# MTNet: A Mix-Training Network for Fast and Accurate Object Detection

QUAN quan [1], HE Fa-zhi [*1], LI Han-ran [1]
[1] *School of Computer Science and Technology, Wuhan University, Wuhan, Hubei, China*

***Abstract*—Object detection is an important topic for image process in multimedia area. Although a number of approaches have been studied, it still remain a challenge. In this paper, we present a mix-training network to improve accuracy of one-stage object detector and achieve state-of-the-art result. Firstly, different from the typical detection preprocess operations such as crop, flip and rotation on images, we propose a pairwise operation to mix two images by convex combination with random weight. Secondly, in the last step of training the object detector, we calculate the two losses between the prediction label of output and the groundtruth label of the two original images. And then the hybrid loss is obtained by adding the two losses together with the same weight as that in above pairwise operation. Finally, the hybrid loss is used for back propagation solution. 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. Our experiments shows the proposed method improves the generalization of the neural network architectures. On PASCAL 2007 VOC and MS COCO dataset, Mix-Training Network (MTNet) outperform the state-of-the-art result in real time processing.**

***Index Terms*—Object Detection, Data Augmentation, Convolutional Neural Network**

## I. Introduction

Deep neural networks have fundamental advantages over traditional mutlimedia processing methods [1]–[11]. For object detection task, since R-CNN [12] was proposed 4 years ago, the accuracy on VOC [13] dataset has gradually improved. Different from R-CNN and Fast R-CNN [14], Faster R-CNN are fully convolutional networks based method. Furthermore, one-stage object detection approach, such as SSD, combines the two stages in Faster R-CNN to get both the bounding boxes and labels in the same output. Although the accuracy of one-stage detector 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 [15]–[19]. 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 [20]. The classical theory for machine learning tell us that the convergence of ERM is guaranteed as long as the size of the learning machine does not increase with the number of training data.

However, a recent research [21] shows suspect opinion, that ERM allows large neural networks to memorize ( instead of generalize from ) the training data despite that previous works conduct a lot of tricks such as taking strong regularization, applying random label for classification problem.

The second problem of ERM is that neural networks trained with ERM give the opposite predictions for the custom examples. Though the application of neural networks is rapid in these years [22], [23], the generalization on testing distribution is not clearly stable and excellent, and the performance can be impacted easily by the training and testing data.

Typical data augmentation to address above problems can be found in classification task [24] and can be formalized by the Vicinal Risk Minimization (VRM) principle [25], which try to train networks on similar but different examples. The basic methods include slightly image rotation, random crop, horizontal flip, mild scaling, etc. Other improved methods are noisy labeled data [26] by adding noises to labels, label smoothing [27] by soften the label from one-hot to no explicit ones and zeros in labels.

However, these data augmentation are oriented for classification task with assumption that the examples in the vicinity share the same class. Although a recent mixup [28] method for classification try to blend the the inputs and their targets across different classes, how well the method work for detection task is not clear.

Furthermore, these data augmentation focus on data, including across classes data. They do not work on loss function. Whether they are suitable for detection task is unknown.

In this paper, we present a novel collaborative mix-training approach, which not only blends the images but also mixes the loss function. The experiments demonstrate that the proposed approach achieves the state-of-the-art performance.

The rest of this paper is organized as follows. Section.II briefly reviews the related work in object detection. Section.III presents our mix-training approach for one-stage object detector. Section.IV conducts the experiments and discusses empirical results. Section.V analyses the high lights of the proposed network. Section.VI concludes this work and discusses the future work.

## II. Related Work

### A. Detection

**Early two-stage detector**: R-CNN [12] is a standard two-stage object detection framework to produce the bounding boxes and to classify and accurate the boxes. [12] combines the steps of cropping box proposals like SS and classifying them through a CNN model, yielding a significant accuracy gain. For speeding up, Fast R-CNN [14] 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.

**Full neural netwok-based two-stage detector**: Faster R-CNN [29] 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 a class-agnostic prediction and a regression prediction of an offset fit the object location for each anchor.

Full neural netwok-based two-stage is later extended to many more advanced versions. Faster R-CNN runs much faster than Fast R-CNN does, however it still has to apply region-specific computation for many times. One typical extension of it is Mask R-CNN [30] , 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 [31] [32] and SSD [33]. 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 YOLO try to precisely localize some objects, especially for ones. SSD [33] is another one-stage object detection approach and is widely used in pedestrian detection, car detection, object tracking, etc. Different from two-stage detection, SSD produces the results of bounding boxes and class labels from the feature map in the same time through the location layer and classification layer, so this framework is faster than two-stage detector but less accurate.

RFBNet [34] is improved the basic SSD. It adds a module called Receptive Field Block (RFB), which consists several convolutional kernels of different size in parallel. Compared with the inceptive block [27], RFB uses different length of stride and bigger kernel to ensure the feature map totally covered. So RFB block expands the receptive field of layers to have the ability to access more information.

We further improve above three state-of-the-art One-stage detectors. In our approach, we present a new data mixed method to augment the data diversity for input data and propose a new loss function to combine two different losses of labels of a pair of images. The proposed MTNet gets a more accurate result than the state-of-the-art networks.

### *B. Data Augmentation*

**Intuitive Image Operations:** Most existing methods of data augmentation used in object detection are limited by using intuitive image operations, such as crop, rotation which are slight changes for objects.

However, these operations does not change the images obviously.

**Label Noisy**: Learning with noisy labeled training data has been extensively studied in machine learning and computer vision literature.

