Railway Infrastructure Defects Recognition Using Fine-grained Deep Convolutional Neural Networks

Huaxi Huang[1], Jingsong Xu[1], Jian Zhang[1,*], Qiang Wu[1] and Christina Kirsch[2]
[1]*Global Big Data Technologies Centre, School of Electrical and Data Engineering.*
University of Technology Sydney, Australia
Emails:{Huaxi.Huang@student., Jingsong.Xu@, Jian.Zhang@, Qiang.Wu@}uts.edu.au
[2] *Sydney Trains, Australia*
Emails: Christina.Kirsch@transport.nsw.gov.au

***Abstract*—Railway power supply infrastructure is one of the most important components of railway transportation. As the key step of railway maintenance system, power supply infrastructure defects recognition plays a vital role in the whole defects inspection sub-system. Traditional defects recognition task is performed manually, which is time-consuming and high-labor costing. Inspired by the great success of deep neural networks in dealing with different vision tasks, this paper presents an end-to-end deep network to solve the railway infrastructure defects detection problem. More importantly, this paper is the first work that adopts the idea of deep fine-grained classification to do railway defects detection. We propose a new bilinear deep network named Spatial Transformer And Bilinear Low-Rank (STABLR) model and apply it to railway infrastructure defects detection. The experimental results demonstrate that the proposed method outperforms both hand-craft features based machine learning methods and classic deep neural network methods.**

***Index Terms*—Railway Infrastructure Defects Recognition; Fine-grained inspection; Deep Convolutional Neural Networks**

I. Introduction

With the development of digital image capturing technologies especially as the popularization of High-Definition (HD) resolution video cameras and infrared cameras, the accuracy, and efficiency of industrial inspection system have been improved significantly [1]–[6]. However, in most railway maintenance center (such as Sydney Trains Maintenance Center) today, defects inspection and recognition tasks are performed by railway maintenance engineers or related experts manually. Reviewers will check every frame of the infrastructure surveillance videos to find possible flaws in different power supply equipment. Firstly, they need to locate the specific objects and then focus on these objects to determine whether they are defective. This detect process is low-efficiency, time-consuming and high labor costing. More importantly, strong domain knowledge of railway infrastructure is needed during assessing the defective of a device. Figure 1 shows an example of the railway power supply infrastructure captured by a special HD video camera. We zoom in the target to the right corner, the green Dropper connects line 1 (L1) and line 3 (L3) and the red Dropper should connect line 4 (L4) and line 5 (L5). From this image, the difference between the defect equipment and the normal equipment is very small, this subtle difference also fits into the scene of fine-grained classification.

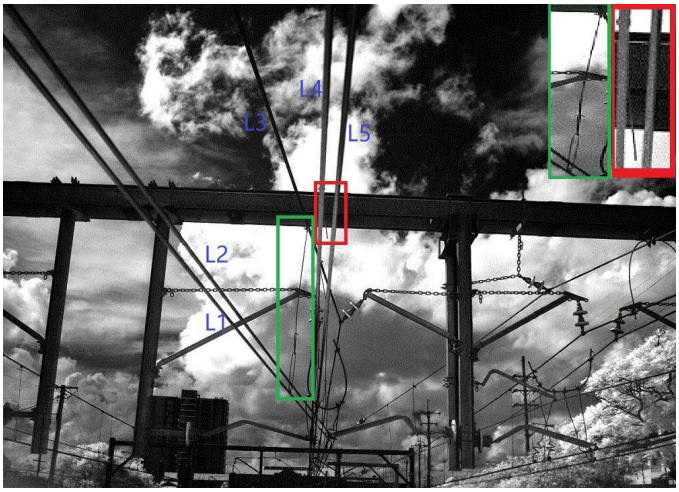
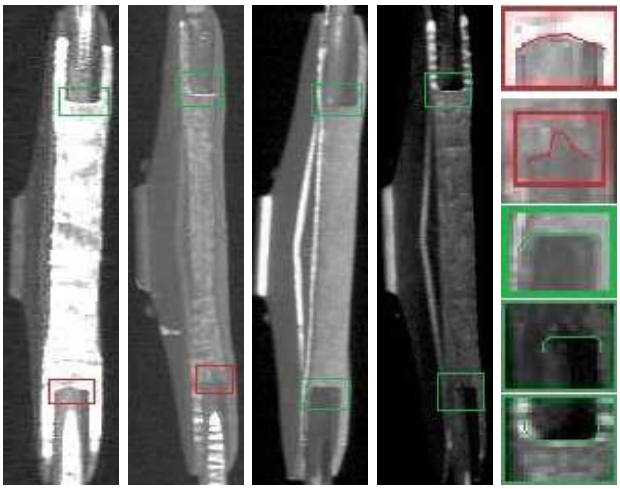


Fig. 1. An example of Railway Power Supply Infrastructure. We labeled two pieces of equipment with red and green boxes. The green one presents a standard Dropper equipment where the red one contains a defect Dropper equipment that is unconnected at the end of line 5 (L5).

For automatic vision-based industrial inspection, there mainly exist three types of methods: naive-vision based detection [7]–[11], hand-craft features with machine learning based inspection frameworks [1]–[3] and deep learning based detect methods [4]–[6].

