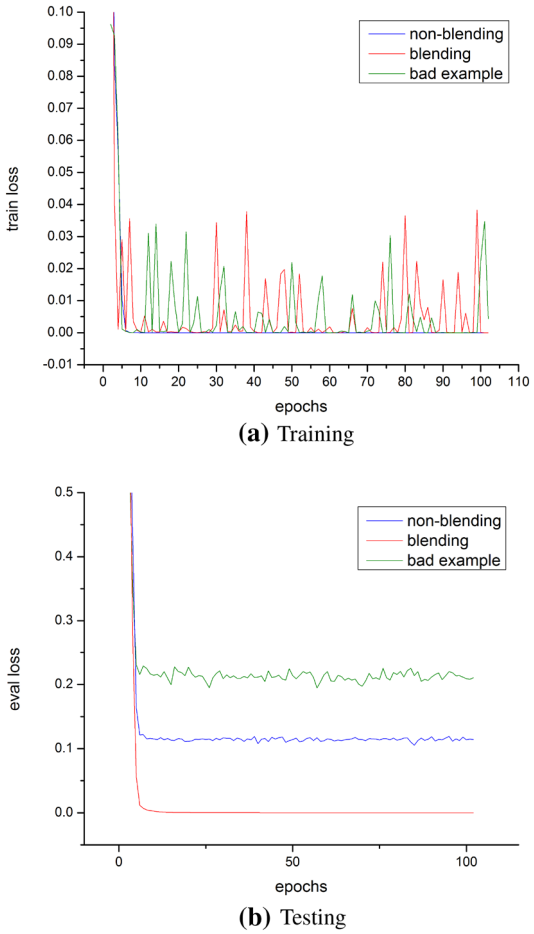
The experiment is set as follows. As shown in Fig. 3, we create a white 10×10 square box containing a 5×5 black box. We establish a data distribution from the original distribution to simulate the natural situation that the detection datasets (like PASCAL VOC, etc.) are sampled from the natural image data distribution. In this experiment, only 10 samples of 25 are selected as training data. In the test phase, we use all data to test the trained model.

For training data distribution $\mathcal{D} := (x_i, y_i)_{i=1}^m$ of location of the black block, it is a sample distribution from real distribution. We denote x_i as the image pixels and y_i as its values of location.

Firstly, we construct a new distribution $\mathcal{D}_v := (\tilde{x_i}, \tilde{y_i}, \tilde{z_i})_{i=1}^m$ from $\mathcal{D} := (x_i, y_i)_{i=1}^n$ for images by proposed blending operation.

$$\begin{cases} \tilde{x_i} = \lambda x_i + (1-\lambda)x_j \\ \tilde{y_i} = y_i \\ \tilde{z_i} = y_j \end{cases} \tag{1}$$

$\lambda \sim \text{Beta}(\beta, \beta)$ In our experiment, we set $\beta = 0.1$.



(a) Training

(b) Testing

Fig. 4 Graphs refer to the losses of models which are trained with blending method (red and green) and no blending (blue), where it should be noted that green is a bad example, **a** shows that blending method is more fluctuant in the training process. In **b**, the red line refers to the model performing better than baseline while the green one is the model performing worse than baseline model. It means that the final models with original blending method are inconsistent and not always good

Secondly, we detect the location of a black box with small-scale AlexNet for the initial test, in which we trained the network by the loss function l_{hybrid} (Eq. 4)

$$\text{loss}_p(\theta) = L_{\text{SM}}(f_\theta(\tilde{x_i}), y_i) \tag{2}$$

$$\text{loss}_q(\theta) = L_{\text{SM}}(f_\theta(\tilde{x_i}), z_i) \tag{3}$$

$$l_{\text{hybrid}}(\theta) = \lambda \text{loss}_p(f\theta) + (1-\lambda)\text{loss}_q(\theta) \tag{4}$$

where L_{SM} denotes Smooth L1 Loss, f_θ and θ are model and its weights.

As shown in Fig. 4, the experimental results show that there is potential for the blending method to improve the detection model, but the training process is unstable which is the reason we should use scheduled intensity.

In another experiment, we test the simple application of the blending method on VOC 2007 (Table 1). The performance is worse than the original model.

Table 1 Ablation analysis of multi-phased training

Method	Backbone	Data	mAP
SSD*[38]	Vgg	07+12	77.2
SSD* + Blending	Vgg	07+12	76.0

SSD* denotes training with tricks mentioned in [38]. *07+12* denotes *VOC trainval2007+trainval2012*

3.3 Blending intensity for training detectors

Unlike image classifiers, object detectors are usually harder to train due to their complexity, especially when using the blending method.

- In the context of this research, the detectors simultaneously produce two different losses: the classification loss and the regression loss. So the complexity of the detection task is higher than the classification task.
- Besides, for each point on the last feature map of the object detector, the prediction of both category and location will be made. Therefore, the loss function of detectors is more complex than the loss function of classifiers.
- Furthermore, blending method creates hybrid categories of objects or objects and backgrounds with hybrid labels combined by labels of original objects, so the human-made images and labels are more complex than original images and labels.

Through the above analysis, it is not suitable to apply the blending method directly to object detectors. Therefore, we propose a multi-phase blending method with incremental blending intensity for training detection networks.

3.4 Enhanced hard negative mining

In the training process of typical object detectors, after the matching step, most of the default boxes are negatives, especially when the number of possible default boxes is large. This introduces a severe imbalance between the positive and negative training examples [38].

The existing method in typical one-stage detectors is hard negative mining. They sorted all negative examples using the highest confidence loss for each default box and pick the top ones so that the ratio between the negatives and positives is at a fixed value.

We consider this problem to be more serious in our method. The blending method will further increase the number of negative examples by creating hybrid categories with more backgrounds than objects, which generally belong to negative examples. However, too many negative examples