Limitations are still existing. Experiments in [35] 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 [36]. Bartlett et al. [37] prove that most of the loss functions are not completely robust to label noise.

**Label Smoothing**: There existed several related label smoothing methods [27], [28].

[27] tries to soft the label by adding additional labels of each classes 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 softlization is random, and little influence on some networks. By contrast to [27], we use the explicit image information to get the same effect of overfitting and avoid any wrong information.

Furthermore, [28] assumes 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 mixup the two original images by simply adding together with a random percentage, the label of each also need to be add together with the same percentage and thus the new images and labels are produced to train the neural networks. In classification problems, [28] achieve better result than [27] .

Our work is different from above literature [27], [28] as follows. (1) Our work is oriented for object detection which includes both regression problems and classification problems, while the above methods are only oriented for classification problems. (2) In addition to mixing the labels, our method also construct a new hybrid loss function, while the above literatures only preprocess the labels.

As brief summary of this section, our contributions lie in three folds: (1) We proposed a new mix method, which is valid for regression problem in context of detection task. (2) We propose a new loss function and a new training method for one-stage detector. (3) We construct Mix-Training Network (MTNet) which outperforms the state-of-the-art real-time detection network.

## III. The Proposed Method

### *A. The principle of the proposal method*

Firstly, most existing methods of data augmentation used in object detection are limited by using intuitive image operations, such as crop, rotation which are slight changes for objects. However, these operations does not change the images obviously. Therefore, our idea is to blend two images by convex combination to explore diversity of the data being trained.

Secondly, for object detection task, we also propose a new loss function to indirectly ”soften” the labels by mixing the losses, which has two synchronized benefits. One is that we can achieve the effect of ”softening” labels to improve the diversity of data. Two is that we can support regression problem for object detection.

Thirdly, in the process of predicting location of bounding boxes, the coordinates of bounding boxes are continuous values, the effect of ”softening” labels are continuous values, which match well the object detection task.

The effect of ”softening” labels is described as following formulations.

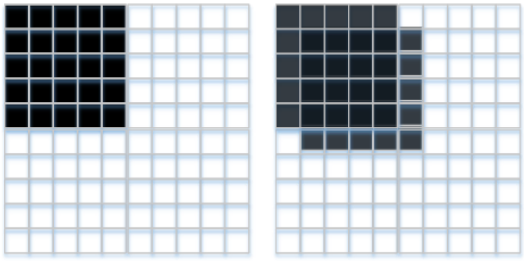


Fig. 1. Left is the original image(black box), right is the mixed image(mixed black box)

$$CEloss(q_i, p_i) = \sum_{i=1} p_i \log q_i$$

$$\tilde{y} = \lambda y_i + (1-\lambda) y_j$$

$$\begin{aligned} &loss(f(\tilde{x}), \tilde{y}) \\ &\quad = (\lambda y_i) log(f(\tilde{x})) + ((1-\lambda) y_j) log(f(\tilde{x})) \\ &\quad = \lambda y_i log(f(\tilde{x})) + (1-\lambda y_j) log(f(\tilde{x})) \\ &\quad = \lambda loss_i(f(\tilde{x}), y_i) + (1-\lambda) loss_j(f(\tilde{x}), j_j) \end{aligned} \tag{1}$$

$$loss(f(\tilde{x}), \tilde{y}) = \lambda loss_i(f(\tilde{x}), y_i) + (1-\lambda) loss_j(f(\tilde{x}), j_j) \tag{2}$$

$CELoss$ refers to CrossEntropy Loss, which widely used in classification task. $\tilde{x}, \tilde{y}$ refer to the mixed image and its label. The formulas above prove that for classification problem, hence the *MixLoss* has the same effect as mixed labels.

### B. Initial Experiment

To initially test the validation 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 Figure.1, We create a white 10*10 square box containing a 5*5 black box. We establish a selected data distribution from the original distribution to simulate the natural situation that the detection datasets (like PASCAL VOC, etc) is a sample distribution from 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. $x_i$ and $y_i$ refer to the image and 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 $\mathcal{D} := (x_i, y_i)_{i=1}^m$ for images by proposed mixing operation.

$$\tilde{x_i} = \lambda x_i + (1-\lambda) x_j$$

$$\tilde{y_i} = y_i$$

$$\tilde{z_i} = y_j$$

$\lambda \sim Beta(\alpha, \alpha)$ In our experiment, we set $\alpha = 0.1$.

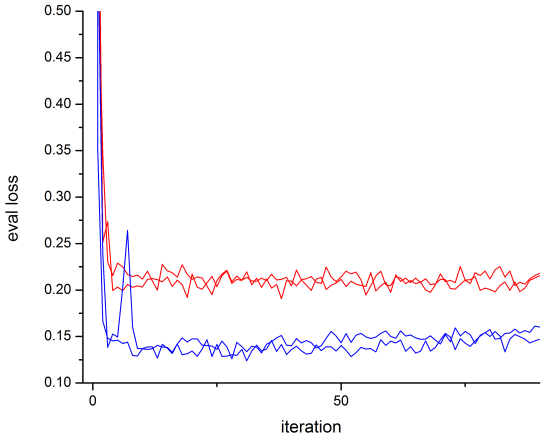
Secondly, we detect the location of black box with small-scale AlexNet for initial test, in which we trained the network by the proposed loss function *MixLoss*

