Cascade Network Based Bolt Inspection In High-Speed Train

Xiaodong Gu[1*], Ji Ding[2]
[1] School of Mathematics and Information Technology, Jiangsu Second Normal University, Nanjing. 210013. P. R. China
[e-mail: guxiaodong@jssnu.edu.cn]
[2] School of Physics and Electronic Engineering, Jiangsu Second Normal University, Nanjing. 210013. P. R.China,
[e-mail: dingji0221@163.com]
*Corresponding author: Xiaodong Gu

Received September 4, 2020; revised March 3, 2021; revised March 30, 2021; accepted September 19, 2021; published October 31, 2021

Abstract

The detection of bolts is an important task in high-speed train inspection systems, and it is frequently performed to ensure the safety of trains. The difficulty of the vision-based bolt inspection system lies in small sample defect detection, which makes the end-to-end network ineffective. In this paper, the problem is resolved in two stages, which includes the detection network and cascaded classification networks. For small bolt detection, all bolts including defective bolts and normal bolts are put together for conducting annotation training, a new loss function and a new boundingbox selection based on the smallest axis-aligned convex set are proposed. These allow YOLOv3 network to obtain the accurate position and bounding box of the various bolts. The average precision has been greatly improved on PASCAL VOC, MS COCO and actual data set. After that, the Siamese network is employed for estimating the status of the bolts. Using the convolutional Siamese network, we are able to get strong results on few-shot classification. Extensive experiments and comparisons on actual data set show that the system outperforms state-of-the-art algorithms in bolt inspection.

Keywords: Few-shot learning, Siamese network, Small object detection, Non-maximum suppression, Convolutional neural networks

This work is partially supported by the National Natural Science Foundation of China. (No. 61701201)

http://doi.org/10.3837/tiis.2021.10.008 **ISSN : 1976-7277**

1. Introduction

Traditional train inspection is usually executed by the workers to detect. However, this method is costly, slow, and inefficient. A machine-vision approach is developed to automate inspection of various components of trains, including bolts, catenaries, pantographs, wheel rims, wheel sets and more. The vision inspection system inspects the various components of trains and estimates their status based on the data captured by the cameras or the laser devices. The system is of great significance to assess the quality of trains and prevent future accidents. Our research focuses on automatic localizing and estimating bolt defects based on computer vision technology.

Some previous work on visual detection of bolts has been reported. Yunguang Dou et al.[1] used a template matching method and a nearest neighbor classifier to determine the position of a bolt. Youngjin Cha et al.[2] utilized the Hough transform first, then, a linear support vector machine was applied to make a distinction between tight and loose bolts. F Marino et al.[3] used two discrete wavelet transforms to pre-process the images, and then employed two multi-layer perceptron classifiers. In [4], Lovedeep ramana et al. used the Viola-Jones algorithm to detect loosened bolt. In fastener inspection field, a probabilistic structure topic model (STM) [5] was employed. In [6], the pixel-wise and histogram similarities based approaches were utilized for detection. In [7], an Adaboost based algorithm was used for detection. In other target detection fields, [8] adopted a sparse representation to track dynamic overhead crane features effectively. In [9], the object-based human identification stage and spatial feature-based human identification stage were fused using the fuzzy holoentropy for accurate human identification. [10] proposed the algorithms to find randomly moving target in unknown environment. Compared with these traditional methods, deep learning based visual inspection systems[11-15] showed their superiority after a long period training with a large dataset. The comparative study of our method with the existing techniques is shown in **Table 1**.

The inspection robot runs on the track under the train. The camera on a mechanical arm captures images at a resolution of 2048*2000, but the size of the bolts is usually 30-40 pixels in the image. For visual detection in high-speed train, there are huge numbers of normal bolts (including many types of bolts, such as reverse mounting bolts, and hexagon bolts), but only a small number of defective bolts (shown in **Fig. 1**). Therefore, it is difficult for an end-to-end object detection networks to detect defective bolts directly. The difficulty of the vision-based bolt inspection system lies in the small sizes of the bolts, and a few training samples of the defective bolts. In this work, deep learning based two-stage method for visual detection of bolts in high-speed train is proposed, which includes the detection network and cascaded classification networks. For small bolt detection, we put defective bolts and normal bolts together for conducting annotation training. A new loss function based on the smallest axis-aligned convex set is proposed to improve the accuracy, and new boundingbox selection algorithm to get the maximum coverage of the bolts. These allow YOLOv3 network to obtain the accurate position and bounding box of the various bolts, regardless of whether they are damaged or missing. Second, each of the located bolts are sent into the trained Siamese network one-by-one. Finally, the Siamese network calculates the similarity metric between the input bolts and standard template, and estimates the status of the input bolts. To the best of our knowledge, the Siamese network is the first to be applied in bolt inspection. Both the detection and the classification network have a training and prediction stage. The training stage is completed by the server and the prediction stage is executed in the inspection computer. The main contributions of our cascade network are summarized as follows:

1) The problem of vision-based bolt inspection is resolved in two stages, which includes the detection network and cascaded classification networks.
2) A new loss function and a new boundingbox selection algorithm based on the smallest axis-aligned convex set are proposed. By incorporating these two improvements into YOLOv3, the average precision has been greatly improved on PASCAL VOC, MS COCO and actual data set.
3) The Siamese network is employed for bolt inspection. The predictive power of the convolutional Siamese network can be generalized to new defects that have never been seen before.

Table 1. Comparative study of our method with the existing techniques.

Methods			Interpretability	Detection Rate	Recognition Rate
Traditional methods	One-stage methods	STM [5]	Strong	Low	
		Pixel-wise and histogram similarities [6]		Low	
		Adaboost [7]		Low	
		Sparse Representation [8]		Low	
		Unknown target search in an unknown environment [10]		Low	
	Two-stage methods	Template matching + nearest neighbor classifier [1]	Strong	Low	Low
		Hough transform + SVM [2]		Low	Low
		Wavelet transforms + multi-layer perceptron [3]		Low	Low
		Viola-Jones + SVM [4]		Low	Low
		Viola-Jones + Bayesian network [9]		Low	Low
Deep-learning based methods	One-stage methods	Mask R-CNN [11]	Weak	Medium	
		CNN [12]		Medium	
		Deep Convolutional Neural Networks (DCNN) [13]		Medium	
		Multitask learning framework [14]		Medium	
	Two-stage methods	**The proposed method**	**Weak**	**High**	**High**
	Three-stage methods	SSD+YOLO+VGG [15]	Weak	High	Low
	Small sample learning	**The proposed method**	**Weak**	**High**	**High**

