Vehicle wheel weld detection based on improved YOLO v4 algorithm

T.J. Liang [1,2], W.G. Pan [1,2], H. Bao [1,2], F. Pan [1,2]
[1] Beijing Key Laboratory of Information Service Engineering, Beijing Union University, Beijing, China;
[2] College of Robotics, Beijing Union University, Beijing, China

Abstract

In recent years, vision-based object detection has made great progress across different fields. For instance, in the field of automobile manufacturing, welding detection is a key step of weld inspection in wheel production. The automatic detection and positioning of welded parts on wheels can improve the efficiency of wheel hub production. At present, there are few deep learning based methods to detect vehicle wheel welds. In this paper, a method based on YOLO v4 algorithm is proposed to detect vehicle wheel welds. The main contributions of the proposed method are the use of k-means to optimize anchor box size, a Distance-IoU loss to optimize the loss function of YOLO v4, and non-maximum suppression using Distance-IoU to eliminate redundant candidate bounding boxes. These steps improve detection accuracy. The experiments show that the improved methods can achieve high accuracy in vehicle wheel weld detection (4.92 % points higher than the baseline model with respect to AP75 and 2.75 % points higher with respect to AP50). We also evaluated the proposed method on the public KITTI dataset. The detection results show the improved method's effectiveness.

Keywords: object detection, vehicle wheel weld, YOLO v4, DIoU.

Citation: Liang TJ, Pan WG, Bao H, Pan F. Vehicle wheel weld detection based on improved YOLO v4 algorithm. Computer Optics 2022; 46(2): 271-279. DOI: 10.18287/2412-6179-CO-887.

Acknowledgments: The work was funded by the National Natural Science Foundation of China (Nos. 61802019, 61932012, 61871039) and the Beijing Municipal Education Commission Science and Technology Program (Nos. KM201911417009, KM201911417003, KM201911417001). Beijing Union University Research and Innovation Projects for Postgraduates (No.YZ2020K001).

Introduction

Object detection [1] is a crucial part of computer vision and has become a popular research topic in theoretical and applied research in the past few years. Because of the continuous progress of machine-learning technology, it can be applied in many fields such as self-driving [2], security surveillance [3], nondestructive industrial inspection [4], aerospace [5], defect classification [6], and medical image processing [7]. In the field of industrial inspection, most factories detect vehicle wheel welds manually. The position of the weld must be adjusted manually, which is not beneficial and wastes time on the assembly line. Therefore, as a fast-progressing technology, automated object detection has developed more mature solutions that can meet the needs of factories and assembly line production modes.

There are two approaches to object detection [8]: those based on feature selection and those based on deep convolutional neural networks (CNNs) [9]. Feature selection is an important step in the object detection algorithm of traditional machine learning methods. The scale-invariant feature transform [10] and histogram of oriented gradients [11] are two features widely employed in support vector machines [12] for object detection. With the fast development of deep convolutional learning techniques, the emergence of object detection methods using CNNs [13] can greatly improve accuracy. In particular, end-to-end training realizes the overall optimization of performance and efficiency. There are two approaches in emerging CNN based detection methods: one-stage based methods, such as the YOLO series [14 – 17] and SSD [18], and two-stage detection methods, such as the region-based CNN series (R-CNN, Fast R-CNN, Faster R-CNN and Mask R-CNN) [19 – 22] or even multi-stage detection, such as the cascade R-CNN [23]. The first type of algorithm has a faster detection speed, and the second type of algorithm has a higher detection accuracy.

A typical object detector architecture consists of four parts – the input, backbone, head, and neck [17]. The backbone network is a pre-trained network used to extract features from the input images. The head predicts the class and location of the object. The aim of the neck is to improve robustness by collecting feature maps from intermediate stages.

In the industry [24], the applied object detection algorithms need to meet real-time requirements. In this paper, we optimize the one-stage based YOLO method to achieve fast and accurate detection of wheel welds. The organization of this paper is as follows. In Section 2, we introduce the research progress in recent years. The detail of the proposed method is presented in Section 3. Section 4 focuses on the implementation and a comparison with previous methods.

1. Related work

There is on region proposal step in the one-stage based detection method. This kind of detection method can generate the category probability and locations coordinate of the object directly, so it has a faster detection speed.

The first version YOLO was proposed in 2015, it further improving the recognition speed in the object detection field, because the single end to end convolutional network speeds up the recognition process. It makes the deep learning based object detection algorithm in real-time becomes possible. YOLO v1 uses L2 loss function in bounding box regression, which is applied to the regression of object detection bounding box in CNN. It defined as formula (1) shows:

$$L(x,\tilde{x}) = \sum_{i \in N} (x_i - \tilde{x}_i)^2 . \quad (1)$$

YOLO v2 improves convolutional neural network with anchor boxes, batch normalization, and multi-scale training. It also improves some deficiencies of YOLO v1, such as inaccurate positioning and lower recall rate than the region-based classification algorithms. YOLO v2 use Intersection-over-Union (IoU) and IoU loss function to improve the deficiency of L2 loss function which is sensitive to variant scales. The paper [25] use image pyramids in Densebox to compensate the defect of L2 loss function, which costs a lot of calculation.