$$L(f) = \min l_{mix}(f) \tag{3}$$

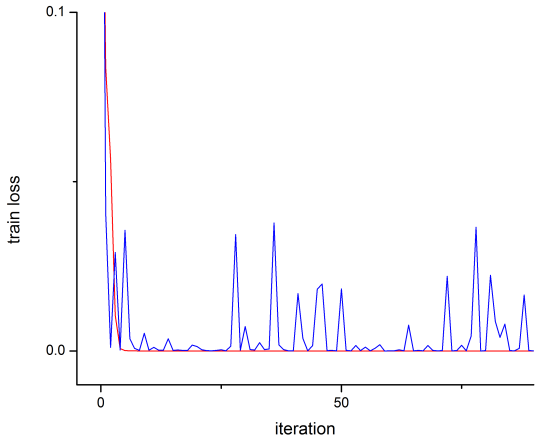
$$l_{mix}(f) = \lambda loss_p(f) + (1-\lambda) loss_q(f) \tag{4}$$

$$loss_p = L(f(\tilde{x_i}), y_i)$$

$$loss_q = L(f(\tilde{x_i}), z_i)$$




(a) Testing




(b) Training

Fig. 2. Graphs refer to the eval loss of models which are trained with mix-training Vs. no mix-training. Blue lines are examples with mix-training and red lines are not. The training loss shows that mix-training is unstable in training process, but the evaluation loss of mix-training is smaller than that of no mix-training.

As shown in Figure.2, the results obviously show that the mix-training is valid. More detailed analysis for mix-training is in Section.V

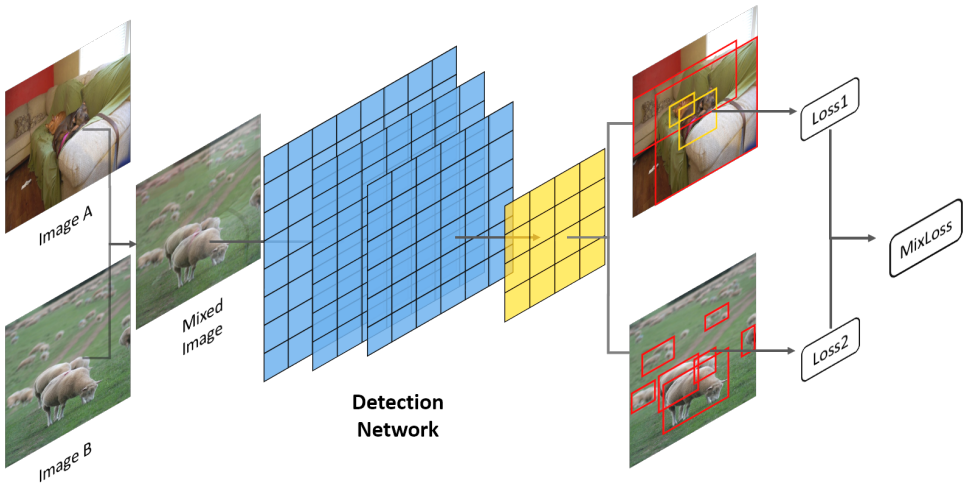



Fig. 3. The overview of Mix-Training Network (MTNet). Image A and B are selected from training batches, and the *MixLoss* combines two losses to one.

### C. Mix-Training Detection Architecture

For most of the detection networks, end-to-end learning is used to train the whole network. End-to-end learning usually refers to omitting any hand-crafted intermediary algorithms and directly learning the solution of a given problem from the sampled dataset. This could involve concatenation of different networks such as multiple CNNs and LSTMs, which are trained simultaneously. The above end-to-end methods have advantages, but still exist much room to improve the network.

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

Therefore we can mix two block of fixed-size outputs with correct alignment. In this way, we can mix both images and labels (softening effect) in object detection task and proposed a novel network, Mix-Training Network (MTNet).

The architecture of the proposed network is shown in Figure.3

- Before inputting data batches to the base network, we present a pairwise operation to mix pairs of images in addition to intuitive image processing operations.
- Also, at the tail of network, we present a hybrid loss function called *MixLoss* to achieve the effect of softening labels.

### D. Details of the Algorithm

Our method includes three major procedures as follows.

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

We construct a new distribution $\mathcal{D}_v$

$$\mathcal{D}_v := (\tilde{x_i}, \tilde{y_p}, \tilde{y_q})_{i=1}^{m}$$

where $\tilde{x} = \lambda * x_p + (1-\lambda) * x_q$, $(x_p, y_p), (x_q, y_q) \in \mathcal{D}_S$ Then we input these mixed image from distribution $\mathcal{D}_v$ to calculate the feature maps.

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

$$loss_{cls}(\theta) = \frac{1}{m} \sum_{i=1}^{m} L_{CE}(f_\theta(x_i), y_i) \tag{5}$$

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

$$loss_{cls}(\theta) = loss_{mix}(\theta) \tag{6}$$

$$loss_{mix}(\theta) = \lambda * loss_i(\theta) + (1 - \lambda) * loss_j(\theta) \tag{7}$$

$$loss_i(\theta) = \frac{1}{m} \sum_{i=1}^{m} L_{CE}(f_\theta(\tilde{x_i}), y_{pi})$$

$$loss_j(\theta) = \frac{1}{m} \sum_{i=1}^{m} L_{CE}(f_\theta(\tilde{x_i}), y_{qj})$$

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

$$loss_{loc}(\theta) = loss_{mix}(\theta) \tag{8}$$

$$loss_{mix}(\theta) = \lambda * loss_i(\theta) + (1 - \lambda) * loss_j(\theta) \tag{9}$$

$$loss_i(\theta) = \frac{1}{m} \sum_{i=1}^{m} L_{SM}(f_\theta(\tilde{x_i}), y_{pi})$$

$$loss_j(\theta) = \frac{1}{m} \sum_{i=1}^{m} L_{SM}(f_\theta(\tilde{x_i}), y_{qj})$$

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

$$\min MixLoss(\theta) = loss_{cls}(\theta) + \gamma * loss_{loc}(\theta) \tag{10}$$

We set $\gamma$ to 1 in our experiments.

