IET Electrical Systems in Transportation

Special Issue: Selected Papers from the 2019 IEEE Vehicle Power and Propulsion Conference (VPPC)

IET Journals
The Institution of Engineering and Technology

Research on deep learning method for rail surface defect detection

ISSN 2042-9738
Received on 30th March 2020
Revised 25th June 2020
Accepted on 22nd September 2020
E-First on 18th November 2020
doi: 10.1049/iet-est.2020.0041
www.ietdl.org

Jiang Hua Feng[1], Hao Yuan[1] ✉, Yun Qing Hu[1], Jun Lin[1], Shi Wang Liu[1], Xiao Luo[1]

[1]CRRC Zhuzhou Institute Co., Ltd, Shidai Road, Zhuzhou, People's Republic of China

✉ *E-mail: electronicYH@163.com*

Abstract: Rail surface defect detection plays a critical role in the maintenance of the rail transportation system. Video analysis technology is a promising method to detect defects due to its low cost and effectiveness. Several attempts with hand-craft features have been made to obtain the detection results by using traditional machine vision algorithms. However, these methods suffer from imprecise results due to challenging conditions, such as deteriorated and changeable lighting environment and various types of complex rail surface defects. Recently, classification methods with complex deep convolutional networks have become popular. Despite their high accuracy, these methods cannot meet the requirements of defects localisation and real-time processing in practice. To solve these problems, this study proposes a novel object detection algorithm to detect rail defects. The net architecture of the proposed algorithm includes a backbone network using MobileNet and several novel detection layers with multi-scale feature maps inspired by you only look once (YOLO) and feature pyramid networks. Two different architectures of MobileNet are used to estimate the performance of defects detection. The experimental results demonstrate the great potential of the proposed algorithm with fast inference speed and high accuracy in the industry.

1 Introduction

In recent years, with the rapid development of the high-speed and heavy-haul railway in China, more strict requirements for the reliability of railway infrastructure have been proposed. As a fundamental component, rail suffers from high risks of all kinds of damages involving rolling contact fatigue, material microstructure degradation and so on, due to the various train load and changeable environments [1, 2]. Increased rail defects will cause potential damages to the locomotives or trains running at high speed in rails due to coupled vibration. It is a great threat to traffic safety and may lead to high maintenance costs [3]. Therefore, it is very important to detect rail surface defects in time [4].

In the past few decades, rail defects detection mainly depends on manual rail monitoring. It is an error-prone, costly and time-consuming process. In addition, it is also dangerous to workers in the field for rail monitoring. At present, these methods are gradually replaced by automated detection. There have been a number of studies involving automated detection that have reported, such as ultrasonic inspection [5], eddy current inspection [6], magnetic flux leakage inspection [7] and non-destructive inspection using video cameras [8, 9]. Ultrasonic inspection is limited by the narrow adaptation range associated with train speed, and the eddy current inspection and magnetic flux leakage inspection are complicated to extract rail surface features from sensor signals. In contrast, video analysis technology has the advantages of non-contact, low cost, fast speed and high precision for rail defects detection. It has become a research hotspot and been widely used in the industry.

At present, several attempts using video cameras have been made to detect rail surface defects automatically based on traditional machine vision technology. These methods generally design hand-craft features or predefined features by analysing the images of rail surface defects manually, and then a corresponding feature learning algorithm is proposed for classification. In [10, 11], hand-craft features have been extracted to detect defects by analysing the boundaries of defects. In [12], a local normalisation algorithm was proposed to enhance the contrast of rail images for feature extraction. Moreover, a reverse Perona–Malik (P–M) diffusion algorithm was used to detect rail surface defects in [13]. These methods are proven to have a certain effect on rail defects detection. However, their common drawbacks are that the detection results generally have low accuracy and recall. Several defects, including linear defects, crack and small defects, are difficult to be detected and distinguished. The authors in [14, 15] proposed a background difference algorithm to detect defects. Tian *et al*. [16] used an improved Sobel operator to find available features of rail defects. Wang *et al*. [17] proposed a fusion principal components analysis (PCA) mode to recognise the rail surface defects with colour features. In [18], a sequenced combination of grey balance model, the phase spectrum and the Otsu threshold segmentation method was proposed to detect rail defects. However, the lack of generation has existed as the main problem for these methods. The detection results are prone to be adversely affected by changeable lighting conditions and deteriorated visual environments.

In practice, these difficult cases are common due to various reasons, for example as a result of the complex and rare defects, due to the blurred images of key components caused by uneven reflection in image acquisition, or because of the extreme weather conditions. These factors determine that traditional machine vision technologies with hand-craft features are not the best choice for defects detection. In recent years, with the rapid development of deep learning, deep convolutional neural networks (CNNs) have been introduced to automatically extract features with good generation and accuracy. With these powerful abilities, several studies [19, 20] have used the deep CNNs to classify the rail surface defects and achieved good results. However, the main drawbacks of these methods are that they cannot locate the defects in images, which is crucial in practice. Moreover, the proposed complex models cannot achieve fast inference.