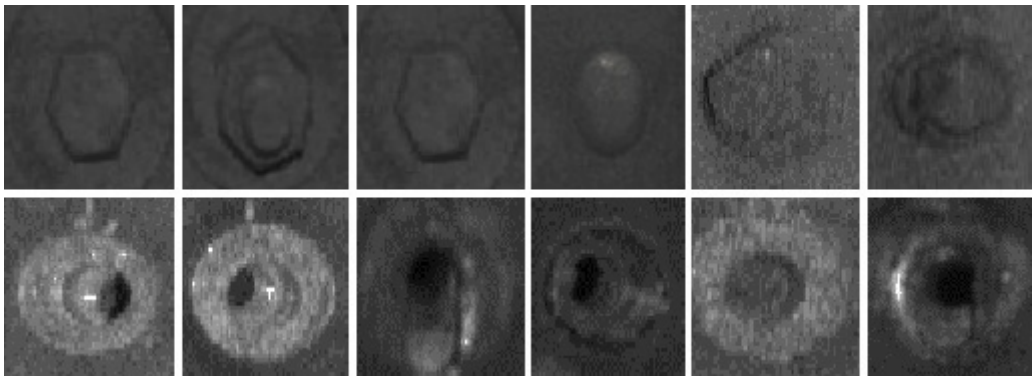

Fig. 1. Bolt images in CRH380A and CRH380AL High-Speed Trains data sets. The top row is normal bolts and the bottom row is defective bolts. The aforementioned data sets are collected from CRH380A and CRH380AL High-Speed Trains at ShangHai bullet train station. Which include 100,000 2D images and corresponding depth maps of train bottom.

The rest of this paper is organized as follows. In Section 2, the related works of object detection and few-shot learning are reviewed. In Section 3, the proposed cascade network, YOLOv3 network with a new loss function and a new boundingbox selection algorithm, and Siamese network for few-shot learning are explained. In Section 4, the experimental results are given. Finally, Section 5 concludes this paper.

2. Related Work

2.1 Object detection networks

Recently, deep learning are wildly used in many fields including digit recognition[16],optical character recognition[17], image classification[18-20], object detection[21] and tracking[40-42] and semantic image segmentation[22]. Generally speaking, object detection algorithms based on deep learning can be divided into two types: two-stage detectors (Fast R-CNN[23], Faster R-CNN[24], RFCN[25], MSCNN[26]) and one-stage detectors (SSD[27] and YOLO [28], RefineDet[29]). The speed of one-stage detector is better than that of two-stage detector, but the accuracy is slightly lower than that of two-stage detector. From the first YOLO[28], YOLOv2[30], the current YOLOv3[31] outperforms most detectors both in detection speed and accuracy. YOLOv3 applies a residual skip connection[32] and feature pyramid network[33] to get more meaningful semantic information from the up-sampling features and finer-grained information from the earlier feature maps. The prediction results are selected by Non-Maximum Suppression (NMS). These allow YOLOv3 to detect small object more accurately. In this work, we put defective bolts and normal bolts into one class for conducting annotation training, and detect bolts based on YOLOv3 with new loss function and new boundingbox selection algorithm.

2.2 Few-shot learning

Early studies about one or few-shot learning [34-35] focus on image fields and can be divided into three categories. First, model-based few-shot learning [36] offers the ability to encode and retrieve new information with new architecture. Second, metric-based few-shot learning [37] conducts classification by measuring the distance between the samples in a batch set and the

samples in a support set. Third, optimization-based few-shot learning [38] performs small sample learning by using new optimization. In the visual inspection of bolts in high-speed train, there are few defective bolts (including damaged and partially worn or missing bolts) because of the high manufacturing standard of the high-speed train. But a large number of sample pairs (including genuine sample pairs and imposter sample pairs) based on the metric can be obtained. In this work, we learn about image representations via a supervised metric-based approach with Siamese network. The network is able to learn information about object classes from one, or only a few, training samples, which uses many layers of non-linearities to capture invariance of transformation with the sample pairs from massive normal bolts and a few defective bolts. After transformation, all normal samples are clustered together and defective samples are placed far away from all normal samples.

3. Cascade Network for Visual Inspection of Bolts

3.1 cascade network

The proposed network for bolt detection is shown in **Fig. 2**. Cascade network includes the object detection and cascaded classification networks, which is a two-stage strategy. YOLOv3 network with a new loss function and a new boundingbox selection is used for detection, and Siamese network is for classification. Input images are first input into the detection network. Various bolts in the image are localized, despite the fact that they may be damaged or missing. Second, images from all of the located bolts are cropped and sent into the trained Siamese network one-by-one. Finally, the Siamese network estimates the status of the target bolts based on the similarity metric between the input bolt and the standard template.

Our algorithm can be extended to multi-component detection with the same framework. Learn from the concept of virtual objects [39], we treat bolts, catenaries, and pantographs in a general way. Different components are detected with the detection network in single inference, and sent to Siamese networks associated with the different components to estimate their status.

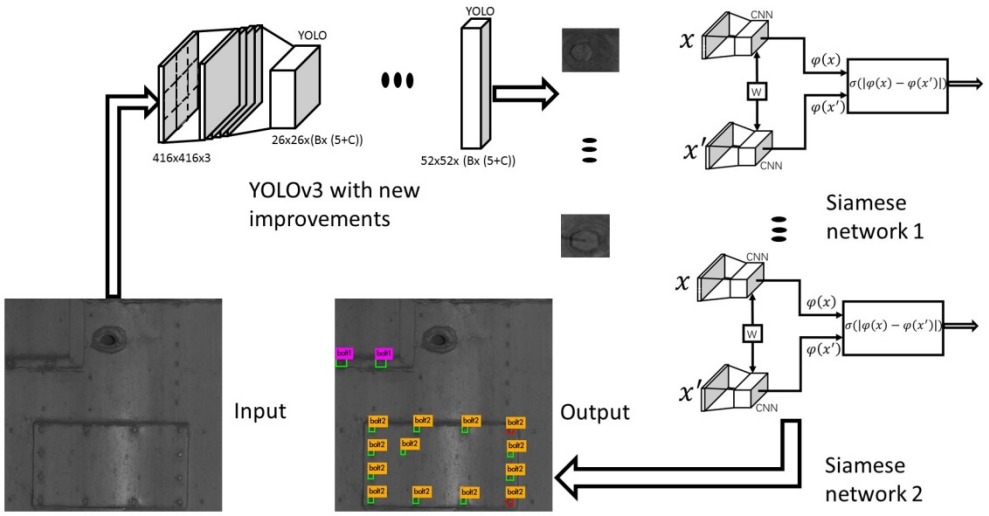

Fig. 2. The cascade network under two classes of bolts (reverse mounting bolts and hexagon bolts). The detected bolts are fed into two Siamese networks by class.

3.2 The smallest convex set based loss function and boundingbox selection algorithm for small object detection

Small object detection is difficult for target detection. This is because the images generated from smaller objects are low-resolution and contain less information than larger objects[43]. In our bolt detection system, each type of bolt should be detected in the detection step of the cascade network. Otherwise, undetected bolts will not be passed to the Siamese networks. To locate as many bolts as possible, we put defective bolts and normal bolts into one class for conducting annotation training.