## IV. EXPERIMENTS

We compare our network with other state-of-the-art networks on same data sets. PASCAL VOC [13] and MS COCO [38] 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 literature, 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 platform, in most of experiments we reload and finetune the pretrained model, we will show the details of each experiments respectively in the following parts.

### A. PASCAL VOC

In this experiment, we follow [33] by using the same settings and hperparameters. For SSD + mix-training , we reload the fully-trained SSD model(mAP=0.772) and do the finetune on the union of PASCAL VOC 2007 *trainval* and 2012 trainval set [13], [39]. We set SGD as the optimizer and the initial learning rate at 0.004, momentum at 0.9, set epoch as 400 and weight decay at 0.0005 and batch size as 32. We set $\gamma$ as 1 and $\alpha$ as 0.1. We used a strategy called warm restart [40] to accelerate the training that gradually ramps up the learning rate from $10^{-6}$ to $0.004$ at first 5 epoch. After the warm-up phase, the learning rate go back to $10^{-6}$ until 200 epoch, and keep it in the following epochs. We trained the model for 7.5 hours totally, and reached the best model at 320 epoch. For DSSD and YOLOv2, the settings are almost same as SSD. For RFB + mix-training , where we get our best result, we also finetune the trained model, and use the similar strategy and parameters as above. Almost settings follow [34]. We set SGD as the optimizer and the initial learning rate at 0.004, momentum at 0.9, set epoch as 400. We set the batch size at 32, weight decay at 0.0005 and epoch at 500. We also use the warm-up strategy that gradually ramps up the learning rate from $10-e6$ to $4-e3$ at first 15 epoch. After the warm-up phase, the learning rate go back to $10^{-6}$ until 250 epoch, and keep it in the following epochs. We reached the best model at around 390 epoch.

As shown in Tabel.I, we can see the comparison between the networks with and without mix-training on the VOC2007 *test set*. SSD* is the updated SSD results with more data augmentation. [33] For fair comparison, we re-implement SSD* with Pytorch-0.4 and CUDA9.0 and apply our method in the same environment. We also use the same data augmentation method in [33]. By using mix-training method, SSD* is greatly improved by $1.3\%$. DSSD and YOLOv2 also are upgraded by $0.8\%$ and $0.6\%$. For state-of-the-art network 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 Tabel.II. 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.

TABLE I
DETECTION RESULTS ON PASCAL VOC 2007 DATASET

| Method | Backbone | Data | mAP |
|---|---|---|---|
| SSD* [33] | Vgg | 07+12 | 77.2 |
| SSD* + MT | Vgg | 07+12 | **78.5** |
| DSSD [41] | Vgg | 07+12 | 78.6 |
| DSSD + MT | Vgg | 07+12 | **79.4** |
| YOLOv2 544 [31] | Darknet | 07+12 | 78.6 |
| YOLOv2 544+MT | Darknet | 07+12 | **79.2** |
| RFB300 [34] | Vgg | 07+12 | 80.5 |
| (Ours) MTNet | Vgg | 07+12 | **80.9** |
| RFB512 [34] | Vgg | 07+12 | 82.2 |
| (Ours) MTNet512 | Vgg | 07+12 | **82.5** |

TABLE II
DETECTION RESULTS ON PASCAL VOC 2012

| Method | Backbone | Data | mAP |
|---|---|---|---|
| SSD* [33] | Vgg | 07++12 | 75.8 |
| SSD*+MT | Vgg | 07++12 | **76.9** |
| YOLOv2 [31] | Darknet | 07++12 | 73.4 |
| YOLOv2+MT | Darknet | 07++12 | **74.0** |
| RFB512 [34] | Vgg | 07++12 | 81.2 |
| MTNet512 | Vgg | 07++12 | **81.4** |

* *07++12* denotes *trainval2007*, *test2007*, *trainval2012*

### B. MS COCO

In this experiment, the hyper parameters are same as previous literature [34] on COCO.

In previous literature, the basic learning rate is set to 0.002, and max epoch to 300. We train our network with *trainval35k* that is also used in previous networks. The No.1 one-stage detection network from [34] on COCO is RFB512-E, hence we also compare MTNet with RFB512-E in this experiment.

As shown in Table.III, Our MTNet achieves the state-of-the-art performance in real-time processing, with a great improvement to RFBNet300 and RFB512-E by $0.8\%$ and $0.3\%$ 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.

### C. Ablation Experiments

Similar as the popular ablation experiments [34], [41], [45], in order to better understand the proposed network, we investigate the effect of each component of *MixLoss* and compare it with [33].