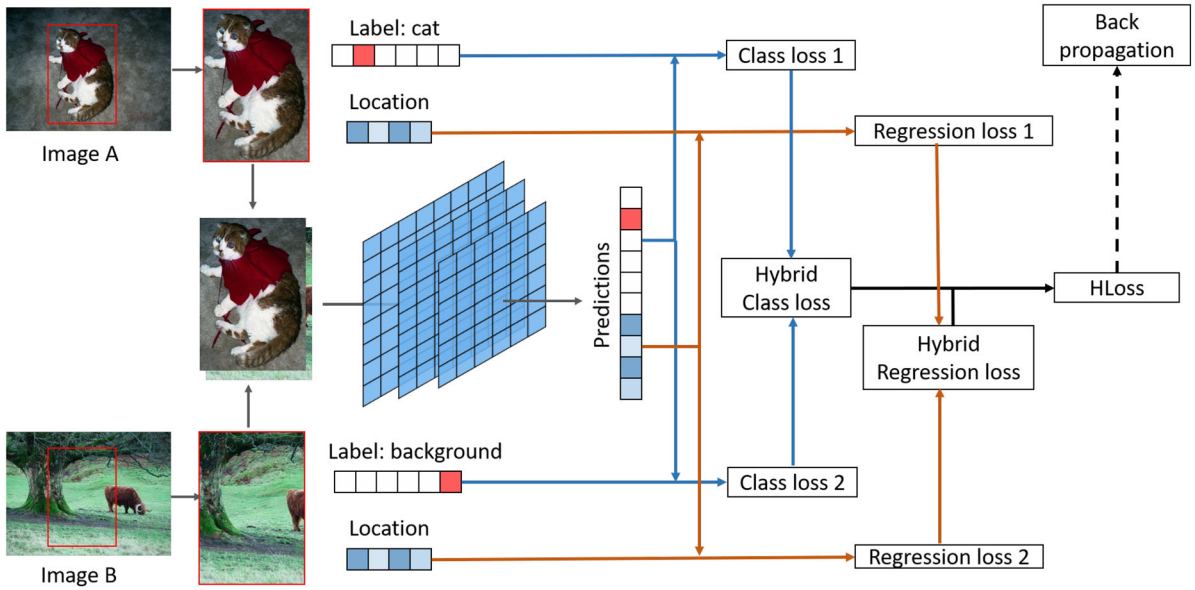


Fig. 5 The overview of incremental blending method. We design two incremental hybrid loss functions containing hybrid classification loss function and hybrid regression loss function

have no advantage in detecting positive examples. On the contrary, they will make the training process more difficult.

Based on the above discovery, we further discard more negative examples in our multi-phase training process than that in previous training methods and processes.

3.5 Blending training architecture and principles

In the one-stage detector, all the labels and the bounding boxes of objects come out simultaneously. The network produces a fixed-size matrix containing all the information of both the detected objects and backgrounds. Each prediction is related to the corresponding area.

Therefore, we can blend two blocks of fixed-size outputs with correct alignment. In this way, we can blend both images and labels (softening effect) in object detection task and propose a novel training architecture, that is, blending training architecture with incremental blending intensity.

The architecture and principles of the proposed method are shown in Fig. 5

- Before inputting data batches to the base network, we present a pairwise operation to hybrid pairs of images in addition to intuitive image processing operations.
- Also, at the tail of the network, we present a hybrid loss function called *HLoss* which contains the hybrid classification loss and the hybrid regression loss.
- The blending degree of the blending method is controlled by the scheduled and incremental blending intensity.

3.6 Details of the Algorithm

For convenience, we abbreviate our method, multi-phased blending method as MPB. MPB includes three parts as follows:

3.6.1 Scheduled blending intensity

We design our scheduled blending intensity λ through sigmoid formulation,

$$\lambda = \frac{\hat{\lambda}}{1 + e^{-\alpha(epoch - n)}} \tag{5}$$

where $\hat{\lambda}$ is the highest value of the blending intensity, α and n are the hyperparameters of λ, and *epoch* denotes the current epoch during training. In most of our experiments, $\hat{\lambda}$, α, n are set to be 0.02, 0.1 and 200 respectively. For the typical detection networks, when epoch goes to around 200, the networks reach the premature stage, from which the loss curve becomes smooth and the network performance keeps stable. Thus, from this stage, we proposed a smoothly incremental blending intensity to further improve the performance of the networks.

3.6.2 Blending method

The blending method includes three major procedures as follows.

In the first step, for a training batch, we randomly select two image and blend them by $x = \lambda x_1 + (1 - \lambda)x_2$. We

construct a new distribution $\mathcal{D}_v$ from source distribution $\mathcal{D}_s := (x_i, y_i)_{i=1}^n$

$$\mathcal{D}_v := (\tilde{x}_i, \tilde{y}_{pi}, \tilde{y}_{qi})_{i=1}^m \tag{6}$$

where $\tilde{x}_i = \lambda x_{pi} + (1-\lambda)x_{qi}, (x_{pi}, y_{pi}), (x_{qi}, y_{qi}) \in \mathcal{D}_s$ and λ is the scheduled blending intensity from Eq. (5). Then we input these blended images from distribution $\mathcal{D}_v$ to calculate the feature maps.

In the second step, we calculate the classification loss and regression loss of the feature maps. The basic classification loss is Crossentropy Loss, $(x, y) \in \mathcal{D}_s, \theta$ denotes the parameters of the network.

$$\text{loss}_{\text{cls}}(\theta) = \frac{1}{m}\sum_{i=1}^{m} L_{\text{CE}}(f_\theta(x_i), y_i). \tag{7}$$

Here we present a new loss function, in which we replace the basic loss with the sum of two losses, $(\tilde{x_i}, y_{pi}, y_{qi}) \in \mathcal{D}_v$

$$\text{loss}_i(\theta) = \frac{1}{m}\sum_{i=1}^{m} L_{\text{CE}}(f_\theta(\tilde{x_i}), y_{pi}) \tag{8}$$

$$\text{loss}_j(\theta) = \frac{1}{m}\sum_{i=1}^{m} L_{\text{CE}}(f_\theta(\tilde{x_i}), y_{qj}) \tag{9}$$

$$\text{loss}_{\text{hybrid}}(\theta) = \lambda \text{loss}_i(\theta) + (1-\lambda)\text{loss}_j(\theta) \tag{10}$$

$$\text{loss}_{\text{cls}}(\theta) = \text{loss}_{\text{hybrid}}(\theta) \tag{11}$$

For localization loss, we modify it in the same way; L_{SM} refers to the Smooth L1 Loss,

$$\text{loss}_i(\theta) = \frac{1}{m}\sum_{i=1}^{m} L_{\text{SM}}(f_\theta(\tilde{x_i}), y_{pi}) \tag{12}$$

$$\text{loss}_j(\theta) = \frac{1}{m}\sum_{i=1}^{m} L_{\text{SM}}(f_\theta(\tilde{x_i}), y_{qj}) \tag{13}$$

$$\text{loss}_{\text{hybrid}}(\theta) = \lambda \text{loss}_i(\theta) + (1-\lambda)\text{loss}_j(\theta) \tag{14}$$

$$\text{loss}_{\text{loc}}(\theta) = \text{loss}_{\text{hybrid}}(\theta) \tag{15}$$

In the third step, we get the *HLoss* by adding $loss_{loc}$ and loss_{cls} together and minimize it to train our network.

$$HLoss(\theta) = \text{loss}_{\text{cls}}(\theta) + \gamma \text{loss}_{\text{loc}}(\theta). \tag{16}$$

We set γ to 1 in our experiments.

3.6.3 Enhanced hard negative mining

After blending operation, we sort all negative examples using the highest confidence loss for each default box and pick the top ones. We keep the ratio between the negatives and positives at 3, besides we discard 20% of these negative examples randomly.

4 Experiments

We apply the method on other networks based on the same datasets. PASCAL VOC [15] and MS COCO [43] have 20 and 80 object categories respectively.