Object detection is a fundamental task in the field of computer vision for locating instances of objects in images. In the past decade, it has been widely used in many fields [21, 22]. The detection accuracy in several fields has even surpassed that of humans [23]. It could be a promising method to detect and locate rail defects.

The advanced object detection algorithms based on deep learning are mainly divided into two categories. One is the two-stage detection algorithm, such as region-based CNN (RCNN) [24], fast RCNN [25], faster RCNN [26]. These kinds of algorithms select the candidate area in the first stage and detect the classification of the selected candidate area in the second stage.

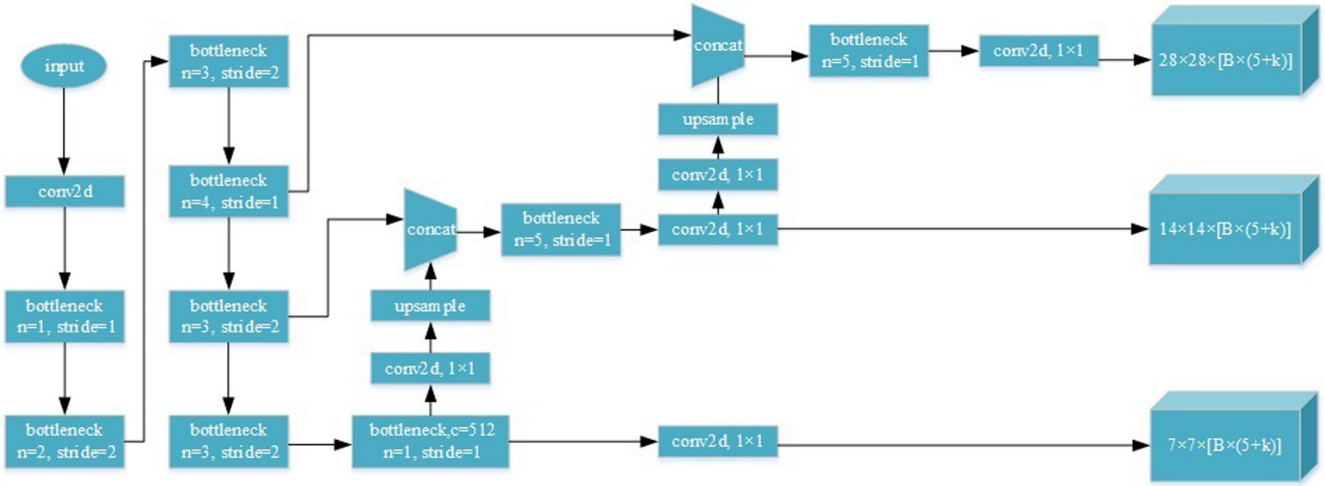

Fig. 1 *M2-Y3 network structure*

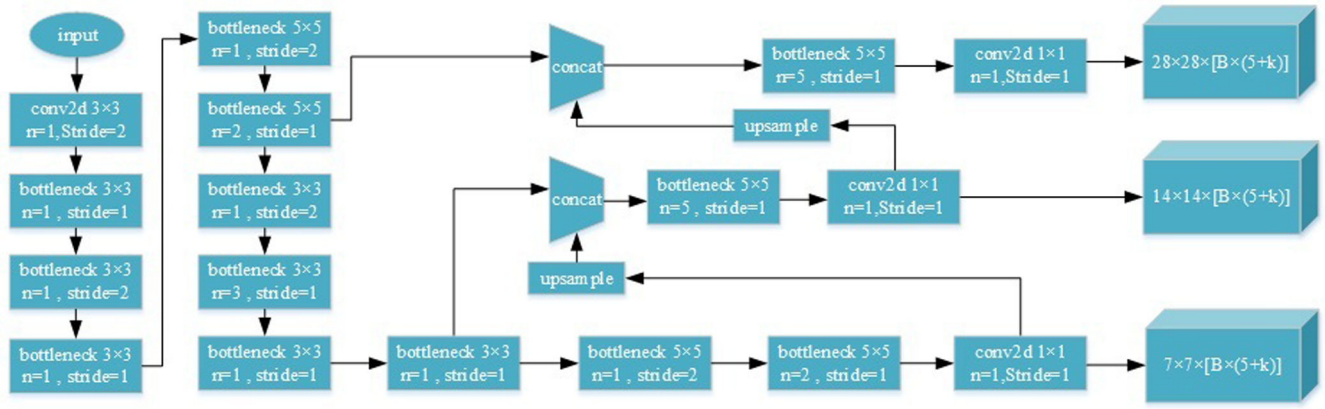

Fig. 2 *M3-Y3 network structure*

Despite high accuracy and efficacy, they are associated with the increased risk of slow inference speed. The other is the one-stage detection algorithm, such as YOLO [27–29] and single shot multi-box detector (SSD) [30]. The basic ideas of these kinds of algorithms are treating object detection as the combination of object position regression and object classification, which can significantly simplify the calculation and save the inference time [31]. In addition, it has been proven that combined with feature pyramid networks (FPNs), the accuracy of these algorithms can obtain further improvement.

In order to detect and locate rail surface defects with high accuracy, this paper proposes a novel object detection algorithm to detect rail defects. The net architecture of the proposed algorithm includes a backbone network using MobileNet and several novel detection layers with multi-scale feature maps inspired by YOLO and FPNs. The main advantages of the proposed method are high accuracy and fast inference speed. Even under occlusion and deteriorated visual conditions, the algorithm indicates great performance and capabilities to detect and locate rail surface defects simultaneously.

The remaining part of this paper is summarised as follows: a novel rail defects detection algorithm is proposed in Section 2. In Section 3, the experimental tests are carried out to validate the feasibility of the proposed algorithm. Finally, some major conclusions are summarised in Section 4.

2 Model architecture

This paper proposes a novel object detection algorithm to detect rail defects. The model architecture includes a backbone network for feature extraction and several detection layers. Two different networks involving MobileNetV2 [32] and MobileNetV3 [33] are used as the backbone networks, respectively. The design of detection layers with multi-scale feature maps are inspired by the YOLOv3 and FPNs. The objects with different sizes in the image are responsible by different