Naive-vision based industrial inspection methods use traditional image features such as colour, gray value and so on. Otsu *et al.* [7] proposed a threshold defects detection method using a gray level histogram of bimodal distribution to inspect candidate images. Ng *et al.* [8] improved the performance of Otsu algorithms by picking up optimal thresholds for both bimodal and unimodal distribution of gray histogram from images. Besides these gray value features, Chan *et al.* [9] presented a fabric defects detection method based on Fourier transform. Lowe *et al.* [10] designed a pipe flaws detection framework using Guided waves and Kumar *et al.* [11] applied

*Corresponding Author.



(a) Defect (b) Defect (c) Standard (d) Standard (e) Detal

Fig. 2. An example of Splice defect. (a) and (b) show V-wear defects in connection of wire and splice labeled by red boxes, (c) and (d) show standard Splices. (e) is the magnification of the first four images details. The first row of (e) shows the defect part of (a) which is a V-wear, the second row shows the defect part of (b) and the remaining rows are samples of standard parts of Splice (c)' bottom part and (d) (both top and bottom parts) which we draw with green lines.

Gabor filters to inspect the defects in textile products.

Naive-vision based inspection methods are fast and efficient in some common industrial processing tasks. But it works lousily when the images are cluttered as well as with complex background as Figure 1 indicates. Some researches have attempted to solve these problems from the viewpoint of machine learning with hand-craft features. For example, Marani *et al.* [1] used K-Means, K-Medoids and hierarchical clustering with thermography features to detect subsurface flaws in composite materials. Jia *et al.* [2] designed an inspection system using Support Vector Machine (SVM) and Huang *et al.* [3] presented a real-time mobile phone workpiece surface detects detection framework based on Naive Bayesian and SVM with Histogram Of Gradient (HOG) and Local Binary Pattern (LBP) features.

Nevertheless, with the advent of the Big Data era, massive amounts of data need to be processed in a more precise and robust way which is difficult for traditional machine learning methods. Nowadays, deep neural networks are widely applied to computer vision and other areas due to their powerful data processing capacity and excellent performance in different applications like AlexNet [12], VGGNet [13] and ResNet [14] for image classification, Faster-RCNN [15] and YOLO [16] for object detection. There are some works build deep networks to deal with industrial defects inspection such as [4]–[6]. [4], [5] used deep structure to deal with rolling element bearings and planetary gearboxes data which are text logs about these equipment and achieved good performance. However, these data are relatively simple compared to images data. Faghih *et al.* [6] proposed a deep convolutional neural network (DCNN) for rail surface defects detection and got a great improvement, but its deep architecture is comparatively shallow which cannot capture higher-level semantic features of images.

Despite the fact that DCNN models have made great progress in defects detection, considering that defects detection is a fine-grained problem, it should be better to use a specific fine-grained model to deal with it. For example, in animal species classification, vehicle type discrimination, and food recognition, where classes in these datasets have small inter-class variation but large intra-class gaps. It is reported that fine-grained methods achieve better performance than classical DCNN models in dealing with these types of tasks [17]–[23].

For railway power infrastructure defects inspection, defects of damaged or worn equipment are usually found in some small parts compared to the whole object as you can see from the Figure 2, where this equipment named Splice is used to connect two different wires that can extend the whole power line. How to find these subtle defects is a typical fine-grained recognition problem.

In this paper, we use a fine-grained way to address this challenge. We follow with Lin's methodology [20], [24] and improve this algorithm using a combination of Spatial Transform and Low-rank operation. The contributions of this work include:

- To our best knowledge, this paper is the first to apply the deep fine-grained model to railway infrastructure defects detection. We deal with the challenge that complexes noisy background as well as subtle variation of objects in a fine-grained way. We define the defect detection as a two-class fine-grained problem: "defect" or "not-defect".
- This paper proposes a new bilinear convolutional neural network named Spatial Transformer And Bilinear Low-Rank (STBLR) model. In order to solve the high variation within the class, we adopt the Spatial Transformer Network and to achieve a more effective performance we present a Low-Rank Bilinear model.
- Cooperated with Sydney Trains, we construct a railway power supply infrastructure defects dataset.
- Through experiments, the proposed methods achieve the best performance on railway power infrastructure defects dataset compared with hand-craft machine learning based methods and classical deep frameworks.

II. Methodology

A. Dataset Construction

The data used in this paper is collected from Sydney Train Maintenance Center. Instead of performing defect detection in the full resolution images directly, we break the task into two steps: general object (no matter it is defective or not) detection and fine-grained object classification on given cropped objects. This paper focuses on the second part. We select 5 most common defects on specific equipment as the objective data which contains 1546 images. In each type of object like Splice, defect objects are classified into a new class and result in 10 classes (defect and non-defect objects). Therefore our dataset exists low inter-class variation but has high intra-class nonconformity due to the different posture and angle of cameras, illumination variations etc.