Firstly, we set the network by only applying our method to localization part. For the part of localization, we apply the mix-training to the progress of localization predicting, by adding the *MixLoss* to the tail of localization part. For the part of classification, as the input images are mixed before training, we keep the random parameter $\lambda$ greater than 0.5 to make sure the first image is the main part and calculate the loss

TABLE III
DETECTION RESULTS ON MS COCO

| Method | Backbone | Time | Avg. 0.5:0.95 | Precision, 0.5 | IoU: 0.75 | Avg. S | Precision. M | Area L |
|---|---|---|---|---|---|---|---|---|
| Faster [29] | VGG | 147ms | 24.2 | 45.3 | 23.5 | 7.7 | 26.4 | 37.1 |
| Faster+++ [23] | ResNet-101 | 3.36s | 34.9 | 55.7 | 37.4 | 15.6 | 38.7 | 50.9 |
| Faster w FPN [42] | ResNet-101-FPN | 240ms | 36.2 | 59.1 | 39.0 | 18.2 | 39.0 | 48.2 |
| R-FCN [43] | ResNet-101 | 110ms | 29.9 | 51.9 | - | 10.8 | 32.8 | 45.0 |
| R-FCN w Deformable CNN [44] | ResNet-101 | 125ms | 34.5 | 55.0 | - | 14.0 | 37.7 | 50.3 |
| Mask R-CNN [45] | ResNext-101-FPN | 210ms | 37.1 | 60.0 | 39.4 | 16.9 | 39.9 | 53.5 |
| YOLOv2 [31] | darknet | 25 ms | 21.6 | 44.0 | 19.2 | 5.0 | 22.4 | 35.5 |
| SSD300* [33] | VGG | 12 ms | 25.1 | 43.1 | 25.8 | - | - | - |
| SSD512* [33] | VGG | 28 ms | 28.8 | 48.5 | 30.3 | - | - | - |
| DSSD513 [41] | ResNet-101 | 182ms | 33.2 | 53.3 | 35.2 | 13.0 | 35.4 | 51.1 |
| RetinaNet500 [46] | ResNet-101-FPN | 90 ms | 34.4 | 53.1 | 36.8 | 14.7 | 38.5 | 49.1 |
| RetinaNet800 [46] | ResNet-101-FPN | 198 ms | 39.1 | 59.1 | 42.3 | 21.8 | 42.7 | 50.2 |
| RFBNet300 [1][34] | VGG | 15 ms | 30.3 | 49.3 | 31.8 | 11.8 | 31.9 | 45.9 |
| RFBNet512-E [2][34] | VGG | 33 ms | 34.2 | 54.7 | 36.1 | 17.6 | 37.0 | 47.6 |
| MTNet300 | VGG | **15 ms** | **31.1** | 50.2 | 32.7 | 12.7 | 33.7 | 48.6 |
| MTNet512 | VGG | **35 ms** | **34.6** | 55.8 | 36.5 | 18.4 | 37.5 | 48.6 |

statistics are from [34]
[1,2] are re-implemented on single 1080ti because [34] used Titan which are not widely available on consumption-level platform.

TABLE IV
ABLATION ANALYSIS

| Method | class | bbox | mAP |
|---|---|---|---|
| SSD+MT | ✓ | | 78.0 |
| SSD+MT | | ✓ | 77.9 |
| SSD+MT | ✓ | ✓ | **78.5** |

TABLE V
COMPARISON WITH LABEL SMOOTHING

| Method | Backbone | Data | mAP |
|---|---|---|---|
| SSD [33] | Vgg | 07+12 | 74.3 |
| SSD* [33] | Vgg | 07+12 | 77.2 |
| SSD*+LM | Vgg | 07+12 | 77.5 |
| SSD*+MT | Vgg | 07+12 | **78.5** |

with it. The results show that the method actually improve the performance on regression task.

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

The complete one is done in previous experiment. We get the final results shown in Table.IV

The results shows that *MixLoss* in both of components contributes to the improvement on performance for object detection. Combination of them achieves the best result.

### D. Comparison with label smoothing

We also compare our method with label smoothing on SSD [33]. SSD is the network with several intuitive augmentation methods, and SSD* is the one with extra augmentation methods [33]. For SSD* + LM(label smoothing), we soften the classification labels for each objects by set 0.9 and $0.1/20$ in which the previous value is 1 and 0 respectively. SSD* + MT(mix-training) is the one with mix-training. All the network are trained under the same environment and same hyher parameters.

We get the final results shown in Table.V. The mix-training is actually better than label smoothing.

## V. ANALYSIS ON MIX-TRAINING

Based on Section.IV, our method improves the performance on object detection network. In this section, we conduct a deep analysis on how this architecture get a better result.

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

- Our experiment compares all the weights between the original SSD network and the improved network with mix-training which are trained in previous experiments.
- As shown in Figure.5, the weights and biases are decreased by mix-training, which means it actually regularize the network.

Secondly, we analysis the final detection result to show how our network improve confidence on previous RFBNet as follow.

- As shown in Figure.4, in best case of the ski, the confidence of it grows 4x from less than 0.1 in RFB to 0.4 in MTNet. In the worst case of woman in green, the confidence of her varies a little from 0.96 in RFBNet to 0.94 in MTNet.
- Our MTNet tries to give more confidence to uncertain objects such as some small and illegible objects which are hard to be detected by previous methods. Our method has slight fluctuations on the high-confidence objects due to the effect of softening and this will not impact the final result.

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

- As shown in Figure.6, our MTNet increase the number of medium or overlap objects by $15.2\%$, which means that MTNet gives more correct detections.