IoU can compare the degree of overlap between any two graphics to determine the similarity, which was defined as formula (2) shows:

$$IoU = \frac{|A \cap B|}{|A \cup B|}, \quad (2)$$

where A is the ground-truth and B is the predicted box. IoU has invariant scale, this means the similarity of these figures is not affected by scale transformation, and it has non-negative; identity of indiscernible; symmetry; triangle inequality and other good characteristics. In this paper we uses IoU loss to improve the IoU metric.

$$Loss_{IoU} = 1 - IoU . \quad (3)$$

Redmon proposed the YOLO v3 algorithm with improvement over the previous version. It improves the prediction of bounding box by using dimension cluster as bounding box. The network uses logical regression to predict the objective score of each bounding box. Each classifier has a threshold, and classes with scores higher than the threshold are assigned to the bounding box. Secondly, the independent logic classifier replaces SoftMax to achieve better classification prediction. Each box predicts the possible classes of the bounding box by multi class classification. It uses the darknet-53 framework to increase the convolution level to 53, which is the basis of YOLO v3. The network is a hybrid of YOLO v2 and darknet-19, which consists of 3×3 and 1×1 filters with quick connection. The darknet-53 can predict at three different scales. It also adds several convolution layers from the basic feature extractor. At each scale, three bounding boxes are predicted. Starting from the early days of the network, this will eventually provide better functionality, and early computing will benefit the prediction of the last layer.

Each convolution part of darknet-53 uses the unique Darknet-Conv-2D structure. The network utilizes L2 regularization in the process of convolution; batch normalization and perform Leaky-ReLU after convolution. In original ReLU all negative values sets to zero, while in Leaky-ReLU all negative values have a non-zero slope value. It defines in mathematical terms as formula (4) shows:

$$y_i = \begin{cases} x_i & x_i \geq 0 \\ \dfrac{x_i}{a_i} & x_i < 0 \end{cases} . \quad (4)$$

The paper [26] proposed Generalized Intersection over Union (GIoU) to improve IoU and loss function with YOLO v3. If there is no overlap between any two shapes A and B, the IoU is 0, then IoU cannot distinguish whether the two shapes A and B are very close or very far away. GIoU is proposed by adding a penalty term as formula (5) shows.

$$GIoU = IoU - \frac{|C \setminus (A \cup B)|}{|C|}, \quad (5)$$

In the above formula C is the minimum closure area covering A and B. and the $GIoU$ loss is defined as formula (6) shows:

$$Loss_{GIoU} = 1 - GIoU . \quad (6)$$

The characteristic of $GIoU$ is to balance overlapping area and other non-overlapping areas at the same time, this feature can more accurately reflect the degree of overlap between the boxes. It also has non-negative; identity of indiscernible; symmetry; triangle inequality and other good characteristics.

When A and B are similar in shape:

$$\lim_{A \to B} GIoU(A,B) = IoU(A,B), \ -1 \leq GIoU(A,B) \leq 1. \quad (7)$$

When A and B completely overlap, $GIoU(A,B) = IoU(A,B) = 1$. When $(A \cup B)/C$ closes to zero, then the area of $A \cup B$ is very small relative to the area of C, $GIoU$ approaches negative one.

YOLO v4 is the latest version of the YOLO, which published in April 2020. YOLO v4 can reach a much higher mean average precision (mAP) [27] and FPS than the previous version YOLO v3 (10 points higher in AP and 12 points higher in FPS). It uses modified Resblock-body and CSPnet as the backbone feature extraction network. It also includes CSPDarknet53 [28] in YOLOv4. The detailed structure of the Resblock-body and CSPDenseNet are hown in fig. 1 and fig. 2. CSPDarknet53 adds the cross stage paritial (CSP) structure on the basis of Darknet53. The CSP structure directly concats part of the feature to the end of the block. This operation makes the gradient of each dense layer in the block no longer directly participate in the shallower gradient calculate. This can greatly reduce memory consumption and computing bottlenecks.

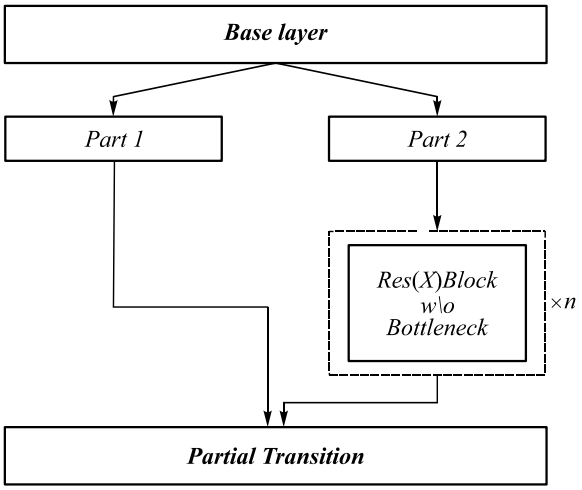

Fig. 1. Resblock-body

Furthermore, the activation function of Darknet-Conv-2D is modified from Leaky ReLU to Mish, and the convolution block is changed from Darknet-Conv-2D-BN-Leaky to Darknet-Conv2D-BN-Mish.The mish function could be defined as formula (8) shows:

$$Mish = x \times \tanh(\ln(1 + e^{x})). \quad (8)$$

From YOLO v1 to YOLO v4, each version has greatly promoted in speed and detection accuracy. The loss functions have been improved in every version such as L2 loss in YOLO v1, IoU and GIoU in YOLO v3. YOLO v4 use CSPDarknet53 as backbone, SPP to enlarge receptive field, PANet, greedy NMS and the head of YOLO v3. This kind of one-stage-based method has fast detection speed, how to improve detection accuracy is a difficult problem.