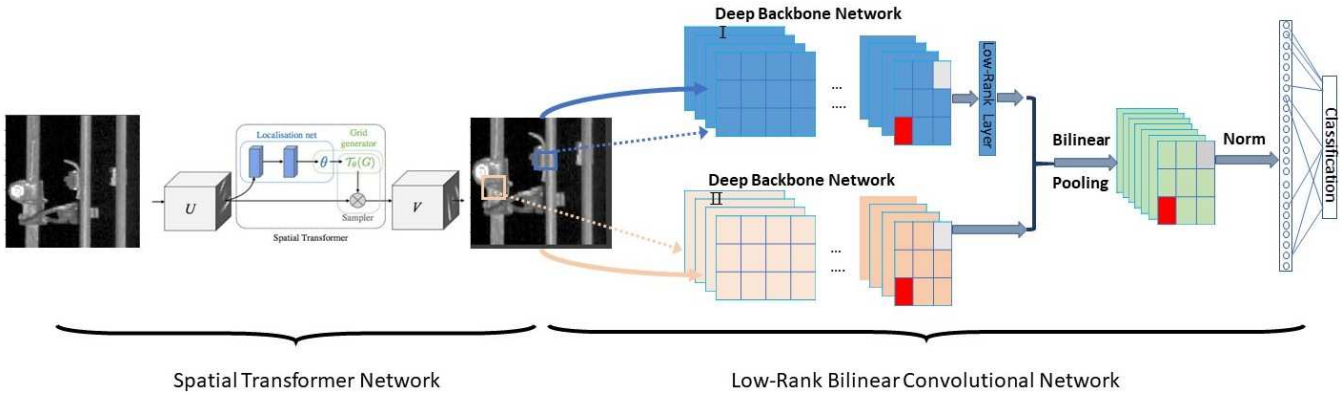


Fig. 3. The complete architecture of STABLR model. STABLR can be divided into two parts: STN and Low-Rank BCNN. STN can learn the invariant representation of the dataset and Low-Rank BCNN can capture the fine-grained features of the input images.

B. STABLR Model

Our STABLR model has two parts: Spatial Transformer Network (STN) and Low-Rank Bilinear Convolutional Neural Network (Low-Rank BCNN).

1) Spatial Transformer Network: One of the challenges of fine-grained classification is the large variation within class due to the pose and location difference of the target objects in the images. Jaberberg *et al.* [21] proposed a Spatial Transformer Network that can automatically learn the invariant representation of the original images and locate the target object in the images at the same time. In our model, we adopt the affine transformer networks as the original paper presented [21].

2) Low-Rank BCNN: Lin *et al.* [20], [24] proposed a sample and effective architecture for fine-grained classification. It applied outer product operation on deep feature maps to obtain second-order feature descriptors for final classification. A classic Bilinear CNNs model for images recognition can be defined as a quadruple:

$$\mathfrak{B} = (\mathfrak{E}_I, \mathfrak{E}_{II}, \mathfrak{P}_b, \mathcal{C}) \tag{1}$$

where $\mathfrak{B}$ is a bilinear CNNs model, $\mathfrak{E}_I$ and $\mathfrak{E}_{II}$ represent feature extractor functions which are specific deep convolutional neural networks like InceptionV3 and InceptionV4 in [23]. $\mathfrak{P}_b$ is the second-order pooling function and $\mathcal{C}$ represents a classifier. A feature extractor is defined as below:

$$\mathfrak{E} : \mathcal{S} \longrightarrow \mathcal{X} \in \mathcal{R}^{c \times hw} \tag{2}$$

where $\mathcal{S} \in \mathcal{R}^{H \times W \times C}$ represent the images with H height, W width and C color channels. Through function $\mathfrak{E}$, an image is transformed into a tensor $\mathcal{M} \in \mathcal{R}^{h \times w \times c}$ with c feature channels and h, w indicate the hight and width of the feature map. Then $\mathcal{M}$ is squeezed to a feature matrix $\mathcal{X}$ with $c \times hw$ dimensions. Given two specific functions $\mathfrak{E}_I : \mathcal{S} \longrightarrow \mathcal{X}_\alpha \in \mathcal{R}^{c_1 \times hw}$ and $\mathfrak{E}_{II} : \mathcal{S} \longrightarrow \mathcal{X}_\beta \in \mathcal{R}^{c_2 \times hw}$. Bilinear pooling operator can be described by the following formula:

$$\begin{aligned} \mathfrak{P}_b(\mathcal{S}, \mathfrak{E}_I, \mathfrak{E}_{II}) &= AVG(\mathcal{X}_\alpha \mathcal{X}_\beta^T) \\ AVG(\mathcal{X}_\alpha \mathcal{X}_\beta^T) &= \frac{1}{hw} \sum_{i=1}^{hw} f_{\alpha,i} f_{\beta,i}^T \end{aligned} \tag{3}$$

where $f_{\alpha,i} \in \mathcal{R}^{c_1}$ and $f_{\beta,i} \in \mathcal{R}^{c_2}$ denote feature vectors at specific location in each feature matrix $\mathcal{X}_\alpha$ and $\mathcal{X}_\beta$ with $i \in [1, hw]$. The pooled feature is a $c_1 \times c_2$ vector. Using matrix outer product, bilinear pooling produces a confertus representation of distinct features from different deep extractors at each location of feature maps in a second-order way.

Notice that an image passes through bilinear pooling in Equation 3 will become a $c_1 \times c_2$ vector. Using VGG-D and VGG-M, $c_1 = c_2 = 512$ that the final length of feature is 262,144 (262K). Using these high dimensional features will result in big overhead for time and storage. To address this problem, some researchers adopted some low-rank approximation methods to replace the original features [24]–[26].