YOLOv3 is suitable for small object detection task, whose boundingbox regression loss function is:

$$\mathrm{MSE}_{Loss} = \lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathrm{I}_{ij}^{obj}\left[\left(x_i-\hat{x}_i^j\right)^2+\left(y_i-\hat{y}_i^j\right)^2\right]+\lambda_{coord}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathrm{I}_{ij}^{obj}\left[\left(\sqrt{w_i^j}-\sqrt{\hat{w}_i^j}\right)^2+\left(\sqrt{h_i^j}-\sqrt{\hat{h}_i^j}\right)^2\right]$$

Where I_{ij}^{obj} denotes that the jth bounding box predictor in cell i is "responsible" for that prediction. The (x,y) coordinates represent the center of the box relative to the bounds of the grid cell and the (w,h) are the width and height of the box relative to the whole image. MSE loss is not quite consistent with accuracy improvement. To bridge the gap between the Intersection over Union (IoU) metric and the distance loss function that is used for boundingbox regression, Hamid Rezatofighi et al.[44] introduced the Generalized Intersection over Union (GIoU) as a new metric and a new loss function for boundingbox regression. GIoU focus on overlapping region of predicted boundingbox and groundtruth box, and non overlapping region of predicted boundingbox and groundtruth box. For small object detection, GIoU unable to reflect the extent to which the predicted boundingbox covers the groundtruth box. In this work, a new loss function $L_{convexIoU}$ for small object detection is proposed based on the smallest convex set enclosing predicted boundingbox and groundtruth box. $L_{convexIoU}$ focus on increasing the proportion of overlapping region to the smallest convex set and reducing the proportion of non overlapping region to the smallest convex set. The smallest axis-aligned convex set of predicted boundingbox and groundtruth box is shown in **Fig. 3**(a).

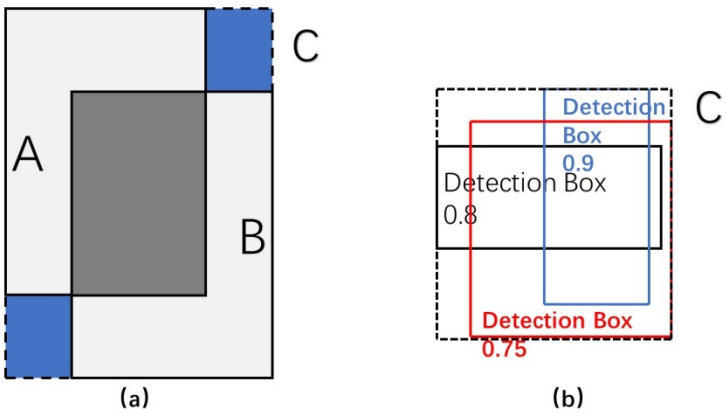

Fig. 3. The smallest axis-aligned convex set. (a) the predicted boundingbox A, the groundtruth box B and the smallest axis-aligned convex set C which encloses both A and B (C is shown in dashed line). The grey area is $A\cap B$, and the blue area is $C-\left(A\cup B\right)$. (b) all the detection boxes including the box Box_d with the maximum score and all other detection boxes with a significant overlap with Box_d, and their smallest axis-aligned convex set C (shown in dashed line).

For predicted boundingbox A and groundtruth box B, $A, B \subseteq S \in R^n$, C is the smallest convex set enclosing A and B. Both A and B are axis-aligned bounding box, therefore their smallest convex set $C \subseteq S \in R^n$ has rectangular shape. We calculate the difference between $A \cap B$ and $C - (A \cup B)$. Then we get the ratio between the aforementioned difference and the smallest axis-aligned convex set C, namely IoU based on the convex set $convexIoU$.

$$convexIoU = \frac{(A \cap B) + (A \cup B) - C}{C} \tag{1}$$

Finally, the loss function $L_{convexIoU} = 1 - convexIoU$. Similar to GIoU, $convexIoU$ is invariant to the scale, and $L_{convexIoU}$ satisfies non-negativity, identity of indiscernibles, symmetry and triangle inequality. Compare with GIoU, $convexIoU$ focus more on increasing the proportion of overlapping region to the smallest convex set and reducing the proportion of non overlapping region to the smallest convex set. Therefore, $convexIoU$ can reflect the extent to which the predicted boundingbox covers the groundtruth box. And

$$\forall A, B \subseteq S, \quad convexIoU(A, B) \le GIoU(A, B) \tag{2}$$

$$\forall A, B \subseteq S, \quad -1 \le convexIoU(A, B) \le 1 \tag{3}$$

Therefore, optimizing $convexIoU$ as loss, $L_{convexIoU}$ is suitable for small object detection. The calculation of $L_{convexIoU}$ is summarized in Alg.1.

For the bolts on High-Speed train, there are no occlusion between the bolts. Consequently, the predicted boundingbox with the maximum coverage is helpful for subsequent identification. But Non-Maximum Suppression (NMS) for selecting the best prediction boundingbox from all detection boxes is hard to get the maximizing coverage of the target. In YOLOv3 and other popular 2D object detectors, the score of the detection box is unrelated to position accuracy. To maximize coverage the object, our final prediction box is set as the smallest convex set which encloses the detection box Box_d with the maximum score and all other detection boxes with a significant overlap (we set the pre-defined threshold to 0.5 in this paper) with Box_d. The smallest convex set $C \subseteq S \in R^n$ has rectangular shape because all detection boxes are axis-aligned bounding box. The smallest axis-aligned convex set of the detection boxes is shown in **Fig. 3**(b) and the boundingbox selection algorithm based on the smallest convex set is shown in Alg. 2. Cooperating the $L_{convexIoU}$ and the boundingbox selection algorithm into YOLOv3 is called YOLOv3 with new improvements in this paper.

Alg 1. Loss function based on the smallest axis-aligned convex set of predicted box and ground truth box

Input: Predicted B^p and ground truth B^g bounding box coordinates:

$$B^p = (x_1^p, y_1^p, x_2^p, y_2^p), B^g = (x_1^g, y_1^g, x_2^g, y_2^g),$$

Output: $L_{convexIoU}$

1. For the predicted box, ensuring $x_2^p > x_1^p$ and $y_2^p > y_1^p$

$$\hat{x}_1^p = \min\left(x_1^p, x_2^p\right),\ \hat{x}_2^p = \max\left(x_1^p, x_2^p\right)$$
$$\hat{y}_1^p = \min\left(y_1^p, y_2^p\right), \hat{y}_2^p = \max\left(y_1^p, y_2^p\right)$$

2. Calculating area of B^g: $A^g = \left(x_2^g - x_1^g\right) \times \left(y_2^g - y_1^g\right)$.

3. Calculating area of B^p: $A^p = \left(x_2^p - x_1^p\right) \times \left(y_2^p - y_1^p\right)$.

4. Calculating intersection I between B^p and B^g:

$$\begin{matrix} x_1^I = \max(\hat{x}_1^p, x_1^g),\ x_2^I = \min(\hat{x}_2^p, x_2^g) \\ y_1^I = \max(\hat{y}_1^p, y_1^g), x_2^I = \min(\hat{y}_2^p, y_2^g) \end{matrix}$$

$$I = \begin{cases} \left(x_2^I - x_1^I\right) \times \left(y_2^I - y_1^I\right) & \text{if } x_2^I > x_1^I \text{ and } y_2^I > y_1^I \\ 0 & \text{otherwise} \end{cases}$$