detection layers of outputs. In this way, the model architecture is significantly simplified and the corresponding computation is reduced, compared with the two-stage detection architecture. In this paper, these two models are denoted as M2-Y3 and M3-Y3. The schematic diagram of two models with a single-scale input and multi-scale outputs are shown in Figs. 1 and 2, respectively. The backbone details of these models are shown in Tables 1 and 2.

2.1 Backbone networks

The MobileNetV2-bottleneck module is shown in Fig. 3, which is divided into two different branches according to the stride. When stride = 1, the bottleneck block contains a convolutional layer, a depth-wise separable convolution layer [34] with an activation function of ReLU6, a convolutional layer with a linear activation function. Also a shortcut connection between bottleneck blocks. When stride = 2, the detail is the same as the case of stride = 1, but the results of bottleneck blocks are directly output to the next layer. The bottleneck block named inverted residuals block is similar to residuals block. It firstly expands the number of channels and then compresses the number of channels, which is contrary to residual blocks. This design is useful to retain more features. The depth-wise separable convolutional layer is used instead of the traditional convolutional layer. It can decouple spatial from the channel, and significantly reduce the computation with just a slight loss on precision. The linear activation function is used instead of ReLU6 in the output layer of the bottleneck. It can avoid losing feature information when the number of tensor channels is small, which is beneficial for improving the capability of feature extraction [32].

The MobileNetV3-bottleneck module is shown in Fig. 4. The difference from MobileNetV2-bottleneck is to place a novel module named squeeze-and-excite into the inverted residual block

Table 1 M2-Y3 structure

Input	Operator	*t*	*c*	*n*	*S*
$224^2 \times 3$	conv2d	—	32	1	2
$112^2 \times 32$	V2-bottleneck	1	16	1	1
$112^2 \times 16$	V2-bottleneck	6	24	2	2
$56^2 \times 24$	V2-bottleneck	6	32	3	2
$28^2 \times 32$	V2-bottleneck	6	64	4	1
$28^2 \times 64$	V2-bottleneck	6	96	3	2
$14^2 \times 96$	V2-bottleneck	6	160	3	2
$7^2 \times 160$	V2-bottleneck	6	512	1	1
$7^2 \times 512$	conv2d 1 × 1	—	$7 \times 7 \times B \times (5 + k)$	1	1

Specification for M2-Y3. V2-bottleneck represents the inverted residual bottleneck of MobileNetV2. The expansion factor *t* is always applied to the input size of the bottleneck. All layers in the same bottleneck module have the same number *c* of output channels. Each line describes a sequence of 1 or more bottleneck modules, repeated *n* times. The first layer of each bottleneck module has a stride *s* and all others use stride 1. All spatial convolutions use 3 × 3 kernels [32].