Unlike [25], [26] using complex matrix dimension reduction methods, [24] applied PCA [27] to activation features before bilinear pooling and achieved consistent performance as [25], [26]. Followed this idea, we propose a simple but effective way to reduce the dimension of activation features:

$$\begin{aligned} & f_{\alpha_{low}} = \theta(f_{\alpha_{high}}) \\ & f_{\alpha_{low}} \in \mathcal{R}^{C_{low}}, \ f_{\alpha_{high}} \in \mathcal{R}^{C_{high}} \\ & C_{high} > C_{low} \end{aligned} \tag{4}$$

$\theta(\cdot)$ is a convolutional layer with 1×1 kernel size and 1 stride. And C_{high}, C_{low} are the input feature channels and output feature channels of the convolutional layer. With this layer, the proposed bilinear model can automatically learn the dimensions reduction rules of the features in an end-to-end way.

According to Lin [20], normalization operation after bilinear pooling could enhance the performance significantly. Without

TABLE I
CNN FEATURES EXTRACTORS FOR RAILWAY INFRASTRUCTURE DEFECTS.

Low-Rank Bilinear Features Extractors			
VGG16		VGG19	
Layer Type	Kernel; Out_dim	Layer Type	Kernel; Out_dim
Conv1_1	3×3; 64	Conv1_1	3×3; 64
Conv1_2	3×3; 64	Conv1_1	3×3; 64
Pool_1	2×2; Maxpooling	Pool_1	2×2; Maxpooling
Conv2_1	3×3; 128	Conv2_1	3×3; 128
Conv2_2	3×3; 128	Conv2_2	3×3; 128
Pool_2	2×2; Maxpooling	Pool_2	2×2; Maxpooling
Conv3_1	3×3; 256	Conv3_1	3×3; 256
Conv3_2	3×3; 256	Conv3_2	3×3; 256
Conv3_3	3×3; 256	Conv3_3	3×3; 256
		Conv3_4	3×3; 256
Pool_3	2×2; Maxpooling	Pool_3	2×2; Maxpooling
Conv4_1	3×3; 512	Conv4_1	3×3; 512
Conv4_2	3×3; 512	Conv4_2	3×3; 512
Conv4_3	3×3; 512	Conv4_3	3×3; 512
		Conv4_4	3×3; 512
Pool_4	2×2; Maxpooling	Pool_4	2×2; Maxpooling
Conv5_1	3×3; 512	Conv5_1	3×3; 512
Conv5_2	3×3; 512	Conv5_2	3×3; 512
Conv5_3	3×3; 512	Conv5_3	3×3; 512
Low-Rank	1×1; 64	Conv5_4	3×3; 512
Feature Maps		Feature Maps	
Bilinear Pooling layer; Output_dim: 64×512			

using normalization in [20], we adopt normalization on bilinear feature vector f in [28]:

$$\frac{sign(sqrt(f))\sqrt{|sqrt(f)|}}{\|sign(sqrt(f))\sqrt{|sqrt(f)|}\|_2} \tag{5}$$

At last, f will pass through a whole connected layer as the classifier and get final prediction results. The complete network architecture is in Figure 3.

C. CNN Feature Acquisition

In bilinear CNNs model (BCNN), how to define proper deep backbone networks is decisive where different deep neural network structure extracts different image features, thus determines the latter classification performance of the B-CNN. In Lin's previous works [20], [24], BCNN used VGG-M [29] and VGG-D [13] which removed the full connected classification layer as feature extractors. In [23], they adopted InceptionV3 [30] and InceptionV4 [31] as backbone networks.

In Section II-B, an image passes through bilinear pooling in Equation 3 will become a $c_1 \times c_2$ vector. Using VGG-D and VGG-M, $c_1 = c_2 = 512$ that the final length of feature is 262,144 (200K) where using InceptionV3 and InceptionV4, $c_1 = 2048$ and $c_2 = 1536$ which result in a 3,145,728 (3000K) feature vector. So the Inception based B-CNN model will have about 10 times more parameters need to be trained compared to VGG based model, and thus slower than VGG based BCNN model. For our railway infrastructure defects detection task, a real-time and high-accuracy inspection is required. Along these lines, in this paper, we choose VGG16 [13] and VGG19 [13] as the feature extractors which are more powerful than VGG-M and with fewer parameters than Inception networks. In addition, after passing through a Low-Rank layer, the final dimension of bilinear pooled features is 64×512 (32K). Unlike [20] resizes the images as 448×448 and [23] resizes input images as 229×229, we resize the input images in a 224×224 pixels size since our dataset is relatively simple and using this way can reduce training time while speeding up detection.

We remove last full connected layers of both VGG16 and VGG19 networks as image descriptor extract functions. The detailed features extractor architecture is illustrated in Table I.

TABLE II
RAILWAY DEFECTS DATASET.

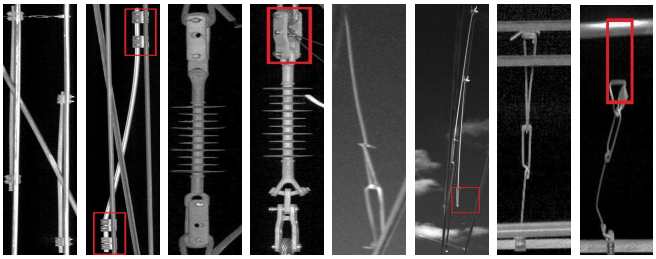
Equipment Name	Class Name	Image Number
Splice	Splice Standard	219
	Splice V-wear	116
Knuckle	Knuckle Standard	332
	Knuckle Misinstalled	36
K-line Insulator	Kline Standard	84
	Kline Twisted	79
Dropper1	Dropper1 Standard	91
	Dropper1 Defects	138
Dropper2	Dropper2 Standard	315
	Dropper2 Broken	136
Summary	10 classes	1546

III. EXPERIMENTS

In this section, we empirically evaluate the STABLR model on the actual railway power supply infrastructure defects dataset. Firstly, we will give a brief introduction to our dataset. Then we will introduce our experiment detailed. Finally, we will analyze the experiment results of both hand-craft machine learning based and classic deep neural networks methods.