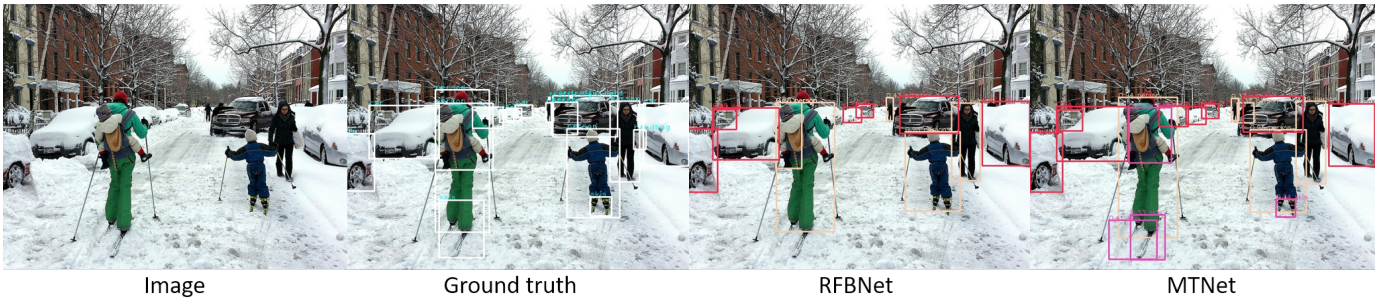
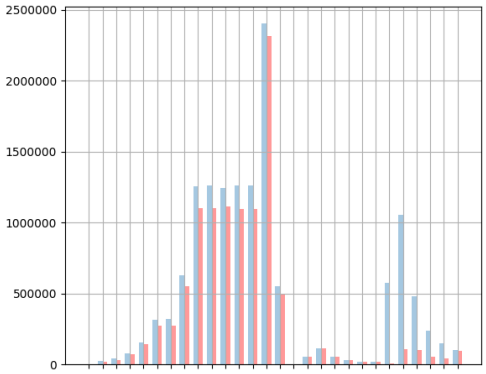
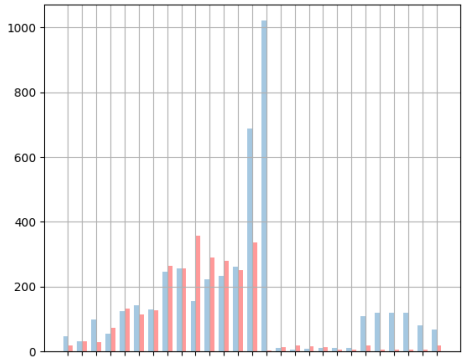



Fig. 4. Comparison between RFBNet and MTNet, MTNet performs better on low-confidence object, and give more detections on uncertain area.




(a) Comparison on weights




(b) Comparison on biases

Fig. 5. The comparison between SSD network with vs. without mix-training. Both are trained in Section.IV. The blue bars refer to the amount of weights or biases where the previous is greater than the latter. The pink bars refer to the amount of weights or biases where the latter is greater than the previous. The histogram shows that the weights become lower via mix-training. So the improved network gain more generalization.

- Benefit from high successful detectin rate, our MTNet gives more accurate predictions than the original network, which eventually leads to a decrease on regression loss.

## VI. Conclusion and Future Work

In this paper, we propose a novel Mix-Training Network for fast and accurate object detection. A pairwise operation is presented to mix two images in addition to intuitive operations.

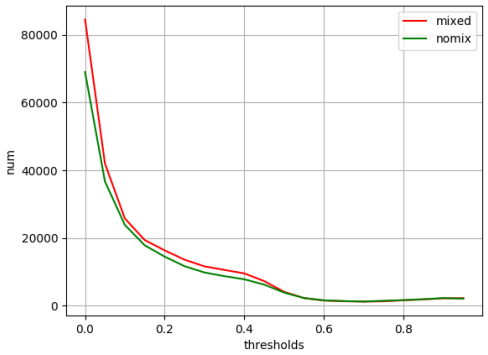



Fig. 6. The number of detected objects of SSD with mix-training and not. The results are obtained from PASCAL VOC 2007 test dataset by MTNet.

We design a hybrid loss function to achieve the effect of softening. In this way, our method is able to regularize the networks to improve the generalization. The proposed MTNet outperforms state-of-the-art networks.

In future work, we will continue the research on following problems but not limited. (1) If hyperparameters of training process is improper, the mix-training process may step into a training dilemma and the network may be traped into a bad local minimum. (2) Especially if one important hyper-parameters $\lambda$ in our MTNet is improper, the training may becomes unstable. (3) Another annoying problem is that the training result from scratch is worse than that from finetuning pre-trained model. Besides the detection task, we will also extend our approach in other areas of computer science and application, such as CAD/Graphics/CSCW [47]–[54].

## References

[1] S. R. Arashloo and J. Kittler, "Dynamic texture recognition using multiscale binarized statistical image features," *IEEE Transactions on Multimedia*, vol. 16, no. 8, pp. 2099–2109, 2014. 1

[2] Y. Zhu, J. Zhu, and R. Zhang, "Contextual object detection with spatial context prototypes," *IEEE Transactions on Multimedia*, vol. 16, no. 6, pp. 1585–1596, 2014. 1

[3] H. Yu, F. He, and Y. Pan, "A novel segmentation model for medical images with intensity inhomogeneity based on adaptive perturbation," *Multimedia Tools and Applications*, 2018, doi: 10.1007/s11042-018-6735-5 1

[4] X. Chen, F. He, and H. Yu, "A matting method based on full feature coverage," *Multimedia Tools and Applications*, pp. 1–29, 2018, doi: 10.1007/s11042-018-6690-1 1