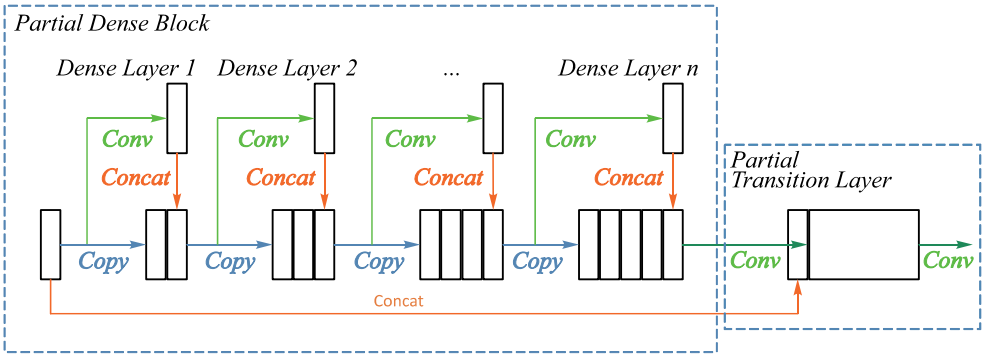

Fig. 2. Cross stage partial DenseNet (CSPDenseNet)

2. The proposed framework

In the one-stage based YOLO v4 object detection algorithm, features are extracted from the input images by the backbone network CSPDarknet53. Then, the head network is used for object detection and non-maximum suppression (NMS) to eliminate redundant bounding boxes. The improvements in the proposed method include anchor boxes preset accord to prior knowledge during training, optimization of the head part, and NMS methods to improve detection accuracy. The proposed method uses k-means to optimize the anchor box, the Distance-IoU (DIoU) loss to improve the loss function, and non-maximum suppression using Distance-IoU (DIoU-NMS) [29] to eliminate the redundant candidate bounding boxes for YOLO v4. The model structure is shown in fig. 3. The red part in the figure indicates the improvements we propose in this paper.

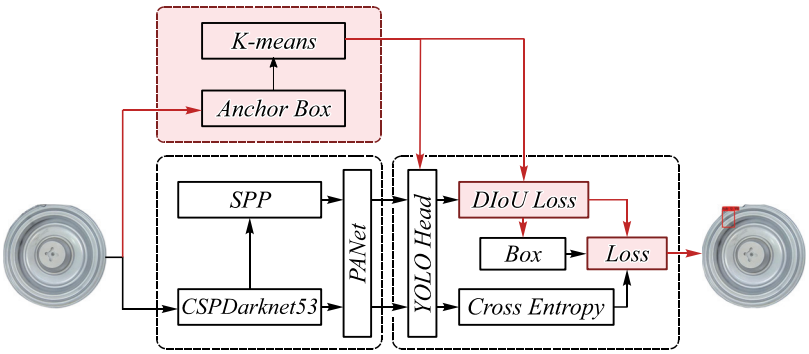

Fig. 3. Proposed YOLO structure

The k-means algorithm is a classic algorithm in the data mining field. To select the appropriate value k in the k-means clustering method, we need to trade-off the ideal overlap value and the training time. Increasing the value of k (number of centroids) will lead to an ideal overlap value, but it will also cause the computation to become positively correlated with the increase in the number of centroids, because of the increase in convolution filters. If a few selected centroids are used, the advantage is that the computation and training time are reduced, but the

disadvantage is that we cannot obtain a good overlap value. Hence, we adopt the elbow method to select the number of centroids. This method is suitable when k is small. When the k used is smaller than the expected value, every time k increases by 1, the corresponding value of the cost will decrease substantially; when the k used is larger than the expected value, every time k increases by 1, the corresponding value of the cost will not change obviously. The k obtained in this way is exactly at an inflection point, which is like an elbow. The distance threshold can be formulated in terms of the Euclidean distance or the Jaccard index, but the latter is more accurate because using the Euclidean distance will make large bounding boxes produce more error than small bounding boxes. The IoU is the ratio of the area of the anchors and ground truth boxes to the non-overlapping area. If the IoU is larger than 50 %, the anchor box is regarded as an object bounding box. Otherwise, it is regarded as ambiguous or unlearned.

In the k-means algorithm, the distances from all sample points to all centroids must be calculated in each iteration, which wastes a lot of time. The Elkan k-means method is used to reduce unnecessary distance calculations.

The distance is measured by calculating the IoU value between the borders to prevent the error caused by the size of the bounding box itself. The dataset is divided into k clusters by a clustering method. Through a series of iterations, the border distance within the clusters is made as small as possible, and the border distance between clusters is made as large as possible, and finally the size of the candidate box is determined by the change in the value of the function. Therefore, we select representative candidate boxes to replace the original candidate box size of the YOLO v4 detection algorithm to improve the detection performance by analyzing the size of the bounding boxes in the dataset.

The DIoU is closer to the bounding box regression mechanism than the GIoU, which is shown in fig. 4. It considers the distance between the target and the anchor, the overlap rate, and the scale, which makes the bounding box regression more stable and avoiding problems such as divergence during training, as in the IoU and GIoU. It also adds a penalty term based on the IoU, which is defined as:

$$DIoU = IoU - \frac{\rho^2(b,b^{gt})}{c^2}. \tag{9}$$

The b and b^{gt} represent the centers of boxes B and B^{gt}, respectively, $\rho()$ represents the Euclidean distance, and c is the length of the diagonal in the minimum closure area overlap of the two boxes.