In PASCAL VOC 2007, a predicted bounding box is positive if its Intersection over Union (IoU) with the ground truth is higher than 0.5, while in COCO, it uses various thresholds for more comprehensive calculation. The metric to evaluate detection performance is the mean average precision (mAP).

In MS COCO, following settings in other studies, we use *trainval35k* as training set, which includes *train2014* and *val2014-minival*. We test on *test2015* as the evaluation result. All our training is based on one 1080TI, and pytorch as the platform, we will show the details of each experiment respectively in the following parts.

4.1 PASCAL VOC

In this experiment, we follow [38] by using the same settings and hyperparameters.

For SSD + MPB, we set SGD as the optimizer and the initial learning rate at 0.004, momentum at 0.9, set epoch at 400 and weight decay at 0.0005 and batch size at 32. We set γ at 1 and $\hat{\lambda}$ at 0.1. We used a strategy called warm restart [44] to accelerate the training that gradually ramps up the learning rate from 10^{-6} to 0.004 at the first 5 epochs. After the warm-up phase, the learning rate goes back to 10^{-6} until 200 epoch and keeps it in the following epochs. For settings of the blending training parameter, $\hat{\lambda}, \alpha, n$ is set to be 0.02, 0.1 and 200 respectively. We trained the model for 7.5 hours totally and reached the best model at 340 epoch. For DSSD and YOLOv2, the settings are almost the same as the SSD.

For RFB + MPB, we use a similar strategy and parameters as above. Almost settings follow [39]. We set SGD as the optimizer and the initial learning rate at 0.004, momentum at 0.9. We set the batch size at 32, weight decay at 0.0005 and epoch at 400. We also use the warm-up strategy that gradually ramps up the learning rate from $10 - e6$ to $4 - e3$ at the first 15 epoch. After the warm-up phase, the learning rate goes back to 10^{-6} until 250 epoch and keep it in the following epochs. Similarly, $\hat{\lambda}, \alpha, n$ is set to be 0.02, 0.1 and 200 respectively. We reached the best model at around 390 epoch.

As shown in Tables 2 and 5, we can see the comparison between the networks with and without MPB on the VOC2007 *test set*. SSD* is the updated SSD results with more data augmentation [38]. For a fair comparison, we reimple-

Table 2 Detection results on PASCAL VOC 2007

Method	Backbone	Data	mAP
SSD* [38]	Vgg	07+12	77.2
SSD* + MPB	Vgg	07+12	**78.5**
DSSD [45]	Vgg	07+12	78.6
DSSD + MPB	Vgg	07+12	**79.4**
YOLOv2 544 [36]	Darknet	07+12	78.6
YOLOv2 544 + MPB	Darknet	07+12	**79.2**
RFB300 [39]	Vgg	07+12	80.5
RFB300+MPB	Vgg	07+12	**80.9**
RFB512 [39]	Vgg	07+12	82.2
RFB512 + MPB	Vgg	07+12	**82.5**

MPB multi-phase blending method
The best or better results in the comparative experiments are bold

Table 3 Detection results on PASCAL VOC 2012

Method	Backbone	Data	mAP
SSD* [38]	Vgg	07++12	75.8
SSD* + MPB	Vgg	07++12	**76.9**
YOLOv2 [36]	Darknet	07++12	73.4
YOLOv2 + MPB	Darknet	07++12	**74.0**
RFB512 [39]	Vgg	07++12	81.2
RFB512 + MPB	Vgg	07++12	**81.4**

07++12 denotes *trainval2007+test2007+trainval2012*
The best or better results in the comparative experiments are bold

ment SSD* with Pytorch-0.4 and CUDA9.0 and apply our method in the same environment. We also use the same data augmentation methods in [38]. By using our method, SSD* is greatly improved by 1.3%. DSSD and YOLOv2 also are upgraded by 0.8% and 0.6%. For the latest fast one-stage detector RFBNet, it is also improved obviously by 0.4% and 0.3% for RFB300 and RFB512 respectively.

Another experiment on PASCAL VOC 2012 is shown in Table 3. The settings are same as the above experiments and training set used in this part is *07++12*, which denote *trainval2007* + *test2007* + *trainval2012*. We can see that the improvements on VOC2012 *test* are also marked. SSD*, YOLOv2, and RFBNet512 are greatly improved by 1.1%, 0.6% and 0.2%, respectively.

4.2 MS COCO

In this experiment, the hyperparameters are the same as the previous literature [39] on COCO.

In previous literature, the basic learning rate is set to 0.002, and max epoch is set to 300. We train our network with *trainval35k* that is also used in previous networks. The No.1 one-stage detection network from [39] on COCO is RFB512-E, and hence we also apply our method on RFB512-E in this experiment. As shown in Table 4, our method achieves an improvement to RFBNet300 and RFB512-E by 0.8% and 0.6% respectively. Although MS COCO is more difficult than PASCAL VOC and exists more hard or unclear objects, our method still works well and achieves a better promotion than VOC (Table 5).

4.3 Performance on LRP

Localization recall precision (LRP) [51] is a new performance metric for object detectors, and it can directly measure bounding box localization accuracy. As in mAP, $moLRP$ is the performance metric for the entire detector. Mean optimal box localization, FP, and FN components denoted by $moLRP_{IoU}$, $moLRP_{FP}$ and $moLRP_{FN}$ respectively are similarly defined as the mean of the class-specific components. We test our models and demonstrate results in Table 6. For each metric, smaller is better.

From Table 6, we can know that $moLRP_{IoU}$, $moLRP_{FP}$ and $moLRP_{FN}$ are actually decreased by MPB, which demonstrates the both improvements on classification and localization.

4.4 Two-stage detector

We also test on Faster R-CNN and results are shown in Table 7. In this experiment, the settings of networks are the same as the original one [34]. Other settings of MPB are the same as the experiment of SSD+MPB.

4.5 Ablation experiments

4.5.1 Blending method

In order to better understand the proposed network, we investigate the effect of each component of *HLoss* and compare it with [38]. The comparison is shown in Table 8.

Firstly, we set up the network just by applying our method to the localization part. For the part of localization, we apply the blending method to the process of localization predicting, by adding the *HLoss* to the tail of the localization part. For the part of the classification, as the input images are blended before training, we keep the random parameter λ greater than 0.5 to make sure the first image is the main part and calculate the loss with it. The results show that the method actually improves the performance on the regression task.

Secondly, we set the *HLoss* only on classification. We implement a similar network by only adding *HLoss* to the tail of the classification part. Most of the operations are similar to the above.

The results show that *HLoss* in both components contributes to the improvement in performance for object detection. A combination of them achieves the best result.