5. Calculating area of the smallest convex set:

$$\begin{matrix} x_1^C = \min(\hat{x}_1^p, x_1^g),\ x_2^C = \max(\hat{x}_2^p, x_2^g) \\ y_1^C = \min(\hat{y}_1^p, y_1^g), y_2^C = \max(\hat{y}_2^p, y_2^g) \\ A^C = \left(x_2^C - x_1^C\right) \times \left(y_2^C - y_1^C\right) \end{matrix}$$

6. $convexIoU = \dfrac{A^p + A^g - A^C}{A^C}$

7. $L_{convexIoU} = 1 - convexIoU$

Alg 2. The boundingbox selection based on the smallest convex set of detection boxes

Input: coordinates of the detection boxes Box_d with the maximum score and all other detection boxes with a significant overlap with Box_d:

$$Box_d = \left(x_1^1, y_1^1, x_2^1, y_2^1\right), Box_d^2 = \left(x_1^2, y_1^2, x_2^2, y_2^2\right), Box_d^3 = \left(x_1^3, y_1^3, x_2^3, y_2^3\right)...$$

Output: coordinates of the predicted boundingbox: $B_p = \left(x_1^p, y_1^p, x_2^p, y_2^p\right)$

1. For all candidate detection boxes, ensuring $x_2^1 > x_1^1,\ y_2^1 > y_1^1; x_2^2 > x_1^2,\ y_2^2 > y_1^2; ...$

$$\hat{x}_1 = \min\left(x_1, x_2\right),\ \hat{x}_2 = \max\left(x_1, x_2\right)$$
$$\hat{y}_1 = \min\left(y_1, y_2\right), \hat{y}_2 = \max\left(y_1, y_2\right)$$

2.
$$\begin{matrix} x_1^p = \min(\hat{x}_1^1, \hat{x}_1^2, ...),\ y_1^p = \min(\hat{y}_1^1, \hat{y}_1^2, ...) \\ x_2^p = \max(\hat{x}_2^1, \hat{x}_2^2, ...), y_2^p = \min(\hat{y}_2^1, \hat{y}_2^2, ...) \end{matrix}$$

3.3 Ablation Study

In this section, we first evaluate the proposed method by incorporating $L_{convexIoU}$ losses into YOLOv3 algorithm on PASCAL VOC[45] and MS COCO[46] two data sets. The specific configurations of their training protocol and evaluation can be found in reference [45] and [46]. We use the original Darknet implementation of YOLOv3 framework and follow their parameters. The MSE loss function is replaced with L_{IoU} , L_{GIoU} and $L_{convexIoU}$ losses, and we use the new boundingbox selection algorithm. We train the network using each loss for 40K iterations with a batch size of 32. The models are trained with 4 Nvidia GTX 1080Ti GPUs and tested on the computer with an Intel Corei5-6600k, 32G RAM and GPU NIVIDIA GTX 1080Ti. The performance comparison has been reported in Table 2 and Table 3. We use AP ($(AP50 + AP55 + ... + AP90 + AP95)/10$, where $AP50$ means mAP with IoU threshold of 0.50 and $AP95$ means mAP with IoU threshold of 0.95) and AP75 (mAP with IoU threshold of 0.75) as performance measurement, i.e.,. For PASCAL VOC 2007 data sets, we report results using IoU and GIoU metric. For MS COCO 2018 data sets, we report results using IoU metric only.

As shown in **Table 2**, GIoU achieves performance with relative improvement of 3.47% AP and 5.56% AP75 using IoU as evaluation metric. While $L_{convexIoU}$ loss achieves performance with relative improvement of 5.20% AP and 7.81% AP75 using IoU as evaluation metric. And $L_{convexIoU}$ loss combined with new boundingbox selection brings improvements of 5.85% AP and 8.84% AP75. As shown in **Table 3**, $L_{convexIoU}$ loss achieves performance with relative improvement of 6.05% AP and 11.41% AP75 using IoU as evaluation metric. And $L_{convexIoU}$ loss combined with new boundingbox selection brings improvements of 9.87% AP and 12.61% AP75. **Table 2** and **Table 3** show the advantage of YOLOv3 with new improvements for target detection.

Table 2. Performance comparison of YOLOv3 trained by using MSE, L_{IoU} , L_{GIoU} and $L_{convexIoU}$ losses. The results are on the PASCAL VOC 2007 test set.

Loss/ Evaluation	**AP**		**AP75**	
	IoU	GIoU	IoU	GIoU
MSE	0.461	0.451	0.486	0.467
L_{IoU} Relative improve%	0.466 1.08%	0.460 1.99%	0.504 3.70%	0.498 6.64%
L_{GIoU} Relative improve%	0.477 3.47%	0.469 3.99%	0.513 5.56%	0.499 6.85%
$L_{convexIoU}$ Relative improve%	0.485 5.20%	0.477 5.76%	0.524 7.81%	0.515 10.25%
YOLOv3 with new improvements Relative improve%	0.488 **5.85%**	0.480 **6.43%**	0.529 **8.84%**	0.520 **11.34%**

Table 3. Performance comparison of YOLOv3 trained by using MSE, L_{IoU} , L_{GIoU} and $L_{convexIoU}$ losses. The results are on the MS COCO 2018 test set.

Loss/ Evaluation	AP	AP75
	IoU	IoU
MSE	0.314	0.333
L_{IoU} Relative improve%	0.321 2.18%	0.348 4.31%
L_{GIoU} Relative improve%	0.333 5.75%	0.362 8.01%
$L_{convexIoU}$ Relative improve%	0.340 6.05%	0.371 11.41%
YOLOv3 with new improvements Relative improve%	0.345 **9.87%**	0.375 **12.61%**