Table 2 M3-Y3 structure

Input	Operator	*c*	*n*	*s*	NL	SE
$224^2 \times 3$	conv2d,3 × 3	16	1	2	HS	—
$112^2 \times 16$	V3-bottleneck 3 × 3	16	1	1	RE	—
$112^2 \times 16$	V3-bottleneck 3 × 3	24	1	2	RE	—
$56^2 \times 24$	V3-bottleneck 3 × 3	24	1	1	RE	—
$56^2 \times 24$	V3-bottleneck 5 × 5	40	1	2	RE	✓
$28^2 \times 40$	V3-bottleneck 5 × 5	40	2	1	RE	✓
$28^2 \times 40$	V3-bottleneck 3 × 3	80	1	2	HS	—
$14^2 \times 80$	V3-bottleneck 3 × 3	80	3	1	HS	—
$14^2 \times 80$	V3-bottleneck 3 × 3	112	1	1	HS	✓
$14^2 \times 112$	V3-bottleneck 3 × 3	112	1	1	HS	✓
$14^2 \times 112$	V3-bottleneck 5 × 5	160	1	2	HS	✓
$7^2 \times 160$	V3-bottleneck 5 × 5	160	2	1	HS	✓
$7^2 \times 160$	conv2d,1 × 1	$7 \times 7 \times B \times (5 + k)$	1	1	HS	—

Specification for M3-Y3. V3-bottleneck represents the MobileNetV3-bottleneck. Each line describes a sequence of 1 or more bottleneck modules, repeated *n* times. All layers in the same bottleneck module have the same number *c* of output channels. The layer of each bottleneck module has a stride *s*. SE denotes whether there is a squeeze and excite in that block. NL denotes the type of non-linearity used. Here, HS denotes *h*-swish and RE denotes ReLU6.

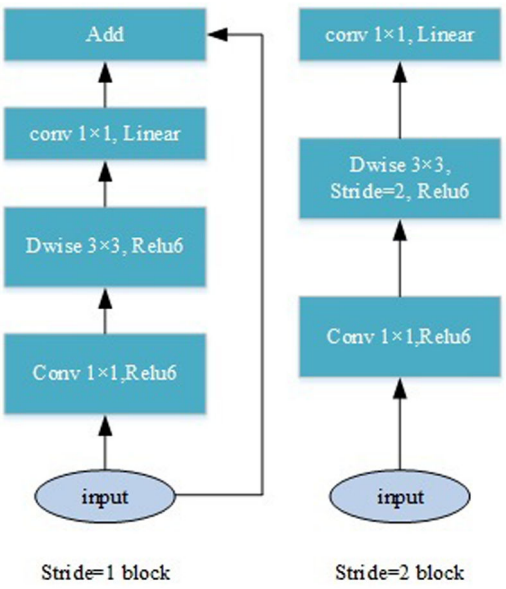

Fig. 3 *MobileNetV2-bottleneck module*

after depth-wise separable convolutional layer. In addition, the layers of MobileNetV3 bottleneck are improved by using swish non-linearities. The hard sigmoid activation function is also used in squeeze and excite to add the swish non-linearity for assuring accuracy in fixed-point arithmetic. The hard sigmoid is defined as follows:

$$h - \text{swish}[x] = x\frac{\text{Relu6}(x+3)}{6}. \quad (1)$$

The number of channels in squeeze and excite has been reduced to one-fourth of the input channels, which contributes to ∼15% latency reduction without loss of mean average precision (mAP) [33].

Table 3 shows the performance comparison of ImageNet dataset for different MobileNet networks. Obviously, as the lightweight networks with fast inference speed, MobileNetV2 and MobileNetV3 also have a good performance on accuracy.

2.2 Detection layers

The designs of M2-Y3 and M3-Y3 are inspired by YOLO series [27–29], which use a regression solution to solve the problem of object detection. The feature maps of network output are divided into $m \times m$ cells. Each cell corresponds with a certain region of the input image of the rail surface. It is assumed that there is a geometric centre point of a defect locating in a cell. This cell is responsible for predicting *B* candidate objects with the confidences. Referring to FPN [35], the detection layers of M2-Y3 and M3-Y3 use a multi-scale prediction structure as shown in Figs. 1 and 2. The dimensions of three output tensors with 7 × 7, 14 × 14 and 28 × 28 cells are $7 \times 7 \times [B \times (5 + k)]$, $14 \times 14 \times [B \times (5 + k)]$ and $28 \times 28 \times [B \times (5 + k)]$, respectively. *k* represents the number of categories of rail defects that need to be predicted. Take the first