Table 4 Comparison between our method and others on MS COCO

Method	Backbone	Time	Avg. 0.5:0.95	Precision, 0.5	IoU: 0.75	Avg. S	Precision. M	Area L
Faster [34]	VGG	147 ms	24.2	45.3	23.5	7.7	26.4	37.1
Faster+++ [26]	ResNet-101	3.36 s	34.9	55.7	37.4	15.6	38.7	50.9
Faster w FPN [46]	ResNet-101-FPN	240 ms	36.2	59.1	39.0	18.2	39.0	48.2
R-FCN [47]	ResNet-101	110 ms	29.9	51.9	–	10.8	32.8	45.0
R-FCN w Deformable CNN [48]	ResNet-101	125 ms	34.5	55.0	–	14.0	37.7	50.3
Mask R-CNN [49]	ResNext-101-FPN	210 ms	37.1	60.0	39.4	16.9	39.9	53.5
YOLOv2 [36]	darknet	25 ms	21.6	44.0	19.2	5.0	22.4	35.5
SSD300* [38]	VGG	12 ms	25.1	43.1	25.8	–	–	–
SSD512* [38]	VGG	28 ms	28.8	48.5	30.3	–	–	–
DSSD513 [45]	ResNet-101	182 ms	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet500 [50]	ResNet-101-FPN	90 ms	34.4	53.1	36.8	14.7	38.5	49.1
RetinaNet800 [50]	ResNet-101-FPN	198 ms	39.1	59.1	42.3	21.8	42.7	50.2
RFBNet300[1] [39]	VGG	15 ms	30.3	49.3	31.8	11.8	31.9	45.9
RFBNet512-E[1] [39]	VGG	33 ms	34.2	54.7	36.1	17.6	37.0	47.6
RFB300+MPB (ours)	VGG	15 ms	**31.1**	50.2	32.7	12.7	33.7	48.6
RFB512+MPB (ours)	VGG	35 ms	**34.8**	55.8	36.5	18.4	37.5	48.9

Statistics are from [39]
[1] Are reimplemented on single 1080ti because [39] used Titan which are not widely available on consumption-level platform
The best or better results in the comparative experiments are bold

Table 5 Class-specific comparative results of MPB on PASCAL VOC 2007

Method	Aeroplane	Bicycle	Bird	Boat	Bottle	Bus	Car	Cat	Chair	Cow	Diningtable
SSD* [38]	79.5	83.9	76.0	69.6	50.5	87.0	85.7	88.1	60.3	81.5	77.0
SSD*+MPB	82.0	86.4	75.8	73.1	55.0	86.4	86.3	86.6	60.4	83.2	75.7
RFB [39]	85.0	86.1	77.7	75.7	60.6	88.9	87.6	86.8	64.2	85.3	77.9
RFB+MPB	84.5	87.3	79.5	74.6	60.0	88.4	88.0	88.3	65.3	84.8	79.8

Method	Dog	Horse	Motorbike	Person	Pottedplant	Sheep	Sofa	Train	Tvmonitor	mAP
SSD* [38]	86.1	87.5	83.97	79.4	52.3	77.9	79.5	87.6	76.8	77.2
SSD*+MPB	85.7	87.5	87.3	79.9	53.3	79.3	80.4	86.6	77.9	**78.4**
RFB [39]	86.1	89.0	87.1	82.2	58.7	81.5	81.1	88.2	81.5	80.6
RFBF+MPB	85.7	89.4	87.1	82.3	57.9	81.4	81.9	88.2	81.6	**80.9**

The best or better results in the comparative experiments are bold

Table 6 Experimental results of SSD and RFB through LRP on MS COCO

Method	mAP	mAP@0.5	moLRP	$moLRP_{IoU}$	$moLRP_{FP}$	$moLRP_{FN}$
SSD-300 [38]	0.161	0.383	0.854	0.281	0.403	0.622
SSD-512 [38]	0.284	0.481	0.763	0.202	0.331	0.549
RFB [39]	0.303	0.493	0.745	0.188	0.320	0.539
RFB512E [39]	0.342	0.547	0.717	0.183	0.299	0.487
RFB+MPB	**0.311**	0.502	**0.735**	0.185	0.304	0.529
RFB512E+MPB	**0.348**	0.558	**0.712**	0.182	0.293	0.480

The best or better results in the comparative experiments are bold

Table 7 Comparative results for Faster RCNN with or without MPB

Method	Backbone	Data	mAP
Faster RCNN [34]	Res101	07+12	80.1
Faster RCNN + MPB	Res101	07+12	**81.1**

The best or better results in the comparative experiments are bold

Table 8 Ablation analysis for hybrid loss function

Method	Class	bbox	mAP
SSD* + MPB	✓		78.0
SSD* + MPB		✓	77.9
SSD* + MPB	✓	✓	**78.5**

Class denotes *classification loss function*, *bbox* denotes *regression loss function*
The best or better results in the comparative experiments are bold

Table 9 Ablation analysis of multi-phased training

Method	Blend	MP	EHNM	mAP
SSD*[38]				77.2
SSD* + MPB	✓			76.0
SSD* + MPB	✓	✓		78.3
SSD* + MPB	✓	✓	✓	**78.5**

Blend Blending method
MP Multi-phase training
EHNM Enhanced Hard Negative Mining
The best or better results in the comparative experiments are bold

4.5.2 Scheduled blending intensity

The comparison between models being trained with or without scheduled blending intensity is shown in Table 9. As we see, models without scheduled blending training lead to be even worse than the baseline model, because object detection datasets are not easy for networks to learn. However, scheduled blending training can overcome this difficulty by gradually increasing the blending intensity, which means that the network has time to adapt to the object detection datasets.

In Fig. 6, the loss of fixed ratio converges slowly, and larger blending intensity makes the network harder to converge , but the loss of our method converges faster because of the low intensity in the early phase.

4.5.3 Enhanced hard negative mining

The comparison between models being trained with or without scheduled and incremental blending intensity is shown in Table 9. As we can see, the enhanced hard negative mining improves the performance of the detection networks, because blending method creates much more negative examples which affect the training process.