Secondly, we test the proposed method on CRH380A and CRH380AL High-Speed Trains data sets, which include a total of 100,000 images. We divide the whole set into training set, validation set and test set according to the ratio 6:2:2. We report the IoU metric in the following experiments. We replace MSE loss function with L_{GIoU} and $L_{convexIoU}$, and use new boundingbox selection algorithm. We test YOLOv3, YOLOv3 with GIoU, and YOLOv3 with new improvements in three situations. First, we conduct annotation training by classifying all bolts in one of four classes (channel bolt, baseplate bolt, reverse mounting bolt and hexagon bolt), and defective bolts are put into any class. The performance comparison is shown in **Table 4**. Then, all bolts are classified as reverse mounting bolts or hexagon bolts, and defective bolts are put into any class. The performance comparison of this phase is shown in **Table 5**. Finally, we place all the bolts (including defective bolts) into one class for conducting annotation training, and the performance comparison is shown in **Table 6**. The precision, recall and AP are used as performance measurement of object detection. As shown in **Table 4**, $L_{convexIoU}$ loss combined with new boundingbox selection algorithm achieves performance with relative improvement of 14.65% AP and 14.75% AP75 using IoU as evaluation metric. In **Table 5**, $L_{convexIoU}$ loss combined with new boundingbox selection algorithm achieves performance with relative improvement of 11.16% AP and 10.73% AP75 using IoU as evaluation metric. In **Table 6**, $L_{convexIoU}$ loss combined with new boundingbox selection algorithm achieves performance with relative improvement of 10.89% AP and 11.90% AP75 using IoU as evaluation metric. Compared with PASCAL VOC and MS COCO data sets, the CRH380A and CRH380AL High-Speed Trains data sets are specialized in small bolt data sets. The results show that YOLOv3 with new improvements is more suitable for small bolt detection. From **Table 4** through **Table 6**, we can see that the finer the classification, the lower the precision, recall and AP. This proves that training with emphasis on low-level category classification only cannot capture reliable information.

In the following experiments, we place all the bolts (including defective bolts) into one class for conducting annotation training, and we do not adjust regularization parameters between bounding box loss and classification loss. **Fig. 4** is accuracy (average IoU) against training iterations when YOLOv3 is trained using MSE loss as well as L_{GIoU} and $L_{convexIoU}$. **Fig. 4** shows that YOLOv3 with new improvements significantly improves the localization accuracy of the bolts. **Fig. 5** is the detection examples using YOLOv3, YOLOv3 with GIoU, and YOLOv3 with new improvements, one can see the detection boxes by YOLOv3 with new improvements are more accurate than that by YOLOv3 and YOLOv3 with GIoU.

Table 4. Performance comparison of YOLOv3 trained by using MSE, L_{IoU}, L_{GIoU} and $L_{convexIoU}$ losses. The results are on the CRH380A and CRH380AL High-Speed Trains data sets. All bolts are classified as four classes (channel bolt, baseplate bolt, reverse mounting bolt and hexagon bolt) for conducting annotation training, and defective bolts are put into any class.

Loss/ Evaluation	**AP** IoU	**AP75** IoU	**Precision**	**Recall**
MSE	0.389	0.393	96.25%	93.04%
L_{IoU}	0.408	0.415	96.86%	94.51%
Relative improve%	4.88%	5.59%		
L_{GIoU}	0.420	0.427	97.01%	95.99%
Relative improve%	7.96%	8.65%		
$L_{convexIoU}$	0.438	0.44	97.54%	96.50%
Relative improve%	12.5%	11.9%		
YOLOv3 with new improvements	**0.446**	**0.451**	**98.10%**	**97.75%**
Relative improve%	**14.65%**	**14.75%**		

Table 5. Performance comparison of YOLOv3 trained by using MSE, L_{IoU}, L_{GIoU} and $L_{convexIoU}$ losses. The results are on the CRH380A and CRH380AL High-Speed Trains data sets. All bolts are classified as reverse mounting bolts or hexagon bolts, and defective bolts are put into any class.

Loss/ Evaluation	**AP** IoU	**AP75** IoU	**Precision**	**Recall**
MSE	0.403	0.410	96.73%	93.80%
L_{IoU}	0.414	0.421	97.46%	96.63%
Relative improve%	2.72%	2.68%		
L_{GIoU}	0.425	0.430	97.85%	97.78%
Relative improve%	5.45%	4.87%		
$L_{convexIoU}$	0.439	0.442	98.12%	98.27%
Relative improve%	8.93%	7.80%		
YOLOv3 with new improvements	0.448	0.454	**98.90%**	**98.79%**
Relative improve%	**11.16%**	**10.73%**		

Table 6. Performance comparison of YOLOv3 trained by using MSE, L_{IoU} , L_{GIoU} and $L_{convexIoU}$ losses. The results are on the CRH380A and CRH380AL High-Speed Trains data sets. All bolts(including defective bolts) are placed into one class for conducting annotation training.

Loss/ Evaluation	AP	AP75	Precision	Recall
	IoU	IoU		
MSE	0.413	0.420	97.35%	94.56%
L_{IoU}	0.421	0.431	98.02%	96.54%
Relative improve%	1.93%	2.61%		
L_{GIoU}	0.429	0.440	98.60%	98.45%
Relative improve%	3.87%	4.76%		
$L_{convexIoU}$	0.441	0.451	99.07%	99.34%
Relative improve%	6.77%	7.38%		
YOLOv3 with new improvements	0.458	0.470	**99.40%**	**99.83%**
Relative improve%	**10.89%**	**11.90%**		

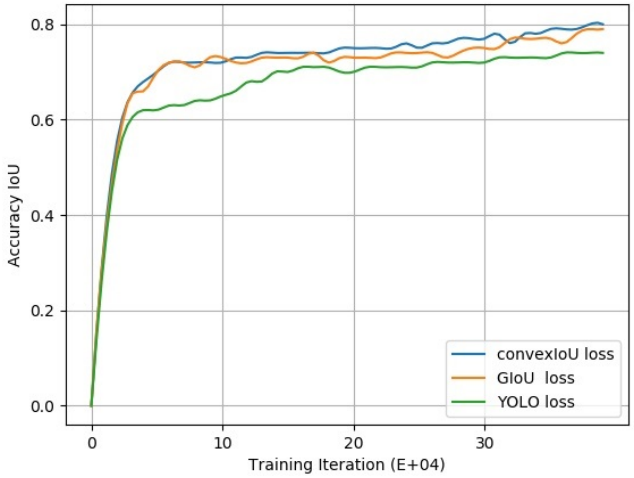

Fig. 4. The accuracy (average IoU) against training iterations when YOLOv3 was trained by using MSE loss, L_{GIoU} and $L_{convexIoU}$.