The DIoU loss is defined as follows:

$$Loss_{DIoU} = 1 - IoU + \frac{\rho^2(b,b^{gt})}{c^2}. \tag{10}$$

Similar to the GIoU, the DIoU also provides a moving direction for the bounding box when it does not overlap with the target box. It directly minimizes the distance between two bounding boxes, which can converge much faster than the GIoU. For the situation in which the two boxes are in the horizontal direction or the vertical direction, the DIoU can perform the regression very quickly, whereas the GIoU almost degenerates into the IoU. Additionally, the DIoU can replace the traditional IoU evaluation strategy and apply it to NMS, which makes the results obtained by NMS more reasonable and effective.

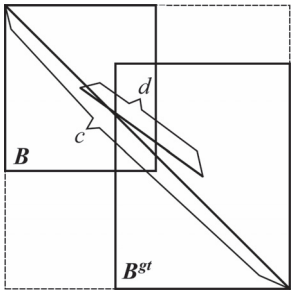

Fig. 4. The schematic diagram of DIoU

NMS is widely used in object detection algorithms. It can determine the size of the IoU to remove redundant candidate boxes and determine the best bounding box with the highest score, and an overlap rate exceeding the threshold plays a role in the final stage of object detection to obtain the best effect.

Soft-NMS [30] was proposed to address the shortcomings of NMS. This improved method can delete the bounding boxes that have an IoU larger than the present value and can also decrease the score. It improves the robustness of traditional NMS. The disadvantage is that there is no standard reference for the manual setting of the threshold, which is completely changed by experimental results and experience. Soft-NMS updates in the pruning step with the following rule:

$$s_i = \begin{cases} s_i, IoU(M,b_i) < N_t, \\ s_i(1 - IoU(M,b_i)), IoU(M,b_i) \geq N_t. \end{cases} \tag{11}$$

The above function decays the scores of detections above a threshold N_t as a linear function of the overlap with M. Therefore, bounding boxes that are close to M are assigned a greater penalty, and the penalties for those boxes that are far away from the M are very small or even equal to 0.

The pruning step has the following Gaussian penalty function:

$$s_i = s_i \exp\left(-\frac{IoU(M,b_i)^2}{\sigma}\right), \forall b_i \notin D. \tag{12}$$

Softer-NMS [31] proposed a weighted average strategy to optimize the threshold. GIoU-NMS [26] adds a penalty item to the IoU. The penalty term increases with the distance of the bounding box. When one box contains another box, GIoU degenerates into IoU and needs more iterations to converge. However, DIoU-NMS uses a penalty formula; this can minimize the center distance of two rectangular boxes. This method can take into account the overlap area

and center distance. If M is the highest score of the predicted box, the DIoU-NMS is defined as follows:

$$s_i = \begin{cases} s_i, & IoU(M, B_i) < \varepsilon \\ 0, & IoU(M, B_i) \geq \varepsilon \end{cases}, \quad (13)$$

in the above formula s_i is the classification score and ε is the threshold.

Anchor boxes are used extensively in one-stage object detection algorithms to set the initial dimensions of the bounding boxes, because they are better than other unsupervised learning algorithms that are biased toward bounding boxes with large dimensions. Using the feedback in the neural network, the initial dimension is corrected until it meets the ground truth dimension. The k-means clustering method can be used to provide feedback, which initializes the normalization (correction) process by taking k random boxes (centroids) as cluster heads. Clusters are then repeatedly assigned around the nearest centroid and updated based on a certain threshold value until convergence. The proposed network's backbone is shown in fig. 5.

To obtain appropriate IoU scores, which are independent of box size, the proposed method use the distance formula in YOLO v2, expressed as follows:

$$d(box, centroid) = 1 - IoU(box, centroid). \quad (14)$$

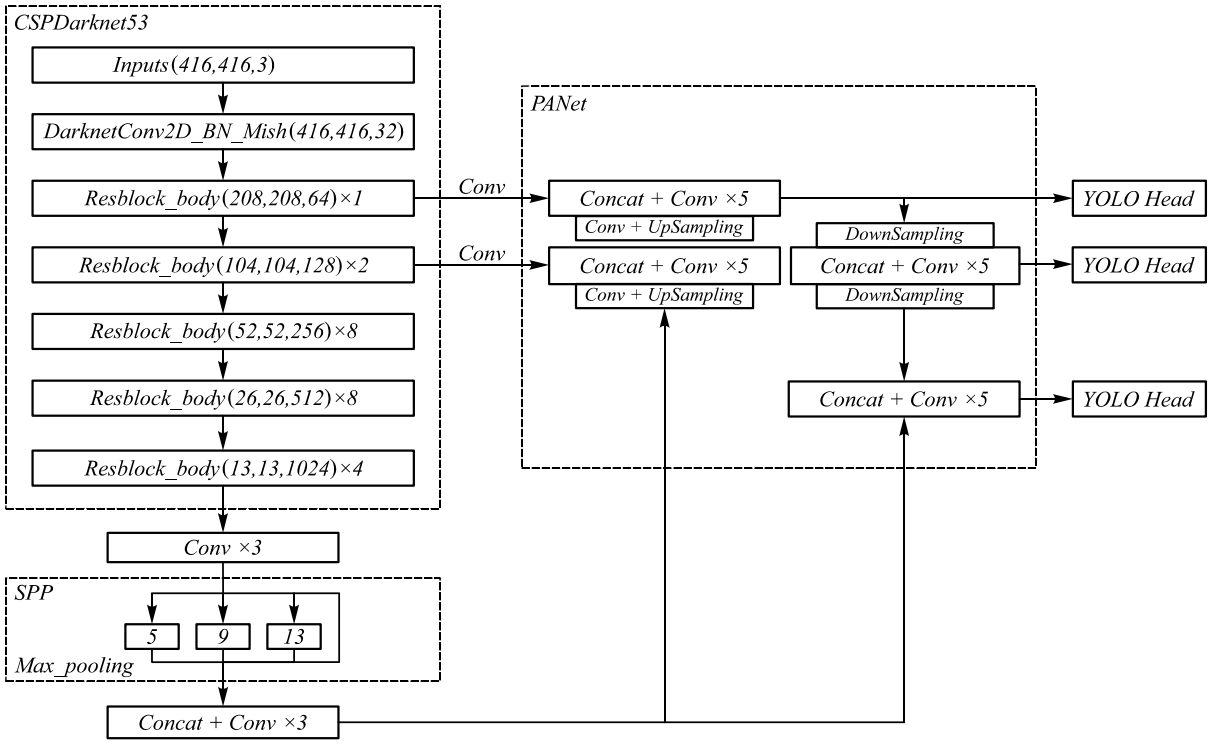

Fig. 5. Backbone structure

IoU is insensitive to the scale of the object because it is the ratio of two areas. However, bounding box regression and IoU optimization in the detection task are not completely equivalent. The *Ln* norm is sensitive to the scale of the object, and the IoU cannot directly optimize the part that does not overlap. The GIoU can solve the problem of disjoint real and detection boxes. However, when the real box and detection box are completely contained, GIoU degenerates to IoU, which has the same effect as IoU, and it cannot distinguish the relative positions between the real box and detection box.

DIoU uses normalized distances between the ground truth and bounding boxes. NMS is used to ensure that an object appearing in multiple boxes in the grid is calculated only once. Using IoU as the criterion to eliminate redundant detection boxes does not work in cases with object occlusion. The proposed method uses DIoU-NMS, which makes the system less susceptible to occlusion because it considers the centroid distance along with overlap area.

3. Dataset

The dataset used in this paper was collected from a real defect detection environment on a production line. The collected wheel images were annotated with the welding seams on each image. The dataset consists of 5300 images: 4240 images were randomly chosen for training the model. The remaining 1060 annotated images were used to test the model.

Each image has a corresponding annotation file, which gives the bounding box and class label of the objects. Because there is only one weld in the manufacturing process of an automobile wheel, each image has only one marked weld. To increase the diversity of the collected images, we rotated the wheel by 0.05° before each image was collected. We used a 30-frame industrial camera to collect video data and extracted key frames from the video to increase the number of annotated images in our dataset. fig. 6 shows the original image we collected. The

center point of each vehicle wheels coincides with the center point of the turntable. To simulate the effect of light, color, and position in a real environment, we did not perform any preprocessing on the images, including processing for brightness, contrast, or noise. Each image was 640 × 480 in size and has three channels. Sample images and annotated images from our dataset are shown in fig. 7.

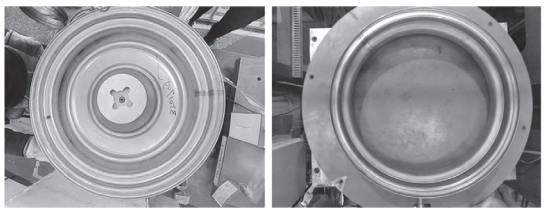

Fig. 6. The collected original image

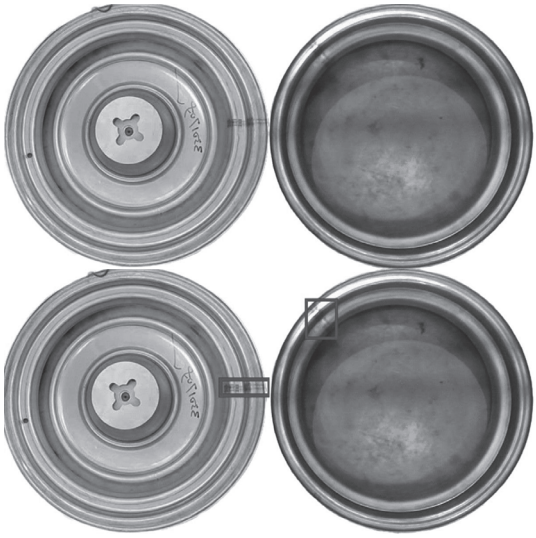

Fig. 7. Vehicle wheel weld dataset

The annotated XML file contains the image name, resolution size, object location, and category. It was first converted to the VOC format. Then, the corresponding images were moved to the target directory. The annotated file was converted to a text file suitable for the YOLO framework.

In this study, the KITTI 2D [32] object detection dataset is used to demonstrate that our proposed method is also effective on other public data sets. KITTI is currently one of the most important object detection datasets. It include a large number of traffic scene image data, which can evaluate the performance of the proposed method. This part of the KITTI dataset contains 7481 annotated images for training, 7518 annotated images for testing, and has 80.256 labels. It is commonly used in computer vision to evaluate detection algorithms.

4. Experimentations and results

The effectiveness of the proposed method based on YOLO v4 was evaluated on the vehicle wheel weld detection dataset, and the robustness of the proposed method was evaluated on the KITTI dataset.