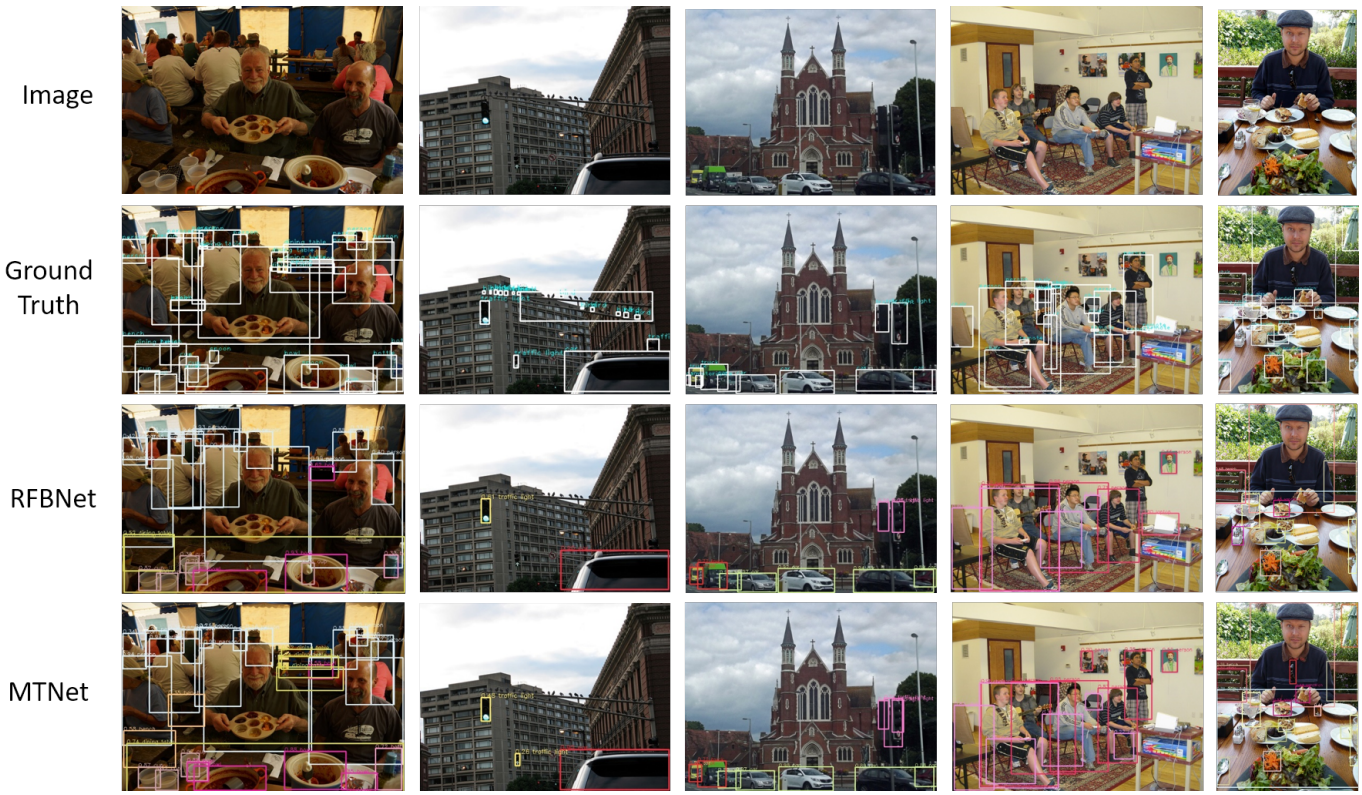



Fig. 7. More examples of comparison. Pictures are selected from MS COCO dataset.


[5] K. Li, F. He, H. Yu, and X. Chen, "A parallel and robust object tracking approach synthesizing adaptive bayesian learning and improved incremental subspace learning," *Frontiers of Computer Science*, 2018, doi: 10.1007/s11704-018-6442-4 1

[6] H. Yu, F. He, and Y. Pan, "A novel region-based active contour model via local patch similarity measure for image segmentation," *Multimedia Tools and Applications*, vol. 77, no. 18, pp. 24 097–24 119, 2018. 1

[7] K. Li, H. E. Fa-Zhi, H.-p. YU, and X. Chen, "A correlative classifiers approach based on particle filter and sample set for tracking occluded target," *Applied Mathematics-A Journal of Chinese Universities*, vol. 32, no. 2, pp. 294–312, 2017. 1

[8] S. Zhang, F. He, W. Ren, and W. Yao, "Joint learning of image detail and transmission map for single image dehazing," *The Visual Computer*, 2018, doi: 10.1007/s00371-018-1612-9 1

[9] K. Li, F. Z. He, and H. P. Yu, "Robust visual tracking based on convolutional features with illumination and occlusion handing," *Journal of Computer Science and Technology*, vol. 33, no. 1, pp. 223–236, 2018. 1

[10] J. Sun, H. E. Fa-Zhi, Y. L. Chen, and C. Xiao, "A multiple template approach for robust tracking of fast motion target," *Applied Mathematics:A Journal of Chinese Universities*, vol. 31, no. 2, pp. 177–197, 2016. 1

[11] B. Ni, F.-z. He, Y.-t. Pan, and Z.-y. Yuan, "Using shapes correlation for active contour segmentation of uterine fibroid ultrasound images in computer-aided therapy," *Applied Mathematics-A Journal of Chinese Universities*, vol. 31, no. 1, pp. 37–52, 2016. 1

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587. 1

[13] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010. 1, 5

[14] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448. 1

[15] S. Wang, J. Cheng, H. Liu, F. Wang, and H. Zhou, "Pedestrian detection via body-part semantic and contextual information with dnn," *IEEE Transactions on Multimedia*, 2018. 1

[16] L. Tian, M. Li, Y. Hao, J. Liu, G. Zhang, and Y. Q. Chen, "Robust 3d human detection in complex environments with depth camera," *IEEE Transactions on Multimedia*, 2018. 1