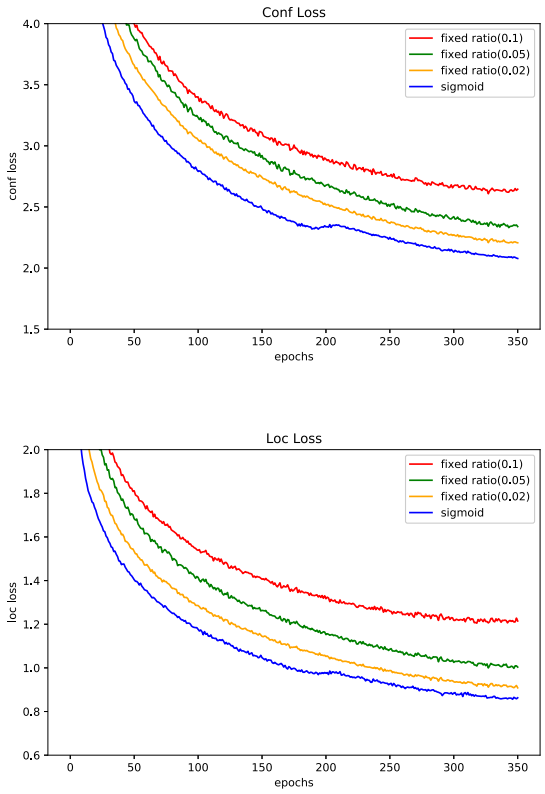


Fig. 6 The confidence and location losses of models with different blending intensity schedules

Table 10 Comparison between different schedules on PASCAL VOC 2007

Method	Schedule	Data	mAP
SSD*[38]	No ratio	07+12	77.2
SSD*	Ratio(0.02)	07+12	76.0
SSD*	Ratio(0.05)	07+12	70.4
SSD*	Ratio(0.1)	07+12	68.8
SSD*	Linear	07+12	70.3
SSD*	Exponential	07+12	78.0
SSD*	Sigmoid	07+12	**78.5**

The best or better results in the comparative experiments are bold

4.6 Comparison of blending schedules

In this experiment, three groups are made to compare the performance: (1) networks trained with no ratio; (2) networks trained with fixed ratio (we set blending intensity at 0.02, 0.05, 0.1); (3) networks trained with scheduled ratio. (We test linear schedule, exponential schedule, and sigmoid schedule.)

According to Table 10, blending with fixed ratio makes the networks worse, and linear scheduled blending method also performs badly due to its fast blending intensity increasing in the early time. Exponential schedule performs better than

Table 11 Comparison between MPB and other methods on PASCAL VOC 2007

Method	Backbone	Data	mAP
SSD [38]	Vgg	07+12	74.3
SSD* [38]	Vgg	07+12	77.2
SSD* + LM	Vgg	07+12	77.5
SSD* + RE	VGG	07+12	77.7
SSD* + MPB	Vgg	07+12	**78.5**

The best or better results in the comparative experiments are bold

Table 12 Comparison of performances for different quantity of blended data

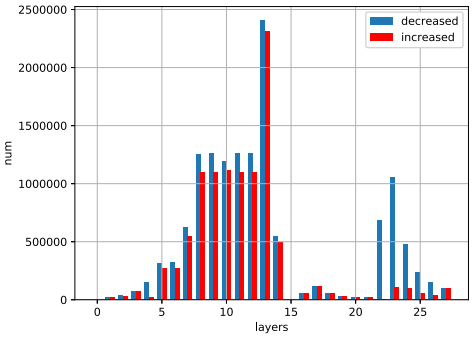
Data	50 k	75 k	100 k	125 k	150 k
CIFAR10	93.60	93.65	93.73	93.81	93.88

the baseline but worse than the sigmoid schedule due to the low intensity in mid time.

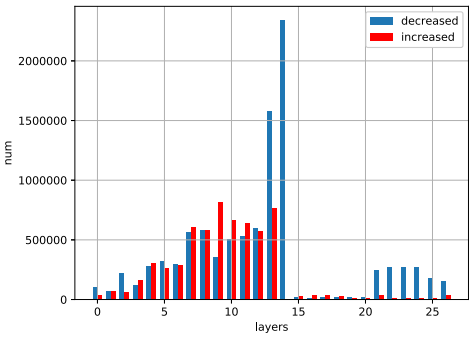
4.7 Comparison with other data augmentation methods

We also compare our method with other data augmentation methods which can work on one-stage detectors (label smoothing [30] and random erasing [52] and traditional methods). SSD is the typical network, and SSD* comes with extra augmentation methods [38]. For SSD* + LM(label smoothing), we soften the classification labels for each object by set 0.9 and 0.1/20 in which the previous value is 1 and 0, respectively. For SSD*+RE(random erasing), we use its default setting. SSD* + MPB is the one with blending method. All the networks are trained under the same environment and same hyperparameters. Our method can further improve the detection models based on traditional augmentation methods. Compared with label smoothing and random erasing, our method is more effective.

We get the final results shown in Table 11. Our method is better than label smoothing.



(a) Comparison on weights

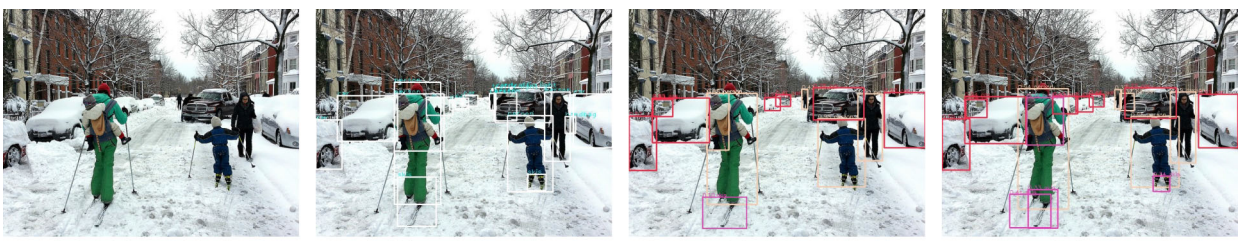


(b) Comparison on biases

Fig. 8 In **a** the blue bar refers to the number of decreased weights from original SSD to SSD with MPB, and the pink bar refers to the number of increased weights from original SSD to SSD with MPB. Similarly, in **b** the blue bar refers to the number of decreased biases from original SSD to SSD with MPB, and the pink bar refers to the number of increased biases from original SSD to SSD with MPB

4.8 Quantity of blended data

We conducted an experiment to compare different amounts of mixed data based on CIFAR10. Five different CIFAR10 datasets are designed including 1 original CIFAR10 dataset of 50 k images and 4 expanded datasets (75 k, 100 k, 125 k, 150 k). These datasets are trained on VGG19, and the final results are shown in Table 12. Obviously, the model trained



(a) Image **(b)** Ground Truth **(c)** RFBNet **(d)** Ours

Fig. 7 Comparison between RFBNet and RFBNet+MPB. RFBNet+MPB performs better on low-confidence object and gives more detections on uncertain area

with expanded dataset outperforms the original model, and the model performs better with more additional blended data.

5 Analysis

Based on Sect. 4, our method improves the performance on the object detection network. In this section, we lead a deep analysis of how this architecture gets a better result.

Firstly, through the proposed method of blending pairs operation, the diversity of the dataset is enhanced, which improves the regularization and generalization of the network. The observation of the experiment result also confirms the proposed idea as follows.

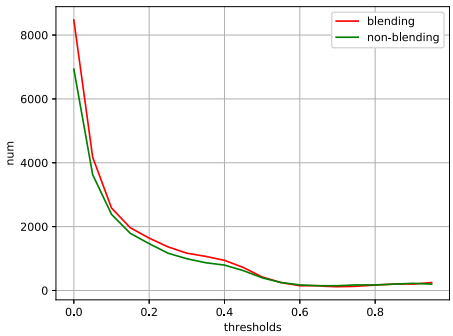


Fig. 9 The number of detected objects of SSD with MPB and not. The results are obtained from PASCAL VOC 2007 test dataset by SSD+MPB

- Our experiment compares all the weights between the original SSD network and the improved network with MPB which are trained in previous experiments.
- As shown in Fig. 8, the weights and biases are decreased by MPB, which means it actually regularizes the network.