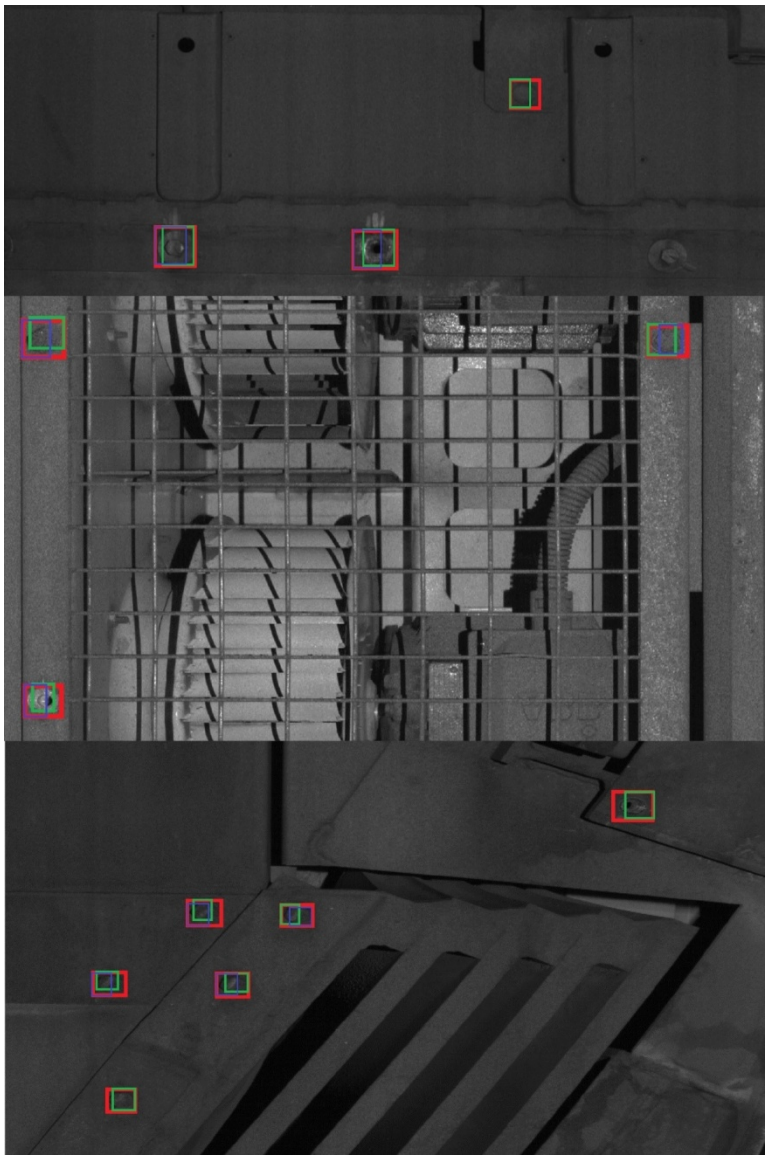

Fig. 5. The detection examples using YOLOv3(shown in blue line),YOLOv3 with GIoU(shown in green line), and YOLOv3 with new improvements(shown in red line).

3.4 Siamese network for few-shot learning

The architecture of Siamese networks [37] is shown in **Fig. 6**. Let x *and* x' be a pair of images, they are genuine pairs if the images x *and* x' belong to the same class (all positive or all negative samples) and impostor pairs otherwise. Let W be the shared parameters that are obtained by averaging the back-propagation calculated value from x *and* x' branch. Let $\varphi(x)$ and $\varphi(x')$ be the two points in the high-dimensional space that are generated by mapping x *and* x'. Let $|\varphi(x)-\varphi(x')|$ be the similarity measurement between x *and* x'. Therefore, $\sigma(|\varphi(x)-\varphi(x')|)$ is the output probability, where σ is sigmoidal activation

function. The regularized cross-entropy function for one pair x_i and x_i' is:

$$L(Y_i) = Y_i \log\left[\sigma\left(\left|\varphi(x_i) - \varphi(x_i')\right|\right)\right] + (1 - Y_i)\log\left[1 - \sigma\left(\left|\varphi(x_i) - \varphi(x_i')\right|\right)\right] + \lambda^T |W|^2 \quad (4)$$

Where,

$$Y_i = \begin{cases} 1 & \text{if } x_i \text{ and } x_i' \text{ are belong to the same class} \\ 0 & \text{otherwise.} \end{cases}$$

We fix a minibatch size of 128 with learning rate η_j , momentum μ_j , and L_2 regularization weights λ_j defined layer-wise. The updated rule at epoch n is as follows:

$$W_{kj}^{(n)}(x_i, x_i') = W_{kj}^{(n)} + \Delta W_{kj}^{(n)}(x_i, x_i') + 2\lambda_j^T \left|W_{kj}\right|^2 \quad (5)$$

$$\Delta W_{kj}^{(n)}(x_i, x_i') = -\eta_j \nabla W_{kj}^{(n)} + \mu_j \Delta W_{kj}^{(n-1)} \quad (6)$$

Where ∇W_{kj} is the partial derivative with respect to the weight W_{kj} between the jth neuron in some layer and the kth neuron in the successive layer. We used the same weight initialization and learning schedule as proposed by Gregory [37]. Once the convolutional Siamese network has been tuned, we can capitalize on powerful discriminative features to generalize the predictive power of the network to new samples.

We set x' branch as a standard template and x branch as a cropped bolt image. Both x *and* x' are normalized at 64×64. When the cropped bolts are sent into the trained Siamese network, their states are estimated based on the probability of the output similarity measurement. We used Pytorch implementation of Siamese networks. The model is trained on the computer with an Intel Corei5-6600k, 32G RAM and GPU NIVIDIA GTX 1080Ti.

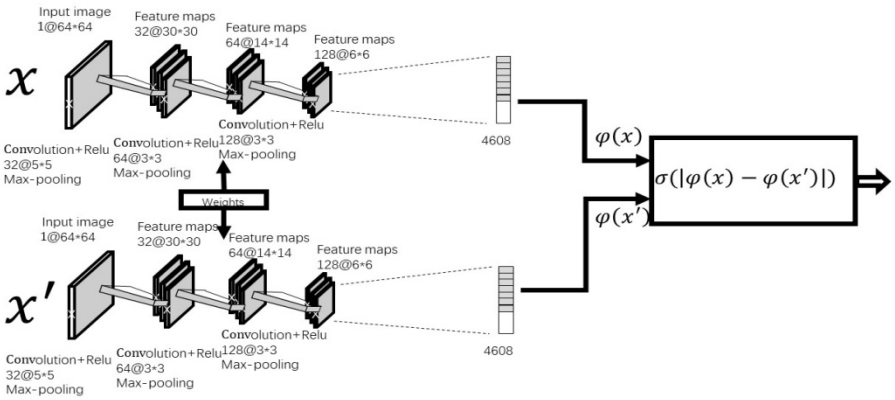

Fig. 6. The architecture of Siamese network.

4. Experimental Results

In the same training set, validation set and test set of the CRH380A and CRH380AL High-Speed Trains data sets, all bolts are classified in one class for conducting annotation training. The time complexity and the space complexity of the proposed cascade network are $Time \sim O(\sum_{l=1}^{D} M_l^2 \bullet K_l^2 \bullet C_{l-1} \bullet C_l)$ and $Space \sim O(\sum_{l=1}^{D} K_l^2 \bullet C_{l-1} \bullet C_l + \sum_{l=1}^{D} M_l^2 \bullet C_l)$, respectively. Where D is the depth of the cascade network including YOLOv3 and Siamese network. M_l^2 is the size of the feature map of the lth convolutional layer, and K_l^2 is the size of the kernel of the lth convolutional layer. C_{l-1} is the number of output channels of the $(l-1)$th convolutional layer, C_l is the number of output channels of the lth convolutional layer. The whole cascade network is lightweight network, and it runs at 20 FPS (frames per second) on the computer with an Intel Corei5-6600k, 32G RAM and GPU NIVIDIA GTX 1080Ti. We use accuracy, precision, recall, f1-score and ROC curve as the evaluation criteria of the proposed algorithm.