The detailed parameters of the proposed method were as follows: the number of training steps was 8000; the initial learning rate was 0.013, which was multiplied by 0.14 at 6400 and 7200 steps; the momentum was 0.9; and the weight decay was 0.0005. The improved method proposed in this paper was run on a computer with a single graphics card for training. The initial size of the batches and mini-batches was 8. The default momentum was 0.949, and the loss normalizer was 0.07, which is the same as an early version of YOLO. The environment of the overall experiment was built using TensorFlow and Keras in Python. The computer configuration included one NVIDIA GeForce 2080Ti graphics card, with 11 GB VRAM.

The whole set of labeled data was clustered using k-means to obtain the appropriate anchor boxes. The experiment shows that the average IoU value was 87.30 % and 85.01 % at k = 12 for the vehicle wheel weld and KITTI datasets, respectively. As shown in Table 1, the increments in the value of k did not translate into meaningful results (as compared with 87.74 % and 85.35 % for k = 15). The final anchor boxes used in the experiments are shown in tab. 2. The vehicle wheel weld and KITTI datasets had 12 anchor boxes after clustering, each of which have different heights and widths, as shown in fig. 8.

Tab. 1. The average IoU in Different K-value

K-value	the average IoU value	
	vehicle wheel weld	KITTI
9	85.83 %	83.36 %
12	87.30 %	85.01 %
15	87.74 %	85.32 %

Tab. 2. Candidate box size after clustering

Detection Algorithm	The size of anchor boxes		
	large	medium	small
Our training set	(116.90)	(30.61)	(10.13)
	(156.198)	(62.45)	(16.30)
	(373.326)	(59.119)	(33.23)
	(250,138)	(93.35)	(25.78)
KITTI training set	(131.118)	(63.32)	(21.22)
	(92.116)	(62.56)	(32.24)
	(200.130)	(73.60)	(26.40)
	(316.181)	(106.57)	(45.40)

Fig. 8 illustrates the selected anchor boxes in the form of a rectangular box. This represents the object box in the vehicle wheel weld dataset accurately.

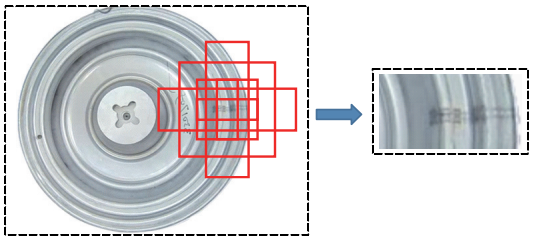

Fig. 8. Predicted anchor boxes

The loss curves of our improved method with respect to training epochs are shown in fig. 9. As the results show, compared to YOLO v3 and YOLO v4, our proposed method quickly converges during training on the vehicle wheel weld dataset.

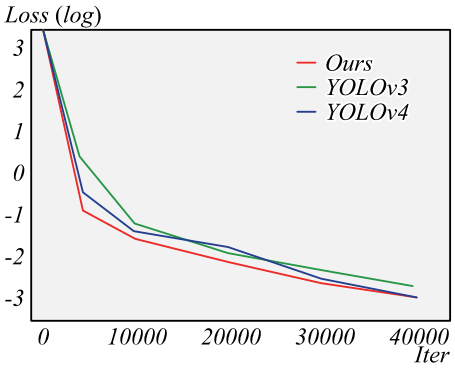

Fig. 9. Loss function

In the experiment, we use the average precision (AP) to compare different models and the accuracy of the different models. Both recall and precision are considered in the calculation of the AP, which takes the average value of the precision rate at each recall point from 0 to 1. Precision is the factor of the ratio of the original objects that were accurately detected, and recall is the proportion of the labeled objects in the image that were detected correctly.

Fig. 10 shows the detection results on the vehicle wheel weld dataset obtained by the proposed improved one-stage method presented in our work. The left part of the figure is the original input vehicle wheel weld image, and the right part is the detection result, which contains the detected bounding boxes overlaid on the images. It can be clearly seen that the improved method proposed in this paper can effectively detect the weld location in the image.

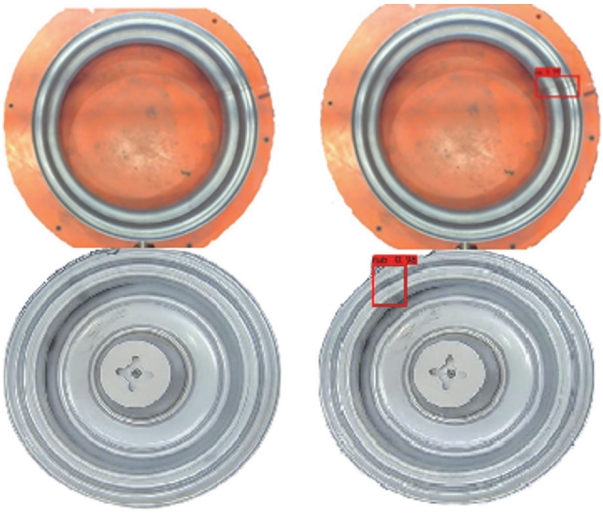

Fig. 10. The weld detection results

The detailed comparative results on the vehicle wheel dataset obtained by our proposed method, the original YOLO v4, and YOLO v3 are listed in Table 3. It lists the detection results based on YOLO v3 and YOLO v4, the improvement obtained by YOLO v4 with respect to YOLO v3, and the improvement obtained by the method proposed in this paper with respect to YOLO v3. The results show that the method proposed in this paper obtains a greater improvement (5.45 % in AP75 and 3.42 % in AP50) than YOLOv3. Moreover, it has a certain degree of improvement (4.92 % in AP75 and 2.76 % in AP50) compared to YOLOv4. Fig. 11 illustrates the accuracies of our method, YOLO v3, and YOLO v4 (as AP50 and AP75) during the process of training on the vehicle wheel weld dataset. The results show that our method quickly achieves a higher accuracy rate during the training process and obtains a higher accuracy rate at the end of the training.