A. Dataset

In section II-A, we introduced the way to construct the railway infrastructure dataset shortly. We pick up 2336 images from the original video data which contains power supply equipment and defects. The maximal resolutions of collected images are 2048 × 5400 pixels and 3792 × 2730 pixels, respectively. According to maintenance logs, we manually label the bounding boxes and crop 5 types of object from each original image and finally produce 1546 images. Summarily, we define ten categories including *Standard Splice, V-worn Splice, Standard Knuckle, Misinstalled Knuckle, Standard K-Line, Twisted K-Line, Standard Dropper-1, Loosen Dropper, Standard Dropper-2, Broken Dropper.* The detailed constituents of this dataset as Table II shows.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 4. An example of railway infrastructure defects categories. (a) and (b) represent Knuckle, (c) and (d) represent Kline, (e) and (f) represent Dropper1, and (g) as well as (h) represent Dropper2. The detailed parts to distinguish defects are labeled with red boxes.

In Figure 2, we give examples of Splice as well as defects samples. For an intuitive understanding of the data set, we list examples of the remaining categories in Figure 4.

B. Experiment Setup

For railway infrastructure defect data, we first resize images into a uniform size with 224×224. Then split the whole dataset evenly into two parts as the train set and test set randomly that 773 images are for training and the rest are for testing. During the training stage, we randomly crop, rotate and randomly horizontal flip train images to augment our train sets.

In our experiment, we compare three kinds of methods: hand-craft machine learning based classification, classic deep CNNs model and deep bilinear methods.

For the hand-craft machine learning based algorithm, we use HOG [32] as images feature and SVM [33] as the classifier. In HOG extraction, we set cell size as 8×8 pixels and block size as 2×2 pixels which are same as the default values in the original paper. By changing the size of the input image, we get 1764 and 8100 dimensions HOG features. We build linear SVM detectors with LIBSVM [34] on MATLAB R2014b platform. And we choose the optimal parameters c, g for SVM using a traversal way.

For classical deep CNNs methods, we use VGG16 [13] and VGG19 [13] networks as candidate models. It is impossible to train such deep architectures with our small dataset from scratch. Thus we use the pre-trained VGG16 and VGG19 models in the ImageNet [35]. We replace the last layer of both VGG16 and VGG19 networks with our classifier layer for defects recognition. We only fine-tune the last layer.

For a more comprehensive analysis of our algorithm, besides original BCNN models [24], we design a Spatial Transformer Network for VGG networks (STNVGG16 and STNVGG19) referred to [21] that add a STN in front of the VGG networks. In addition, we compare our proposed method with other bilinear models: B-LR models that add our proposed Low-Rank layer to original BCNNs, STNBM models that add STN network in front of BCNN models. In each of above bilinear models, we adopt three types of feature extractors: (16-16) represents two independent VGG16 structures as feature extractors, (19-19) represents two independent VGG19 structures as feature extractors and (16-19) that use a VGG19 and a VGG16 as feature extractors. Summarily, we compare 15 methods with our STABLR models.

TABLE III
DETECTION RESULTS FOR RAILWAY SUPPLY POWER INFRASTRUCTURE DEFECTS DATASET.

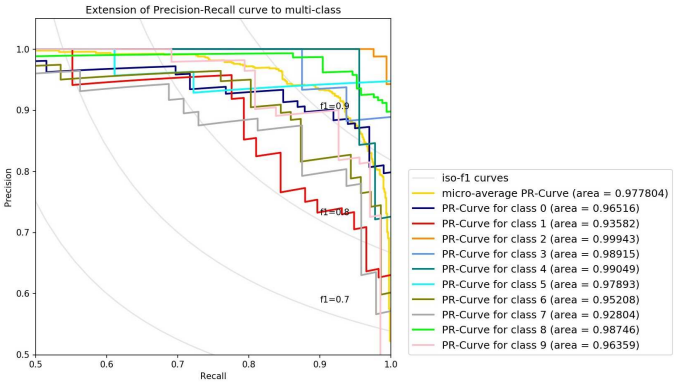
Methods	Feature Dim	Accuracy	Avg_Time(ms)
HOG_SVM	1764	78.33%	2
HOG_SVM	8100	76.39%	10
VGG16	4096	85.81%	39
VGG19	4096	84.51%	40
STNVGG16	4096	87.13%	40
STNVGG19	4096	87.40%	42
BCNN(16-16)	512×512	91.04%	64
BCNN(19-19)	512×512	90.67%	63
BCNN(16-19)	512×512	92.24%	65
B-LR(16-16)	64×512	90.74%	**40**
B-LR(19-19)	64×512	90.40%	**42**
B-LR(16-19)	64×512	91.41%	**41**
STNBM(16-16)	512×512	91.88%	66
STNBM(19-19)	512×512	**91.59%**	68
STNBM(16-19)	512×512	**94.09%**	69
STABLR(16-16)	64×512	**92.67%**	50
STABLR(19-19)	64×512	91.18%	54
STABLR(16-19)	64×512	92.14%	52

Both classical deep networks and bilinear models are implemented using PyTorch. For all deep models, we use Adam optimizer to update networks with Adam' default initial parameters like learning rate and weight decay. Both the training and the testing batch size of all deep models is 8. We then train these models 300 epochs identically. For the sake of fairness, we freeze the feature extract structures for all deep models and fine-tune the remaining parts of the networks. Deep models experiments of our paper are based on two NVIDIA P4000 GPUs and one NVIDIA P5000 GPU.