Secondly, we analyze the final detection result to show how our network improves confidence in the previous RFBNet as follows.

- As shown in Fig. 7, in the best case of the ski, the confidence of it grows 4x from less than 0.1 in RFB to 0.4 in RFBNet+MPB. In the worst case of a woman in green, the confidence of her varies a little from 0.96 in RFBNet to 0.94 in RFBNet+MPB (Fig. 8).
- Networks being trained with our method tries to give more confidence to uncertain objects such as some small and illegible objects which are hard to be detected by previous methods (Fig. 9). Our method has slight fluctuations on the high-confidence objects due to the effect of softening and this will not impact the final result. More examples are listed in Figs. 10 and 11.

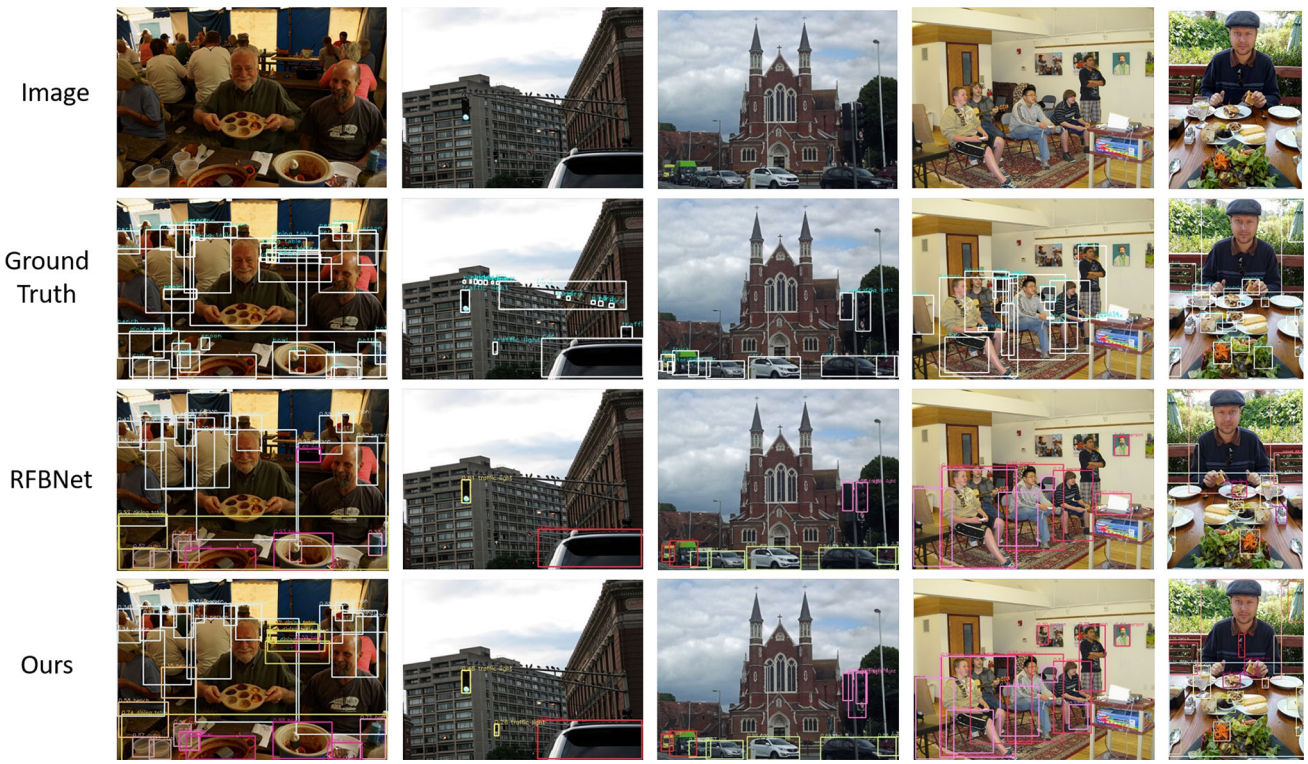
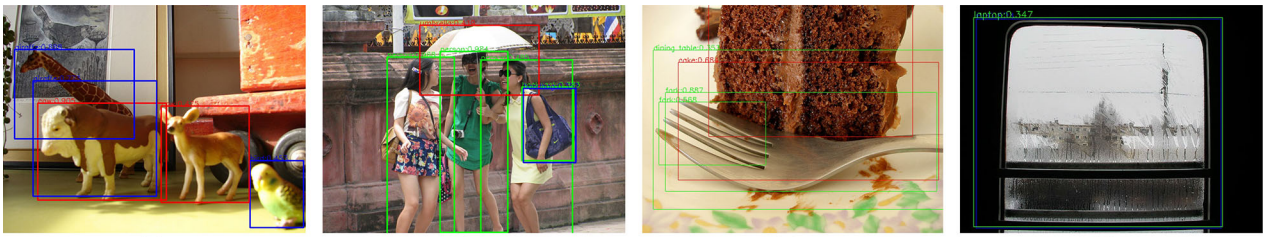


Fig. 10 More examples of comparison. Pictures are selected from MS COCO dataset. As we can see, the model trained with MPB gives more possible predictions boxes than the original one



(a) (b) (c) (d)

Fig. 11 Bad examples on MS COCO dataset, **a** gives a wrong bounding box of giraffe, **b** gives labels of handbag and backpack simultaneously to the handbag with low confidence, **c** gives a wrong bounding box of the fork, **d** label the window incorrectly

Thirdly, We also explore the improvement on the number of the successfully detected objects for medium or low overlap ($overlaps <= 0.5$) with ground truth as follows.

- As shown in Fig. 9, RFBNet+MPB increases the number of medium or overlap objects by 15.2%, which means that RFBNet+MPB gives more correct detections.
- Benefit from high successful detection rate, RFBNet+MPB gives more accurate predictions than the original network, which eventually leads to a decrease on regression loss.

6 Conclusion and future work

In this paper, we propose a novel multi-phase blending method with incremental blending intensity for training detection networks. Besides, we design an incremental hybrid loss function containing both classification loss function and regression loss function. Furthermore, we discard more negative examples than the existing methods. In this way, we can stabilize the training process of object detection networks and eventually regularize the networks to achieve remarkable improvements on one-stage detectors. The experiments demonstrate the validity of the proposed method. One limitation is that hyperparameters is handcrafted. It is necessary to take several experiments to find the best hyperparameters for each model. Thus, in future work, we will explore adaptive blending training methods to automatically searching the optimized hyperparameters. Secondly, we also want to continue the research on other specific problems in the detection task. Finally, will also plan to extend our idea to other areas of computer science and applications [53–61], especially in the areas of intelligent computing [62–64] and visual computing [65–67].