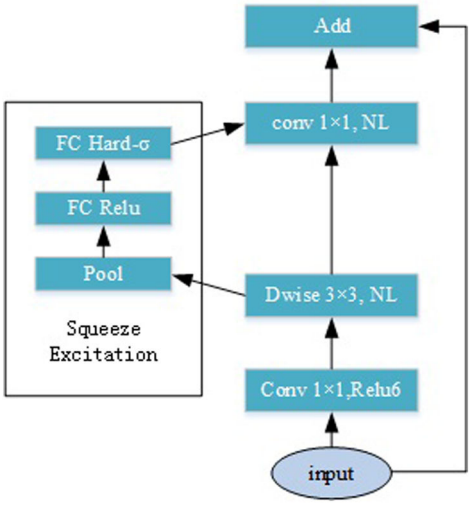

Fig. 4 *MobileNetV3-bottleneck module*

Table 3 Model comparison

Network	Top 1	Params	MAdds
MobileNetV1	70.6	4.2M	575M
MobileNetV2	72.0	3.4M	300M
MobileNetV2(1.4)	74.7	6.9M	585M
MobileNetV3-Large1.0	75.2	5.4M	219M

Performance comparison on ImageNet dataset for different networks. The total number of multiply-adds is also taken into account [33]

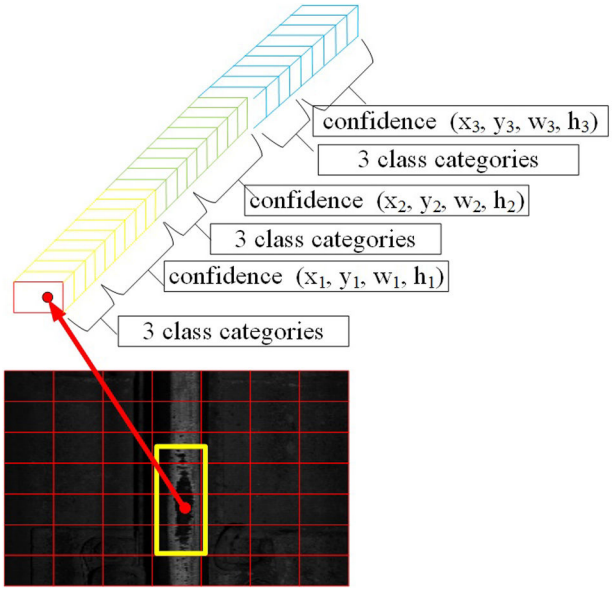

Fig. 5 *Rectangular region prediction*

output tensor as an example. Setting $B = 3$, which means each cell is responsible for predicting three bounding boxes. As shown in Fig. 5, since the geometric centre point of the rail surface defect is located in the fifth row and the fourth column of the predefined cells, this cell is possible to predict the defect. The prediction results contain the type and the confidence of the predicted object and relevant parameters of the bounding boxes involving width–height (w, h) and centre coordinates (x, y), so that the channel of the output tensor is $3 \times (5 + 3)$ and the dimension is $7 \times 7 \times [3 \times (5 + 3)]$.

2.3 Loss function and anchor box clustering

In this paper, the loss function referred to YOLOv3 is used to train models:

$$\text{loss(object)} = l_{\text{box}} + l_{\text{obj}} + l_{\text{cls}} \tag{2}$$

$$l_{\text{box}} = \lambda_{\text{coord}}(2 - w_i \times h_i) \times \sum_{i}^{m\times m}\sum_{j}^{B} I_{ij}^{\text{obj}}\left[\left(x_i - \hat{x}_i^j\right)^2 + \left(y_i - \hat{y}_i^j\right)^2 + \left(w_i^j - \hat{w}_i^j\right)^2 + \left(h_i^j - \hat{h}_i^j\right)^2\right] \tag{3}$$

$$l_{\text{obj}} = -\sum_{i}^{m\times m}\sum_{j}^{B} I_{ij}^{\text{obj}}\left[\hat{C}_i^j \log\left(C_i^j\right) + \left(1 - \hat{C}_i^j\right)\log\left(1 - C_i^j\right)\right] - \lambda_{\text{no obj}}\sum_{i}^{m\times m}\sum_{j}^{B} I_{ij}^{\text{no obj}}\left[\hat{C}_i^j \log\left(C_i^j\right) + \left(1 - \hat{C}_i^j\right)\log\left(1 - C_i^j\right)\right] \tag{4}$$

corrugation fatigue block stripping off block

Fig. 6 *Typical rail surface defects*

Table 4 Sample distribution

	Corrugation	Fatigue block	Stripping off block
train	29,064	40,644	72,708
valid	7750	10,838	19,389
test	1938	2710	4847

$$l_{\text{cls}} = -\sum_{i}^{m\times m}\sum_{i}^{B} I_{ij}^{\text{obj}} \sum_{c\in \text{classes}} \left[\hat{p}_i^j(c)\log\left(p_i^j(c)\right) + \left(1-\hat{p}_i^j(c)\right)\log\left(1-p_i^j(c)\right)\right] \quad (5)$$

where I_{ij}^{obj} denotes that the jth bounding box predictor in cell i is 'responsible' for that prediction [29], and $p_i(c)$ represents the conditional class probabilities. C_i^j represents the confidence score from the ground truth.