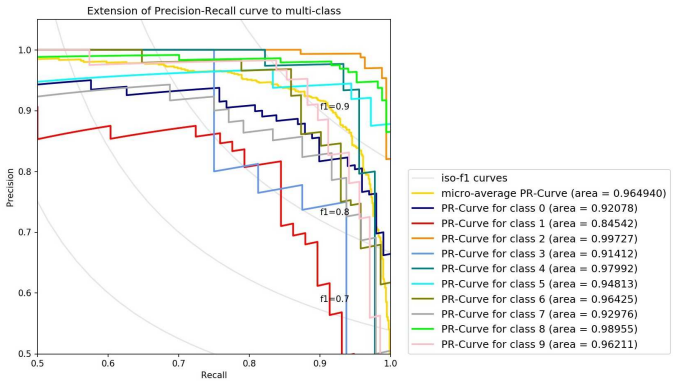
C. Experiments Results Analysis

In this section, we show the results about detection accuracy, average detect time per image, precision, and recall [36] of above models for railway infrastructure dataset.

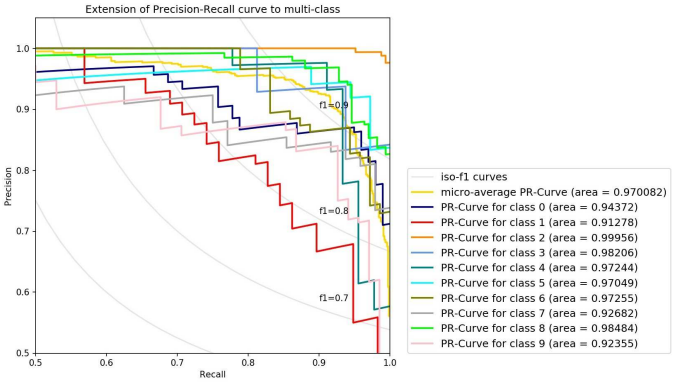
In Table III, Avg_Time means average classification time per image for both methods, all models have been run at least five times to obtain average performance. The training is relatively stable. Compared with hand-craft machine learning based methods, deep learning based methods achieve significant improvement in classification accuracy. For example, the classification accuracy of the classic VGG network increased from 78% to 85%, an increase of about 7 percentage points. It indicates that the deep networks induced features have a better



(a) STABLR_VGG16-VGG16



(b) STABLR_VGG19-VGG19



(c) STABLR_VGG16-VGG19

Fig. 5. P-R curves for all classes in STABLR models.

discriminative capacity than traditional hand-craft features. We also observe that simply adding the dimensions of the hand-craft features does not significantly improve the classification accuracy, but may reduce the accuracy as Table III shows. In addition, BCNNs models outperform classical DCNN methods in test accuracy with an improvement from 87.13% for VGG16 to 92.24% for BCNN(16-19) which indicates that B-CNN models can obtain more subtle features than classical DCNNs. For BCNN models, the fusion of heterogeneous models seems better than the fusion of homogeneous models, this may be caused by that heterogeneous networks can capture more different image features than heterogeneous networks, therefore the bilinear pooled features could have stronger fine-grained classification ability.

It can be observed that the Low-Rank layer will make a small reduction of classification performance that the accuracy of B-LR models is approximately one percent lower than the BCNN models. After the STN is integrated, the performance of entire networks is improved that STNVGG models outperform VGG models and STNBM models outperform BCNN models. From Table III, STNBM(16-19) achieves the best recognition performance among all (16-19) bilinear models, STNBM(19-19) achieves best performance among all (19-19) bilinear models and so is STABLR(16-16) model. We found that our STABLR can achieve a similar performance as STNBM with 8 times fewer feature dimensions.

Compared with recognition time for per image, HOG_SVM is the fastest methods with 10 ms per image or less due to the relatively low complexity of SVM models contrast to deep models. VGG models are faster than most bilinear models except for B-LR models. B-LR model is 20 ms faster than BCNN and STNBM models. STNBM models are the slowest models among all methods owing to the huge network parameters. Our STABLR model is 10 ms faster than STNBM and BCNN models with a good classification performance. It indicates that STABLR is the most effective model with good classification performance as well as high recognition speed.

Two important indicators for evaluating the performance of defect detection models are precision and recall. We first use micro-average precision and recall scores [37] to validate our STABLR models for multi-class classification. From Figure 5, micro-average precision scores of STABLR models are 0.977804, 0.964940, 0.970082 for VGG16-VGG16, VGG19-VGG19 and VGG16-VGG19, respectively, which validate the high performance of our methods.