Funding This study was funded by NSFC (Grant No. 61472289).

Compliance with ethical standards

Conflict of interest All the authors declare that they have no conflict of interest.

References

1. Kán, P., Kafumann, H.: Deeplight: light source estimation for augmented reality using deep learning. Vis. Comput. **35**(6–8), 873–883 (2019)
2. Luciano, L., Hamza, A.B.: Deep similarity network fusion for 3d shape classification. Vis. Comput. **35**(6–8), 1171–1180 (2019)
3. Li, H., He, F., Liang, Y., Quan, Q.: A dividing-based many-objectives evolutionary algorithm for large-scale feature selection. Soft Comput. (2019). https://doi.org/10.1007/s00500-019-04324-5
4. Zhang, S., He, F., Ren, W., Yao, W.: Joint learning of image detail and transmission map for single image dehazing. Vis. Comput. (2018). https://doi.org/10.1007/s00371-018-1612-9
5. Pan, Y., He, F., Yu, H.: A correlative denoising autoencoder to model social influence for top-n recommender system. Front. Comput. Sci. (2019). https://doi.org/10.1007/s11704-019-8123-3
6. Chen, X., He, F., Yu, H.: A matting method based on full feature coverage. Multimed. Tools Appl. **78**(9), 11173–11201 (2019)
7. Yu, H., He, F., Pan, Y.: A novel segmentation model for medical images with intensity inhomogeneity based on adaptive perturbation. Multimed. Tools Appl. **78**(9), 11779–11798 (2019)
8. Li, K., He, F., Yu, H., Chen, X.: A parallel and robust object tracking approach synthesizing adaptive bayesian learning and improved incremental subspace learning. Front. Comput. Sci. **13**(5), 1116–1135 (2019)
9. Yu, H., He, F., Pan, Y.: A novel region-based active contour model via local patch similarity measure for image segmentation. Multimed. Tools Appl. **77**(18), 24097–24119 (2018)
10. Li, K., Fa-Zhi, H.E., Yu, H-p, Chen, X.: A correlative classifiers approach based on particle filter and sample set for tracking occluded target. Appl. Math. J. Chin. Univ. **32**(2), 294–312 (2017)
11. Li, K., He, F.Z., Yu, H.P.: Robust visual tracking based on convolutional features with illumination and occlusion handing. J. Comput. Sci. Technol. **33**(1), 223–236 (2018)
12. Sun, J., Fa-Zhi, H.E., Chen, Y.L., Xiao, C.: A multiple template approach for robust tracking of fast motion target. Appl. Math. J. Chin. Univ. **31**(2), 177–197 (2016)
13. Yu, H., He, F., Pan, Y.: A scalable region-based level set method using adaptive bilateral filter for noisy image segmentation. Multimed. Tools Appl. (2019). https://doi.org/10.1007/s11042-019-08493-1
14. Girshick, R., Donahue, J., Darrell, T., Malik, J.: Rich feature hierarchies for accurate object detection and semantic segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 580–587 (2014)
15. Everingham, M., Van Gool, L., Williams, C.K., Winn, J., Zisserman, A.: The pascal visual object classes (voc) challenge. Int. J. Comput. Vis. **88**(2), 303–338 (2010)
16. Girshick, R.: Fast r-cnn. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 1440–1448 (2015)