The two parameters, λ_{coord} and $\lambda_{\text{no obj}}$ are used to increase the loss from bounding box coordinate predictions and decrease the loss from confidence predictions for boxes that do not contain objects.

In order to improve the detection effect of small targets, the weight of localisation error and bounding box coordinate error is modified by adding weight: $(2-w_i \times h_i)$.

The object position is obtained by using the regression method based on the anchor box. In order to improve the precision of the models, the anchor box should be pre-calculated before model training by using the datasets of rail defect images as mentioned in YOLO. The k-means clustering algorithm is introduced to calculate the anchor box. The steps are as follows:

(1) Take the width and height of the rail surface defect in images as a sample, which is denoted as (w_n, h_n), $n\in\{1, 2, \ldots, N\}$. The corresponding centre point of the rail surface defect is denoted as (x_n, y_n), $n\in\{1, 2, \ldots, N\}$.

(2) Randomly select k group points (W_m, H_m), $m\in\{1, 2, \ldots, k\}$ in the set as the initial cluster centre.

(3) Calculate the distance from all points in the set to the k cluster centres, and assign the samples to the nearest cluster centres to obtain k point clusters. The formula is as follows:

$$d = 1 - \text{IoU}[(x_n, x_n, w_n, h_n), (x_n, x_n, W_m, H_m)] \quad (6)$$

where IoU represents intersection over union.

(4) The cluster centre point is recalculated. N represents the number of target boxes of the mth cluster. The formula is as follows:

$$W_{\hat{m}} = \frac{1}{N_m}\sum w_m \quad (7)$$

$$H_{\hat{m}} = \frac{1}{N_m}\sum h_m. \quad (8)$$

(5) Repeat steps 3 and 4 until the cluster centre points do not change anymore, which indicates the end of clustering. k cluster centre parameters are obtained as the width and height of the candidate anchor boxes for rail surface defects.

3 Experiment

3.1 Dataset

The images of rail surface are captured by the line scan cameras mounted on the comprehensive inspection train. During pre-processing, these line images are converted into images with dimensions of $224 \times 224 \times 3$ for model training. The typical rail surface defects, such as corrugation, fatigue block and stripping off the block, are labelled manually as shown in Fig. 6. Various data enhancement methods, including random shifting, contrast enhancement, random noise addition, rotation and horizontal flip, are used to improve the generation of the proposed model. The dataset is randomly divided into three subsets for training, validation and test with the respective proportion of 0.75:0.2:0.05. The specific sample distribution of these rail defects images is shown in Table 4.

3.2 Train and test

The proposed algorithms are trained and evaluated on a computer with Xeon E5-2600 v3 CPU, GTX1080 TI GPU and 16 GB memory. YOLOv3, M2-Y3 and M3-Y3 are implemented with Keras on Ubuntu 16.04.

The training data set are used to train models. The sizes of anchor boxes are first calculated by the proposed k-means method. The batch size of training is set to 32. In the first 50 iterations, the learning rate is set to 1×10^{-3}. After 50 iterations, if the loss does not decrease for five consecutive iterations, the learning rate drops to one-tenth of the original. If the loss does not decrease for ten consecutive rounds, the training stops. At last, the losses of models are reduced to 5.83, 5.76 and 5.68%.

The test results of rail defects detection are shown in Figs. 7–9. The types and positions of the proposed defects are indicated by distinctive colours in images. It is obvious that all tested detection

Fig. 7 *YOLOv3 test samples*
(a) Result of YOLOv3 corrugation detection, ***(b)*** Result of YOLOv3 fatigue block detection, ***(c)*** Result of YOLOv3 stripping off block detection

Fig. 8 *M2-Y3 test samples*
(a) Result of M2-Y3 corrugation detection, ***(b)*** Result of M2-Y3 fatigue block detection, ***(c)*** Result of M2-Y3 stripping off block detection

Fig. 9 *M3-Y3 test samples*
(a) Result of M3-Y3 corrugation detection, ***(b)*** Result of M3-Y3 fatigue block detection, ***(c)*** Result of M3-Y3 stripping off block detection

Table 5 Test result

	Corrugation AP, %	Fatigue block AP, %	Stripping off block AP, %	mAP, %	FPS
YOLOv3	90.13	80.64	78.39	83.37	33–53
M2-Y3	95.28	84.60	82.33	87.40	55–70
M3-Y3	96.03	84.72	67.99	82.91	98–110

models can detect and locate the rail surface defects with high accuracy. It is a significant improvement to classify and locate defects.