The performance comparison of different algorithms is shown in **Table 7**. The second and third column are end-to-end defect detection network, the accuracy, precision, recall and f1-score of classification are the same as that of detection. The results show that end-to-end network is not suitable for defect detection of bolts in High-speed train. The forth column is SSD+YOLO+VGG, and we use YOLOv3 with new improvements + VGG network in the fifth column. The last column is our proposed cascade network. **Table 7** shows the high reliability and precision of the proposed algorithm. The **Fig. 7** is ROC curves of different algorithms, which are created by plotting the true positive rate against false positive rate at various threshold settings. As shown in the **Table 7** and **Fig. 7**, two-stage methods and three-stage methods are better than one-stage methods, and our cascade network outperforms the other methods because of improvement of detection network and small sample learning network.

Table 7. Performance comparison of different algorithms.

Algorithms	**Mask R-CNN**	**YOLOv3 with new improvements**	**SSD +YOLO +VGG**	**YOLOv3 with new improvements + VGG**	**YOLOv3 with new improvements + Siamese**
FPS	40	40	15	20	20
Accuracy	50.01%	52.33%	79.03%	79.20%	**99.15%**
Precision	79.30%	81.79%	80.96%	81.16%	**98.61%**
Recall	28.94%.	30.50%	78.58%	78.82%	**99.76%**
F1-score	0.4240	0.4443	0.7975	0.7997	**0.9918**

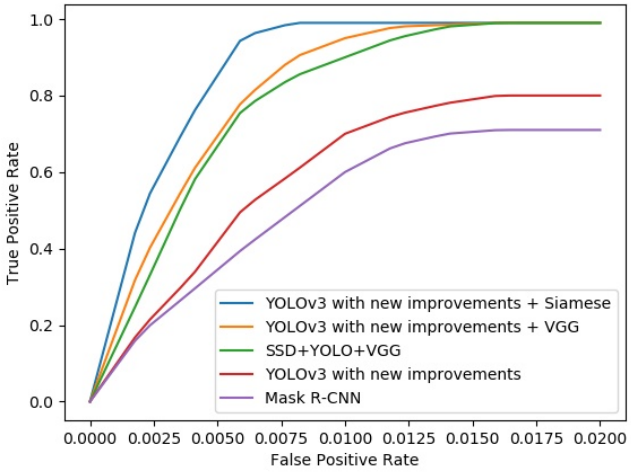

Fig. 7. ROC curves of different algorithms.

5. Conclusion

In this paper, cascade network based two-stage method for visual detection of the bolts in high-speed trains is proposed. By incorporating the new loss function and the new boundingbox selection algorithm into YOLOv3 framework, the detection network can obtain the position and the bounding box of the various bolts in any condition. The Siamese network is employed for few-shot learning. The output of the Siamese network is the probability of the output similarity measurement, and this measurement is used to determine the status of the detected bolt. The effectiveness of the cascade network based bolt detection system was verified at WuHan bullet train station. Overall, the proposed approach shows a promising application in bolt inspection. Nevertheless, the results suggest some further improvements.

1) In our bolt inspection system, detection stage is the premise of subsequent classification. Therefore, object detection based on RGB-D data and data augmentation should be tried to improve the detection capability.
2) We consider introducing attention mechanism to capture long-range dependencies and contextual information.
3) The edge computing of the proposed algorithm should be implemented to speed up the processing at the end.

Acknowledgments

This work is partially supported by the National Natural Science Foundation of China. (No. 61701201)

References

[1] Dou Y, Huang Y, Li Q, et al., “A fast template matching-based algorithm for railway bolts detection," *International Journal of Machine Learning and Cybernetics*, vol. 5(6), pp. 835-844, 2014. Article(CrossRef Link)

[2] Cha Y, You K, Choi W, et al., “Vision-based detection of loosened bolts using the Hough transform and support vector machines," *Automation in Construction*, vol. 71, pp. 181-188, 2016. Article(CrossRef Link)

[3] Marino F, Distante A, Mazzeo P L, et al., “A Real-Time Visual Inspection System for Railway Maintenance: Automatic Hexagonal-Headed Bolts Detection," *systems man and cybernetics*, vol. 37(3), pp. 418-428, 2007. Article(CrossRef Link)

[4] Ramana, L., Choi W., Cha, Y., “Fully automated vision-based loosened bolt detection using the Viola–Jones algorithm,” *Structural Health Monitoring-an International Journal*, vol. 18, pp. 422-434, 2019. Article(CrossRef Link)

[5] Feng H, Jiang Z, Xie F, et al., “Automatic Fastener Classification and Defect Detection in Vision-Based Railway Inspection Systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 63(4), pp. 877-888, 2014. Article(CrossRef Link)

[6] Aytekin C, Rezaeitabar Y, Dogru S, et al., “Railway Fastener Inspection by Real-Time Machine Vision," *systems man and cybernetics*, vol. 45(7), pp. 1101-1107, 2015. Article(CrossRef Link)

[7] Xia Y, Xie F, Jiang Z, et al., “Broken Railway Fastener Detection Based on Adaboost Algorithm," in *Proc. of international conference on optoelectronics and image processing*, pp. 313-316, 2010. Article(CrossRef Link)

[8] Wang T, Tan N, Zhang C, et al., “A Novel Sparse Representation Based Visual Tracking Method for Dynamic Overhead Cranes: Visual Tracking Method for Dynamic Overhead Cranes,” *International Journal of Ambient Computing and Intelligence*, vol.10, no.4, pp. 45-59, 2019. Article(CrossRef Link)

[9] Angadi, S., & Nandyal, S., “Human Identification System Based on Spatial and Temporal Features in the Video Surveillance System,” *International Journal of Ambient Computing and Intelligence*, vol. 11(3), pp. 1-21, 2020. Article(CrossRef Link)

[10] Mininath K. Nighot, Ashok Ghatol, Vilas M. Thakare, “Self-Organized Hybrid Wireless Sensor Network for Finding Randomly Moving Target in Unknown Environment,” *IJIMAI*, vol. 5(1) , pp. 16-28, 2018. Article(CrossRef Link)

[11] Chen, W., Li, Y., & Li, C., “A Visual Detection Method for Foreign Objects in Power Lines Based on Mask R-CNN,” *International Journal of Ambient Computing and Intelligence*, vol. 11(1), pp. 34-47, 2020. Article(CrossRef Link)

[12] Cha Y, Choi W, Buyukozturk O, et al., “Deep Learning-Based Crack Damage Detection Using Convolutional Neural Networks," *Computer-aided Civil and Infrastructure Engineering*, vol. 32(5), pp. 361-378, 2017. Article(CrossRef Link)

[13] Giben X, Patel V M, Chellappa R, et al., “Material classification and semantic segmentation of railway track images with deep convolutional neural networks," in *Proc. of international conference on image processing*, pp. 621-625, 2015. Article(CrossRef Link)