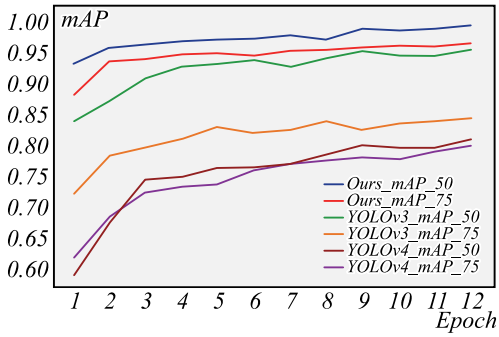

Fig. 11. Training epoch and mAP

Tab. 3. Comparisons of different network in vehicle wheel weld data set

Algorithm	AP75	AP50
YOLO v3	78.91	94.83
YOLO v4	79.44	95.49
Ours	84.36	98.25

To evaluate the effectiveness of the proposed improved method in this paper, we used the KITTI dataset to test our improved method. The detection results of different networks that have been tested on the KITTI dataset are listed in tab. 4. Here, we take the van, pedestrian, and cyclist classes in the KITTI 2D object detection dataset as examples to compare the performances. These comparisons show that the method proposed in this paper has different levels of improvement, and the total mAP is increased by 2.67 %. This detailed comparison result fully verifies our improved method's effectiveness.

Conclusions

We proposed an improved method based on the one-stage object detection algorithm YOLO v4 in this paper, which can detect a vehicle wheel weld in a image. The final results show our proposed method achieved results with an AP75 of 84.36 % (4.92 % point higher than that of the baseline model) and an AP50 of 98.25 % (2.76 % points higher than that of the baseline model) on the vehicle wheel weld dataset, which indicates that our method can perform better on the defect detection task, with higher accuracy and better stability. The improved method enables real-time computation on a single GPU. The results on the KITTI dataset also verify our improved

proposed method's effectiveness. This method can be applied in industrial production to detect vehicle wheel welds, which takes less time and labor than the manual method. This method can also be applied in other fields, where real-time detection and high accuracy are needed. However, the limitation of this method is that the vehicle wheels need to be stationary or moving at low speed during detection, because the weld of a vehicle wheel moving at high speeds is difficult to detect. The method proposed in this paper needs to use prior information about the dataset. It is also a multi-stage processing architecture. Future work could focus on changing the structure of the convolution network and finding a more effective loss function or other training techniques to improve the one-stage object detection accuracy, or an adaptive processing architecture could be considered.

Tab. 4. Comparisions of different methods on KITTI data set

Algorithm	Average Precision (%)									
	Van			Pedestrian			Cyclist			mAP(%)
	Easy	Moderate	Hard	Easy	Moderate	Hard	Easy	Moderate	Hard	
YOLO v3	74.32	60.18	47.54	83.51	78.37	64.25	85.94	80.64	54.39	69.90
YOLO v4	76.14	64.27	51.46	85.24	77.53	67.91	88.27	78.72	58.11	71.96
Ours	75.98	67.84	58.06	87.84	79.52	72.30	89.57	81.25	59.25	74.63

References

[1] Viola P, Jones M. Robust real-time object detection. Int J Comput Vis 2004; 57(2): 137-154.

[2] Chen TT, Wang RL, Dai B, Liu DX, Song JZ. Likelihood-field-model-based dynamic vehicle detection and tracking for self-driving. IEEE trans Intell Transp Syst 2016; 17(11): 3142-3158.

[3] Fu ZH, Chen YW, Yong HW, Jiang RX, Zhang L, Hua XS. Foreground gating and background refining net-work for surveillance object detection. IEEE Trans Image Process 2019; 28(12): 6077-6090.

[4] Kong H, Yang J, Chen ZH. Accurate and efficient inspection of speckle and scratch defects on surfaces of planar products. IEEE Trans Industr Inform 2017; 13(4): 1855-1865.

[5] Guo ZX, Shui PL. Anomaly based sea-surface small target detection using k-nearest neighbour classification. IEEE Trans Aerosp Electron Syst 2020; 56(6): 4947-4964.

[6] Imoto K, Nakai T, Ike T, Haruki K, Sato Y. A CNN-based transfer learning method for defect classifica-tion in semiconductor manufacturing. IEEE Trans Semicond Manuf 2019, 32(4): 455-459.

[7] Pashina TA, Gaidel AV, Zelter PM, Kapishnikov AV, Nikonorov AV. Automatic highlighting of the region of interest in computed tomography images of the lungs. Computer Optics 2020; 44(1): 74-81. DOI: 10.18287/2412-6179-CO-659.

[8] Zou ZX, Shi ZW, Guo YH, Ye JP. Object detection in 20 years: A survey. arXiv Preprint 2019. Source: ⟨https://arxiv.org/abs/1905.05055⟩.

[9] Lecun Y, Bottou L, Bengio Y, Haffner P. Gradient-based learning applied to document recognition. Proc IEEE 1998, 86(11): 2278-2324.

[10] Lowe DG. Object recognition from local scale-invariant features. IEEE Int Conf on Computer Vision, Kerkyra 1999: 1150-1157. DOI: 10.1109/ICCV.1999.790410.

[11] Dalal N, Triggs B. Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego 2005: 886-893. DOI: 10.1109/CVPR.2005.177.