More detailed, we give Precision-Recall (P-R) curves of each class in railway dataset using 1-vs-all strategy in Figure 5. It can be observed that in each class, precision and recall scores are close to 1.0 thus indicate the effectiveness of our STABLR models. We also notice that class 0 with dark blue line and class 1 with the red line which corresponds to Splice equipment are the hardest class to discriminate as we mentioned before. This type of flaws has a subtle variation compared to standard equipment. Notice that the iso-f1 curve presents all possible standard F1 scores which a higher score of F1 means a better model. All STABLR models have F1 scores above 0.8 and VGG16-VGG16 model gets the highest F1 score.

IV. Conclusion

In this paper, we propose a novel railway power supply infrastructure defects detection method STABLR. We use Spatial Transformer Network to learn the invariant representation of the dataset and adopt a simple but effective low-rank approximation method to reduce the dimension of original BCNN' activation features. We convert detection into recognition task and construct a railway infrastructure defects dataset. More importantly, this paper is the first work

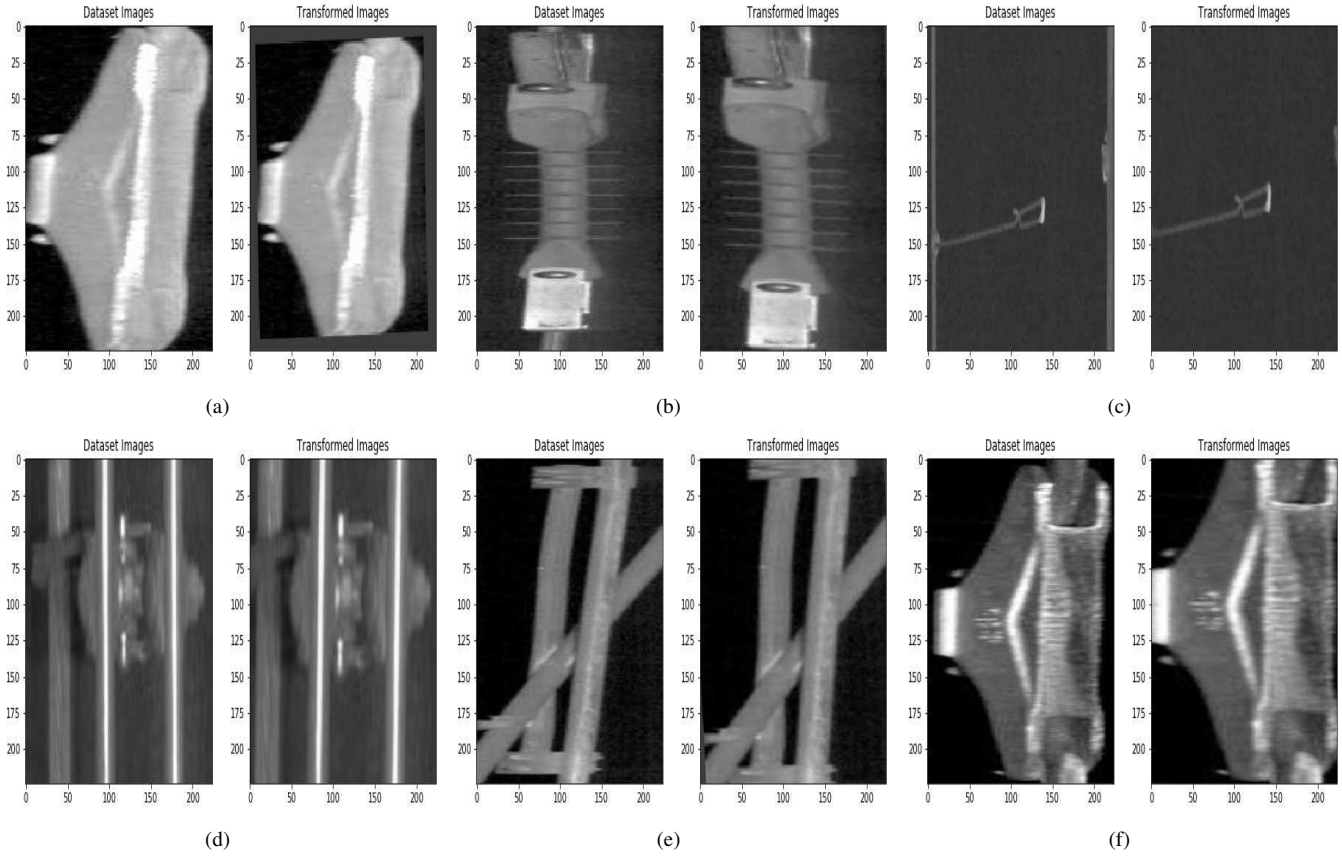


Fig. 6. Some samples that Spatial Transformer Networks produce.

that applies the fine-grained bilinear CNNs model to railway infrastructure defects detection problem. Experimental results have shown the effectiveness and high performance of the proposed method in terms of both accuracy and speed.

ACKNOWLEDGMENT

The authors greatly appreciate the financial support from the Rail Manufacturing Cooperative Research Centre (funded jointly by participating rail organisations and the Australian Federal Governments Business-Cooperative Research Centres Program) through Project R3.7.3 - Rail infrastructure defect detection through video analytics. The authors would like to thank Sydney Train Maintenance Center for generously providing all images, professional descriptions of defects and their particular useful insights on railway data.