In this paper, AP with IoU > 0.5 [36] and frames per second (FPS) are used to estimate the performance of proposed models. The detailed test result is shown in Table 5. It can be observed that M3-Y3 has the fastest inference speed with FPS range of 55–70

since the backbone network MobileNetV3 of this model is concise and effective. However, M2-Y2 has the best mAP with 83.37%. It seems that MobileNetV2 is more focused on the balance of accuracy and inference speed, comparing with MobileNetV3. Moreover, these two proposed models demonstrate better performance in comparison with the original YOLOv3 algorithm, which is contributed by the combination of advantages of MobileNet and YOLOv3.

4 Conclusion

This study provides new insights into rail surface defects detection. As it is well known, the traditional machine vision methods with hand-craft features generally lack robustness in practice under changeable lighting conditions and challenging environments. The novel rail surface defects detection models with different deep convolutional networks involving M2-Y3 and M3-Y3 are proposed in this paper. Two well-known and lightweight networks MobileNetV2 and MobileNetV3 are used as the backbone networks for features extraction. The design of detection layers with multi-scale feature maps are inspired by YOLOv3 and FPNs. The same loss function of YOLOv3 and carefully collected and labelled data sets of rail surface defects are used to train proposed models. The experimental results show that the proposed models can detect and locate the rail surface defects in real time and achieve high detection accuracy. In comparison with the original YOLOv3 algorithm, the proposed models also show a better performance due to the combination of advantages of MobileNet and YOLOv3.

5 References

[1] Tian, G.Y., Gao, B., Gao, Y., *et al.*: 'Review of railway rail defect non-destructive testing and monitoring', *Chin. J. Sci. Instrum.*, 2016, **37**, (8), pp. 1763–1780

[2] Zhou, Y., Zhang, J., Wang, S., *et al.*: 'Simulation on rail head crack initiation life prediction considering rail wear', *J. of the China Railw. Soc.*, 2016, **38**, (7), pp. 91–97

[3] Wang, J.: 'Rail surface defect detection based on visual attention and PLSA model', *J. of Railw. Sci. Eng.*, 2015, **12**, (3), pp. 509–514

[4] Ministry of Railways of the People's Republic of China: 'Rules of maintenance of railway lines', 2001

[5] Utrata, D., Clark, R.: 'Groundwork for rail flaw detection using ultrasonic phased array inspection', *Rev. of Quant. Nondestruct. Eval.*, 2003, **22**, (1), pp. 799–805

[6] Papaelias, M.P., Lugg, M.C., Roberts, C., *et al.*: 'High-speed inspection of rails using ACFM techniques', *NDT. E. Int.*, 2009, **42**, (4), pp. 328–335

[7] Chenchen, D., Wenbo, L., Wangcai, C.: 'Rail crack recognition based on multi-sensor feature-decision fusion', *Electron. Meas. Technol.*, 2017, **40**, (11), pp. 157–160

[8] Rubinsztejn, Y.: 'Automatic Detection of Objects of Interest from Rail Track Images', Manchester University, Manchester, 2011

[9] Marino, F., Distante, A., Ettore, S., *et al.*: 'A real-time visual inspection system for railway maintenance: automatic hexagonal-headed bolts detection', *IEEE Trans. on Syst. Man Cybern., C, Appl. Rev.*, 2007, **37**, (3), pp. 418–428

[10] Dubey, A.K., Jaffery, Z.A.: 'Maximally stable extremal region marking-based railway track surface defect sensing', *IEEE Sens. J.*, 2016, **16**, (24), pp. 9047–9052

[11] Yuan, X.C., Wu, L.S., Chen, H.W.: 'Rail image segmentation based on Otsu threshold method', *Opt. Precis. Eng.*, 2016, **24**, (7), pp. 1772–1781

[12] Li, Q., Ren, S.: 'A real-time visual inspection system for discrete surface defects of rail heads', *EEE Trans. Instrum. Meas.*, 2012, **61**, (8), pp. 2189–2199

[13] He Zh, D., Wang, Y.N., Mao, J.X., *et al.*: 'Research on inverse P-M diffusion-based rail surface defect detection', *Acta Autom. Sin.*, 2014, **40**, (8), pp. 1667–1679

[14] He Zh, D., Wang, Y.N., Liu, J., *et al.*: 'Background differencing-based high-speed rail surface defect image segmentation', *Chin. J. Sci. Instrum.*, 2016, **37**, (3), pp. 640–649

[15] Zh, M.Y., Yue, B., Ma, H.F., *et al.*: 'Rail surface defects detection based on gray scale gradient characteristics of image', *Chin. J. Instrum.*, 2018, **39**, (4), pp. 220–229