[12] Suykens JAK, Vandewalle J. Least squares support vector machine classifiers. Neural Process Lett 1999; 9: 293-300.

[13] Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. Int Conf on Neural Information Processing Systems, New York 2012: 1097-1105.

[14] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. IEEE Conf on Computer Vision and Pattern Recognition, Las Vegas 2016: 779-788. DOI: 10.1109/CVPR.2016.91.

[15] Redmon J, Farhadi A. YOLO9000: better, faster, stronger. IEEE Conf on Computer Vision and Pattern Recognition, Honolulu 2017: 7263-7271. DOI: 10.1109/CVPR.2017.690.

[16] Redmon J, Farhadi A. YOLOv3: An incremental improvement. arXiv Preprint 2018. Source: ⟨https://arxiv.org/abs/1804.02767⟩.

[17] Bochkovskiy A, Wang CY, Mark-Liao HY. YOLOv4: Optimal speed and accuracy of object detection. arXiv Preprint 2020. Source: ⟨arXiv:2004.10934⟩.

[18] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY,Berg AC. SSD: Single shot multibox detector, European Conf on Computer Vision European, Cham 2016: 21-37.

[19] Gidaris S, Komodakis N. Object detection via a multi-region and semantic segmentation-aware CNN model. Int Conf on Computer Vision, Santiago 2015: 1134-1142. DOI: 10.1109/ICCV.2015.135.

[20] Girshick R. Fast R-CNN. Int Conf on Computer Vision, Santiago 2015: 1440-1448. DOI: 10.1109/ICCV.2015.169.

[21] Ren SQ, He KM, Girshick R, Sun J. Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 2016; 39(6): 1137-1149.

[22] He KM, Gkioxari G, Dollar P, Girshick R. Mask R-CNN. IEEE Int Conf on Computer Vision, Venice 2017: 2980-2988. DOI: 10.1109/ICCV.2017.322.

[23] Cai ZW, Vasconcelos N. Cascade R-CNN: Delving into high quality object detection. IEEE Conf on Computer Vision and Pattern Recognition, Salt Lake City 2018: 6154-6162. DOI: 10.1109/CVP R.2018.00644.

[24] Zhou HY, Zhuang ZL, Liu Y, Liu Y, Zhang X. Defect classification of green plums based on deep learning. Sensors 2020; 20(23): 6993.

[25] Huang LC, Yang Y, Deng YF, Yu YN. Densebox: Unifying landmark localization with end to end object detection. arXiv Preprint 2015. Source: ⟨https://arxiv.org/abs/1509.04874⟩.

[26] Rezatofighi H, Tsoi N, Gwak JY, Sadeghian A, Reid L, Savarese S. Generalized intersection over union: A metric

and a loss for bounding box regression. IEEE Conf on Computer Vision and Pattern Recognition, Long Beach 2019: 658-666. DOI: 10.1109/CVPR.2019.00075.
[27] Everingham M, Gool LV, Williams CKI, Winn J, Zisserman A. The PASCAL visual object classes (VOC) Challenge. Int J Comput Vis 2010; 88: 303-338.
[28] Wang CY, Mark-Liao HY, Wu YH, Chen PY, Hsieh JW, Yeh IH. CSPNet: A new backbone that can enhance learning capability of CNN. IEEE Conf on Computer Vision and Pattern Recognition Workshops 2020: 1571-1580. DOI: 10.1109/CVPRW50498.2020.00203.
[29] Zheng ZH, Wang P, Liu W, Li JZ, Ye RG, Ren DW. Distance-IoU Loss: Faster and better learning for bounding box regression. arXiv Preprint 2019. Source: ⟨https://arxiv.org/abs/1911.08287⟩.
[30] Bodla N, Singh B, Chellappa R, Davis LS. Soft-NMS – Improving object detection with one line of code. IEEE Int Conf on Computer Vision, Venice 2017: 5562-5570. DOI: 10.1109/ICCV.2017.593.
[31] He YH, Zhang XY, Savvides M, Kitani K. Bounding box regression with uncertainty for accurate object detection. arXiv Preprint 2018. Source: ⟨https://arxiv.org/abs/1809.08545⟩.
[32] Geiger A, Lenz P, Urtasun R. Are we ready for autonomous driving? The KITTI vision benchmark suite. IEEE Conf on Computer Vision and Pattern Recogni-tion 2012; 3354-3361. DOI: 10.1109/CVPR.2012.6248074.

Authors' information

Tian Jiao Liang (b. 1998) graduated from University of Jinan in 2020, major in computer science. Currently he is pursuing the Master degree in College of Robotics, Beijing Union University. His research interest includes machine learning and object detection.

Wei Guo Pan (b. 1984) graduated from University of Chinese Academy of Sciences. He is currently a lecturer in Beijing Key Laboratory of Information Service Engineering, Beijing Union University. His research interest includes machine learning, object detection and intelligent driving. He is also the **corresponding author** of this paper.
E-mail: *ldtweiguo@buu.edu.cn* .

Hong Bao (b. 1958) graduated from Beijing Jiaotong University. He is a professor in Beijing Key Laboratory of Information Service Engineering, Beijing Union University. His research interests include pattern recognition, computer vision and digital image processing.

Feng Pan (b. 1978) graduated from Nanjing University of Science and Technology. He is currently an associate professor at College of Robotic, Beijing Union University from 2016. His research interest includes machine learning and intelligent driving.

Received March 5, 2021. The final version – August 18, 2021.