[17] Ç. Aytekin, H. Possegger, T. Mauthner, S. Kiranyaz, H. Bischof, and M. Gabbouj, "Spatiotemporal saliency estimation by spectral foreground detection," *IEEE Transactions on Multimedia*, vol. 20, no. 1, pp. 82–95, 2018. 1

[18] Z. Shao, W. Wu, Z. Wang, W. Du, and C. Li, "Seaships: A large-scale precisely annotated dataset for ship detection," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2593–2604, 2018. 1

[19] B. Poblete, J. Guzmán, J. Maldonado, and F. Tobar, "Robust detection of extreme events using twitter: Worldwide earthquake monitoring," *IEEE Transactions on Multimedia*, vol. 20, no. 10, pp. 2551–2561, 2018. 1

[20] V. Vapnik, *Statistical learning theory. 1998*. Wiley, New York, 1998, vol. 3. 1

[21] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016. 1

[22] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014. 1

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778. 1, 6

[24] P. Y. Simard, Y. A. LeCun, J. S. Denker, and B. Victorri, "Transformation invariance in pattern recognition—tangent distance and tangent propagation," in *Neural networks: tricks of the trade*. Springer, 1998, pp. 239–274. 1

[25] O. Chapelle, J. Weston, L. Bottou, and V. Vapnik, "Vicinal risk minimization," in *Advances in neural information processing systems*, 2001, pp. 416–422. 1

[26] T. Xiao, T. Xia, Y. Yang, C. Huang, and X. Wang, "Learning from massive noisy labeled data for image classification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 2691–2699. 1

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2818–2826. 1, 2

[28] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "mixup: Beyond

empirical risk minimization," *arXiv preprint arXiv:1710.09412*, 2017. 1, 2

[29] S. Ren, K. He, R. Girshick, and 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. 2, 6

[30] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2

[31] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," *arXiv preprint*, 2017. 2, 5, 6

[32] J. Redmon, S. Divvala, R. Girshick, and 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. 2

[33] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37. 2, 5, 6

[34] S. Liu, D. Huang, and a. Wang, "Receptive field block net for accurate and fast object detection," in *The European Conference on Computer Vision (ECCV)*, September 2018. 2, 5, 6

[35] C.-M. Teng, "A comparison of noise handling techniques." in *FLAIRS Conference*, 2001, pp. 269–273. 2

[36] X. Zhu and X. Wu, "Class noise vs. attribute noise: A quantitative study," *Artificial intelligence review*, vol. 22, no. 3, pp. 177–210, 2004. 2

[37] P. L. Bartlett, M. I. Jordan, and J. D. McAuliffe, "Convexity, classification, and risk bounds," *Journal of the American Statistical Association*, vol. 101, no. 473, pp. 138–156, 2006. 2

[38] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755. 5

[39] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results," http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 5

[40] I. Loshchilov and F. Hutter, "Sgdr: Stochastic gradient descent with warm restarts," *arXiv preprint arXiv:1608.03983*, 2016. 5

[41] C.-Y. Fu, W. Liu, A. Ranga, A. Tyagi, and A. C. Berg, "Dssd: Deconvolutional single shot detector," *arXiv preprint arXiv:1701.06659*, 2017. 5, 6

[42] T.-Y. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, "Feature pyramid networks for object detection." in *CVPR*, vol. 1, no. 2, 2017, p. 4. 6

[43] J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," in *Advances in neural information processing systems*, 2016, pp. 379–387. 6

[44] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei, "Deformable convolutional networks," *CoRR, abs/1703.06211*, vol. 1, no. 2, p. 3, 2017. 6

[45] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Computer Vision (ICCV), 2017 IEEE International Conference on*. IEEE, 2017, pp. 2980–2988. 5, 6

[46] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *IEEE transactions on pattern analysis and machine intelligence*, 2018. 6

[47] X. Yan, F. He, N. Hou, and H. Ai, "An efficient particle swarm optimization for large-scale hardware/software co-design system," *International Journal of Cooperative Information Systems*, vol. 27, no. 01, p. 1741001, 2018. 7

[48] X. Yan, F. He, and Y. L. Chen, "A novel hardware/software partitioning method based on position disturbed particle swarm optimization with invasive weed optimization," *Journal of Computer Science and Technology*, vol. 32, no. 2, pp. 340–355, 2017. 7

[49] D. Zhang, F. He, S. Han, and X. Li, "Quantitative optimization of interoperability during feature-based data exchange," *Integrated Computer-Aided Engineering*, vol. 23, no. 1, pp. 31–51, 2016. 7

[50] X. Lv, F. He, Y. Cheng, and W. Yiqi, "A novel crdt-based synchronization method for real-time collaborative cad systems," *Advanced Engineering Informatics*, vol. 38, pp. 381–391, 2018. 7

[51] X. Lv, F. He, W. Cai, and Y. Cheng, "Supporting selective undo of string-wise operations for collaborative editing systems," *Future Generation Computer Systems*, vol. 82, pp. 41–62, 2018. 7

[52] ——, "A string-wise crdt algorithm for smart and large-scale collaborative editing systems," *Advanced Engineering Informatics*, vol. 33, no. 3, pp. 397–409, 2017. 7

[53] Y. Chen, F. He, Y. Wu, and N. Hou, "A local start search algorithm to compute exact hausdorff distance for arbitrary point sets," *Pattern Recognition*, vol. 32, no. 2, pp. 340–355, 2017. 7

[54] Y. Wu, F. He, D. Zhang, and X. Li, "Service-oriented feature-based data exchange for cloud-based design and manufacturing," *IEEE Transactions on Services Computing*, vol. 11, no. 2, pp. 341–353, 2018. 7