[16] Tian, S., Kong, J.Y., Wang, X.D.: 'Improved Sobel algorithm for defect detection of rail surfaces with enhanced efficiency and accuracy', *J. Central South Univ.*, 2016, **23**, (11), pp. 2867–2875

[17] Wang, H., Wang, M., Zhang, H.: 'Vision saliency detection of rail surface defects based on PCA model and color features', *Process. Autom. Instrum.*, 2017, **38**, (1), pp. 73–76

[18] Liu, Q.Q., Zhou, H.Y., Wang, X.S.: 'Research on rail surface defect detection method based on gray equalization model combined with gabor filter', *Chin. J. of Surface Technol.*, 2018, **47**, (11), pp. 300–304

[19] Souk Up, D., Huber-Mörk, R.: 'Convolutional neural networks for steel surface defect detection from photometric stereo images'. Int. Symp. on Visual Computing, Cham, 2014, pp. 668–677

[20] Faghih-Roohi, S, Hajizadeh, S., Núñez, A., *et al.*: 'Deep convolutional neural networks for detection of rail surface defects'. Int. Joint Conf. on Neural Networks IEEE, Vancouver, BC, Canada, 2016, pp. 2584–2589

[21] Du, X.Y., Dai, P., Li, Y., *et al.*: 'Automatic detection algorithm of railway plug based on deep learning', *Chin. J. of the China Railw. Soc.*, 2017, **38**, (3), pp. 89–96

[22] Dai, P., Wang, S.C., Du, X.Y., *et al.*: 'Machine vision method for flawless track fasteners based on semi-supervised deep learning', *Chin. J. of the China Railw. Soc.*, 2018, **39**, 161, (4), pp. 45–51

[23] He, K., Zhang, X., Ren, S., *et al.*: 'Delving deep into rectifiers: surpassing human-level performance on ImageNet classification'. IEEE Int. Conf. on Computer Vision, Santiago, Chile, 2015, pp. 1026–1034

[24] Girshick, R., Donahue, J., Darrell, T., *et al.*: 'Rich feature hierarchies for accurate object detection and semantic segmentation'. IEEE Conf. on Computer Vision & Pattern Recognition, Ohio USA, 2014, pp. 580–587

[25] Girshick, R.: 'Fast R-CNN'. IEEE Int. Conf. on Computer Vision, Santigago Chile, 2015, pp. 1440–1448

[26] Ren, S., He, K., Girshick, R., *et al.*: 'Faster R-CNN: towards real-time object detection with region proposal networks'. Int. Conf. on Neural. Information Processing Systems, Montreal, Canada, 2015, pp. 91–99

[27] Redmon, J., Divvala, S., Girshick, R., *et al.*: 'You only look once: unified, real-time object detection'. Proc. of Computer Vision and Pattern Recognition, Boston, MA, 2015, pp. 779–788

[28] Redmon, J., Farhadi, A.: 'YOLO9000: better, faster, stronger'. Proc. of the Computer Vision and Pattern Recognition, Honolulu USA, 2017, pp. 21–25

[29] Redmon, J, Farhadi, A.: 'YOLOv3: An Incremental Improvement', 2018

[30] Liu, W., Anguelov, D., Erhan, D., *et al.*: 'SSD: single shot MultiBox detector'. Computer Vision-ECCV, The Netherlands, Amsterdam, 2015, pp. 21–37

[31] Huang, J.J., Chen, N.N., Fan, Y.: 'Vehicle target detection based on global and local convolutional feature fusion', *J. of Southwest Univ. of Sci. Technol.*, 2018, **33**, (4), pp. 79–85

[32] Sandler, M., Howard, A., Zhu, M., *et al.*: 'Mobilenetv2: inverted residuals and linear bottlenecks'. The IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), Salt Lake USA, 2018, pp. 4510–4520

[33] Howard, A., Sandler, M., Chu, G., *et al.*: 'Searching for Mobile net v3'. Proc. of the IEEE Int. Conf. on Computer Vision, Seoul, Korea, 2019, pp. 1314–1324

[34] Howard, A.G., Zhu, M., Chen, B., *et al.*: 'Mobilenets: efficient convolutional neural networks for Mobile vision applications', https://arxiv.org/abs/1704.04861, 2017

[35] Lin, T.Y., Dollár, P., Girshick, R., *et al.*: 'Feature pyramid networks for object detection', https://arxiv.org/abs/1612.03144, 2016

[36] Everingham, M., Gool, L.V., Williams, C.K.I., *et al.*: 'The pascal visual object classes challenge (VOC2011) result'. *Int. J. Comput. Vis.*, 2010, **88**, (2), pp. 303–338