REFERENCES

[1] R. Marani, D. Palumbo, U. Galietti, E. Stella, and T. D'Orazio, "Automatic detection of subsurface defects in composite materials using thermography and unsupervised machine learning," in *IEEE International Conference on Intelligent Systems (IS)*, 2016, pp. 516–521.

[2] H. Jia, Y. L. Murphey, J. Shi, and T.-S. Chang, "An intelligent real-time vision system for surface defect detection," in *IEEE International Conference on Pattern Recognition (ICPR)*, vol. 3, 2004, pp. 239–242.

[3] H. Huang, C. Hu, T. Wang, L. Zhang, F. Li, and P. Guo, "Surface defects detection for mobilephone panel workpieces based on machine vision and machine learning," in *IEEE International Conference on Information and Automation (ICIA)*, 2017, pp. 370–375.

[4] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mechanical Systems and Signal Processing*, vol. 72, pp. 303–315, 2016.

[5] C. Lu, Z.-Y. Wang, W.-L. Qin, and J. Ma, "Fault diagnosis of rotary machinery components using a stacked denoising autoencoder-based health state identification," *Signal Processing*, vol. 130, pp. 377–388, 2017.

[6] S. Faghih-Roohi, S. Hajizadeh, A. Núñez, R. Babuska, and B. De Schutter, "Deep convolutional neural networks for detection of rail surface defects." in *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 2584–2589.

[7] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.

[8] H.-F. Ng, "Automatic thresholding for defect detection," *Pattern Recognition Letters*, vol. 27, no. 14, pp. 1644–1649, 2006.

[9] C.-h. Chan and G. K. Pang, "Fabric defect detection by fourier analysis," *IEEE Transactions on Industry Applications*, vol. 36, no. 5, pp. 1267–1276, 2000.

[10] M. J. Lowe, D. N. Alleyne, and P. Cawley, "Defect detection in pipes using guided waves," *Ultrasonics*, vol. 36, no. 1-5, pp. 147–154, 1998.

[11] A. Kumar and G. K. Pang, "Defect detection in textured materials using gabor filters," *IEEE Transactions on Industry Applications*, vol. 38, no. 2, pp. 425–440, 2002.

[12] A. Krizhevsky, "One weird trick for parallelizing convolutional neural networks," *arXiv preprint arXiv:1404.5997*, 2014.

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

[14] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770–778.

[15] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time

object detection with region proposal networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 91–99.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 779–788.

[17] N. Zhang, J. Donahue, R. Girshick, and T. Darrell, "Part-based r-cnns for fine-grained category detection," in *European Conference on Computer Vision (ECCV)*, 2014, pp. 834–849.

[18] D. Lin, X. Shen, C. Lu, and J. Jia, "Deep lac: Deep localization, alignment and classification for fine-grained recognition," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1666–1674.

[19] Y. Peng, X. He, and J. Zhao, "Object-part attention model for fine-grained image classification," *IEEE Transactions on Image Processing*, vol. 27, no. 3, pp. 1487–1500, 2018.

[20] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear cnn models for fine-grained visual recognition," in *IEEE International Conference on Computer Vision (ICCV)*, 2015, pp. 1449–1457.

[21] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in Neural Information Processing Systems*, 2015, pp. 2017–2025.

[22] Y. Cui, F. Zhou, J. Wang, X. Liu, Y. Lin, and S. Belongie, "Kernel pooling for convolutional neural networks," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, no. 2, 2017, p. 7.

[23] H. Chen, J. Wang, Q. Qi, Y. Li, and H. Sun, "Bilinear cnn models for food recognition," in *IEEE International Conference on Digital Image Computing: Techniques and Applications (DICTA)*, 2017, pp. 1–6.

[24] T.-Y. Lin, A. RoyChowdhury, and S. Maji, "Bilinear convolutional neural networks for fine-grained visual recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 6, pp. 1309–1322, 2018.

[25] Y. Gao, O. Beijbom, N. Zhang, and T. Darrell, "Compact bilinear pooling," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 317–326.

[26] S. Kong and C. Fowlkes, "Low-rank bilinear pooling for fine-grained classification," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, pp. 7025–7034.

[27] I. Jolliffe, "Principal component analysis," in *International Encyclopedia of Statistical Science*, 2011, pp. 1094–1096.

[28] T.-Y. Lin and S. Maji, "Improved bilinear pooling with cnns," *arXiv preprint arXiv:1707.06772*, 2017.

[29] K. Chatfield, K. Simonyan, A. Vedaldi, and A. Zisserman, "Return of the devil in the details: Delving deep into convolutional nets," *arXiv preprint arXiv:1405.3531*, 2014.

[30] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2818–2826.

[31] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-resnet and the impact of residual connections on learning." in *National Conference of the American Association for Artificial Intelligence (AAAI)*, vol. 4, 2017, p. 12.

[32] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, 2005, pp. 886–893.

[33] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[34] C.-C. Chang and C.-J. Lin, "Libsvm: a library for support vector machines," *ACM Transactions on Tntelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.

[35] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[36] D. M. Powers, "Evaluation: from precision, recall and f-measure to roc, informedness, markedness and correlation," 2011.

[37] V. Van Asch, "Macro-and micro-averaged evaluation measures [[basic draft]]," *Belgium: CLiPS*, 2013.