[14] Gibert, X.; Patel, V. M. Chellappa, R., “Deep Multitask Learning for Railway Track Inspection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, pp. 153-164, 2017. Article(CrossRef Link)

[15] Chen J, Liu Z, Wang H, et al., “Automatic Defect Detection of Fasteners on the Catenary Support Device Using Deep Convolutional Neural Network,” *IEEE Transactions on Instrumentation and Measurement*, vol. 67(2), pp. 257-269, 2018. Article(CrossRef Link)

[16] Lecun Y, Boser B E, Denker J S, et al., “Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1(4), pp. 541-551, 1989. Article(CrossRef Link)

[17] Lecun Y, Bottou L, B. Y., et al., “Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86(11), pp. 2278-2324, 1998. Article(CrossRef Link)

[18] Krizhevsky A, Sutskever I, Hinton G E, et al., “ImageNet Classification with Deep Convolutional Neural Networks," *Communications of the ACM*, Vol. 60(6), pp. 84–90, 2017. Article(CrossRef Link)
[19] Russakovsky O, Deng J, Su H, et al., “ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision*, vol. 115(3), pp. 211-252, 2015. Article(CrossRef Link)
[20] Szegedy C, Liu W, Jia Y, et al., “Going deeper with convolutions," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1-9, 2015. Article(CrossRef Link)
[21] Girshick R, Donahue J, Darrell T, et al., “Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 580-587, 2014. Article(CrossRef Link)
[22] Shelhamer, E.; Long, J. Darrell, T., “Fully Convolutional Networks for Semantic Segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 640-651, 2017. Article(CrossRef Link)
[23] Girshick R., “Fast R-CNN," in *Proc. of international conference on computer vision*, pp. 1440-1448, 2015. Article(CrossRef Link)
[24] Ren S, He K, Girshick R, et al., “Faster R-CNN: towards real-time object detection with region proposal networks,” *IEEE Trans Pattern Anal Mach Intell*, vol. 39(6), pp.1137-1149, 2017. Article(CrossRef Link)
[25] Dai J, Li Y, He K, et al., “R-FCN: Object Detection via Region-based Fully Convolutional Networks," in *Proc. of the 30th International Conference on Neural Information Processing Systems*, pp. 379–387, 2016. Article(CrossRef Link)
[26] Cai Z, Fan Q, Feris R S, et al., “A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection," in *Proc. of European conference on computer vision*, pp. 354-370, 2016. Article(CrossRef Link)
[27] Liu W, Anguelov D, Erhan D, et al., “SSD: Single Shot MultiBox Detector," in *Proc. of European conference on computer vision*, pp. 21-37, 2016. Article(CrossRef Link)
[28] Redmon J, Divvala S K, Girshick R, et al., “You Only Look Once: Unified, Real-Time Object Detection," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 779-788, 2016. Article(CrossRef Link)
[29] Zhang S, Wen L, Bian X, et al., “Single-Shot Refinement Neural Network for Object Detection," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4203-4212, 2018. Article(CrossRef Link)
[30] Redmon J, Farhadi A., “YOLO9000: Better, Faster, Stronger," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6517-6525, 2017. Article(CrossRef Link)
[31] Redmon J, Farhadi A., “YOLOv3: An Incremental Improvement," arXiv: *Computer Vision and Pattern Recognition*, 2018. Article(CrossRef Link)
[32] He K, Zhang X, Ren S, et al., “Deep Residual Learning for Image Recognition," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 770-778, 2016. Article(CrossRef Link)
[33] Lin T, Dollar P, Girshick R, et al., “Feature Pyramid Networks for Object Detection," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 936-944, 2017. Article(CrossRef Link)
[34] Fefei L, Fergus, Perona, et al., “A Bayesian approach to unsupervised one-shot learning of object categories," in *Proc. of international conference on computer vision*, pp. 1134-1141, 2003. Article(CrossRef Link)
[35] Feifei L, Fergus R, Perona P, et al., “One-shot learning of object categories," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28(4), pp. 594-611, 2006. Article(CrossRef Link)
[36] Munkhdalai T, Yu H., “Meta networks," in *Proc. of international conference on machine learning*, pp. 2554-2563, 2017.
[37] Gregory Koch Richard Zemel Salakhutdinov, R., “Siamese neural networks for one-shot image recognition," in *Proc. of the 32nd International Conference on Machine Learning*, 2015.

[38] Ravi S, Larochelle H., "Optimization as a Model for Few-Shot Learning," in *Proc. of international conference on learning representations*, 2017.
[39] Espada J P, Martinez O S, Garciabustelo B C, et al., "Virtual Objects on the Internet of Things," *International Journal of Interactive Multimedia and Artificial Intelligence*, vol.1(4), pp. 23-29, 2011. Article(CrossRef Link)
[40] Shuai Liu, Xinyu Liu, Shuai Wang, et al., "Fuzzy-Aided Solution for Out-of-View Challenge in Visual Tracking under IoT Assisted Complex Environment," *Neural Computing & Applications*, vol. 33, no. 4, pp. 1055-1065, 2021. Article(CrossRef Link)
[41] Shuai Liu, Dongye Liu, Khan Muhammad, et al., "Effective Template Update Mechanism in Visual Tracking with Background Clutter," *Neurocomputing*, vol. 458, pp. 615-625, 2021. Article(CrossRef Link)
[42] Shuai Liu, Shuai Wang, Xinyu Liu, et al., "Human Memory Update Strategy: A Multi-Layer Template Update Mechanism for Remote Visual Monitoring," *IEEE Transactions on Multimedia*, 23, pp. 2188-2198, 2021.Article(CrossRef Link)
[43] Kisantal M, Wojna Z, Murawski J, et al., "Augmentation for small object detection," arXiv: *Computer Vision and Pattern Recognition*, pp. 119-133, 2019. Article(CrossRef Link)
[44] Rezatofighi H, Tsoi N, Gwak J, et al., "Generalized Intersection Over Union: A Metric and a Loss for Bounding Box Regression," in *Proc. of 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 658-666, 2019. Article(CrossRef Link)
[45] M. Everingham, L. Van Gool, C. K. I.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. Article(CrossRef Link)
[46] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. of European conference on computer vision*, Springer, pp. 740–755, 2014. Article(CrossRef Link)

Xiaodong Gu received his PhD degree in DaLian university of Technology. After that he was a postdoctoral fellow in Institute of Automation, Chinese Academy of Sciences. He worked in National Space Science center Chinese academy of Science from 2003 to 2014. He is currently a professor at the Jiangsu second normal university. His research interests include object detection, visual tracking, image processing and embedded systems.

Ji Ding received his PhD degree in Nanjing University of Aeronautics and Astronautics. After that he was a postdoctoral fellow in HoHai University. He is an associate professor at the Jiangsu second normal university. His current research interests include image processing and embedded systems, passive microwave circuits design.