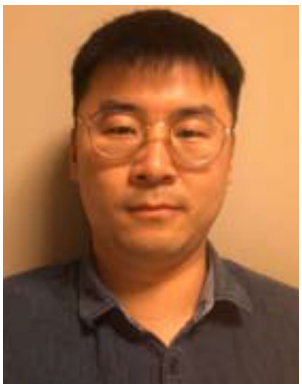
17. Zhang, J., Wang, C., Li, C., Qin, H.: Example-based rapid generation of vegetation on terrain via CNN-based distribution learning. Vis. Comput. **35**(6–8), 1181–1191 (2019)
18. Yu, Z., Liu, Q., Liu, G.: Deeper cascaded peak-piloted network for weak expression recognition. Vis. Comput. **34**(12), 1691–1699 (2018)
19. Li, Y., Wang, Z., Yang, X., Wang, M., Poiana, S.I., Chaudhry, E., Zhang, J.: Efficient convolutional hierarchical autoencoder for human motion prediction. Vis. Comput. **35**(6–8), 1143–1156 (2019)
20. Arashloo, S.R., Kittler, J.: Dynamic texture recognition using multiscale binarized statistical image features. IEEE Trans. Multimed. **16**(8), 2099–2109 (2014)
21. Zhang, S., Han, Z., Lai, Y.-K., Zwicker, M., Zhang, H.: Stylistic scene enhancement GAN: mixed stylistic enhancement generation for 3d indoor scenes. Vis. Comput. **35**(6–8), 1157–1169 (2019)
22. Vapnik, V.: Statistical Learning Theory, vol. 3. Wiley, New York (1998)
23. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: beyond empirical risk minimization, arXiv preprint arXiv:1710.09412 (2017)
24. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning requires rethinking generalization, arXiv preprint arXiv:1611.03530 (2016)
25. Simonyan K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition, arXiv preprint arXiv:1409.1556 (2014)
26. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778 (2016)
27. Simard, P.Y., LeCun, Y.A., Denker, J.S., Victorri, B.: Transformation invariance in pattern recognition–tangent distance and tangent propagation. In: Orr, G.B., Müller, K.-R. (eds.) Neural Networks: Tricks of the Trade, pp. 239–274. Springer, Berlin (1998)
28. Chapelle, O., Weston, J., Bottou, L., Vapnik, V.: Vicinal risk minimization. In: Advances in Neural Information Processing Systems, pp. 416–422 (2001)
29. Xiao, T., Xia, T., Yang, Y., Huang, C., Wang, X.: Learning from massive noisy labeled data for image classification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2691–2699 (2015)
30. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826 (2016)
31. Guo, H., Mao, Y., Zhang, R.: Mixup as locally linear out-of-manifold regularization, arXiv preprint arXiv:1809.02499 (2018)
32. Takahashi, R., Matsubara, T., Uehara, K.: Ricap: random image cropping and patching data augmentation for deep CNNs. In: Proceedings of The 10th Asian Conference on Machine Learning, ser. Proceedings of Machine Learning Research, J. Zhu and I. Takeuchi, Eds., vol. 95. PMLR, 14–16 Nov 2018, pp. 786–798
33. Uijlings, J.R., Van De Sande, K.E., Gevers, T., Smeulders, A.W.: Selective search for object recognition. Int. J. Comput. Vis. **104**(2), 154–171 (2013)
34. Ren, S., He, K., Girshick, R., Sun, J.: Faster R-CNN: towards real-time object detection with region proposal networks. In: Advances in Neural Information Processing Systems, pp. 91–99 (2015)
35. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. IEEE Trans. Pattern Anal. Mach. Intell. (2018)
36. Redmon, J., Farhadi, A.: Yolo9000: better, faster, stronger, arXiv preprint (2017)
37. Redmon, J., Divvala, S., Girshick, R., Farhadi, A.: You only look once: Unified, real-time object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 779–788 (2016)
38. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., Berg, A.C.: SSD: single shot multibox detector. In: European Conference on Computer Vision, pp. 21–37. Springer (2016)
39. Liu, S., Huang, D., Wang, A.: Receptive field block net for accurate and fast object detection. In: The European Conference on Computer Vision (ECCV) (2018)
40. Teng, C.-M.: A comparison of noise handling techniques. In: FLAIRS Conference, pp. 269–273 (2001)
41. Zhu, X., Wu, X.: Class noise vs. attribute noise: a quantitative study. Artif. Intell. Rev. **22**(3), 177–210 (2004)
42. Bartlett, P.L., Jordan, M.I., McAuliffe, J.D.: Convexity, classification, and risk bounds. J. Am. Stat. Assoc. **101**(473), 138–156 (2006)
43. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L.: Microsoft coco: common objects in context. In: European Conference on Computer Vision, pp. 740–755. Springer (2014)
44. Loshchilov, I., Hutter, F.: SGDR: stochastic gradient descent with warm restarts," arXiv preprint arXiv:1608.03983 (2016)
45. Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., Berg, A.C.: DSSD: deconvolutional single shot detector, arXiv preprint arXiv:1701.06659 (2017)
46. Lin, T.-Y., Dollár, P., Girshick, R.B., He, K., Hariharan, B., Belongie, S.J.: Feature pyramid networks for object detection. In: CVPR, vol. 1, no. 2, p. 4 (2017)
47. Dai, J., Li, Y., He, K., Sun, J.: R-FCN: object detection via region-based fully convolutional networks. In: Advances in Neural Information Processing Systems (NIPS 2016), pp. 379–387. Neural Information Processing Systems Foundation, Inc (2016)
48. Dai, J., Qi, H., Xiong, Y., Li, Y., Zhang, G., Hu, H., Wei, Y.: Deformable convolutional networks, CoRR, abs/1703.06211, vol. 1, no. 2, p. 3 (2017)
49. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask R-CNN. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2961–2969 (2017)
50. Lin, T.-Y., Goyal, P., Girshick, R., He, K., Dollár, P.: The IEEE International Conference on Computer Vision (ICCV), pp. 2980–2988. Springer (2017)
51. Oksuz, K., Can Cam, B., Akbas, E., Kalkan, S.: Localization recall precision (lrp): a new performance metric for object detection. In: Proceedings of the European Conference on Computer Vision (ECCV), pp. 504–519 (2018)
52. Zhong, Z., Zheng, L., Kang, G., Li, S., Yang, Y.: Random erasing data augmentation," arXiv preprint arXiv:1708.04896 (2017)
53. Yan, X., He, F., Hou, N., Ai, H.: An efficient particle swarm optimization for large-scale hardware/software co-design system. Int. J. Coop. Inf. Syst. **27**(01), 1741001 (2018)
54. Liu, X., Xu, Q., Wang, N.: A survey on deep neural network-based image captioning. Vis. Comput. **35**(3), 445–470 (2019)
55. Zhang, J., He, F., Chen, Y.: A new haze removal approach for sky/river alike scenes based on external and internal clues. Multimed. Tools Appl. (2019). https://doi.org/10.1007/s11042-019-08399-y
56. Abbasi, A., Kalkan, S., Sahillioğlu, Y.: Deep 3d semantic scene extrapolation. Vis. Comput. **35**(2), 271–279 (2019)
57. Hou, N., He, F., Zhou, Y., Chen, Y.: An efficient GPU-based parallel tabu search algorithm for hardware/software co-design. Front. Comput. Sci. (2019). https://doi.org/10.1007/s11704-019-8184-3
58. Li, H., He, F., Yan, X.: IBEA-SVM: an indicator-based evolutionary algorithm based on pre-selection with classification guided by SVM. Appl. Math. J. Chin. Univ. **34**, 1–26 (2019)
59. Zhang, S., He, F.: DRCDN:learning deep residual convolutional dehazing networks. Vis. Comput. (2019). https://doi.org/10.1007/s00371-019-01774-8

60. Wu, Y., He, F., Zhang, D., Li, X.: Service-oriented feature-based data exchange for cloud-based design and manufacturing. IEEE Trans. Serv. Comput. **11**(2), 341–353 (2018)
61. Zhang, Z., Han, C., He, S., Liu, X., Zhu, H., Hu, X., Wong, T.-T.: Deep binocular tone mapping. Vis. Comput. **35**(6–8), 997–1011 (2019)
62. Luo, J., He, F., Yong, J.: An efficient and robust bat algorithm with fusion of opposition-based learning and whale optimization algorithm. Intell. Data Anal. **24**(3), 500–519 (2020)
63. Yong, J-s, He, F-z, Li, H-r, Zhou, W-q: A novel bat algorithm based on cross boundary learning and uniform explosion strategy. Appl. Math. J. Chin. Univ. **34**(4), 480–502 (2019)
64. Zhou, Y., He, F., Qiu, Y.: Dynamic strategy based parallel ant colony optimization on GPUs for TSPs. Sci. China Inf. Sci. **60**(6), 068102 (2017)
65. Rasool, S., Sourin, A.: Real-time haptic interaction with RGBD video streams. Vis. Comput. **32**(10), 1311–1321 (2016)
66. Dal Corso, A., Frisvad, J.R., Mosegaard, J., Baerentzen, J.A.: Interactive directional subsurface scattering and transport of emergent light. Vis. Comput. **33**(3), 371–383 (2017)
67. Eren, M.T., Balcisoy, S.: Evaluation of x-ray visualization techniques for vertical depth judgments in underground exploration. Vis. Comput. **34**(3), 405–416 (2018)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Quan Quan is currently a masters student in School of Computer Science, Wuhan University. His research interests are image processing, computer graphics, deep learning and machine learning.

Fazhi He received his bachelors, masters and PhD degrees from Wuhan University of Technology. He was a post-doctor researcher in The State Key Laboratory of CAD&CG at Zhejiang University, a visiting researcher in Korea Advanced Institute of Science & Technology and a visiting faculty member in the University of North Carolina at Chapel Hill. Now he is a professor in School of Computer Science, Wuhan University. His research interests are artificial intelligence, intelligent computing, computer graphics, image processing, computer-aided design, computer-supported cooperative work and co-design of software/hardware.



Haoran Li is currently a PhD student in School of Computer Science, Wuhan University. His research interests are multi-objective optimization, pattern recognition, deep learning and machine learning.