https://doi.org/10.2352/EI.2023.35.5.IRIACV-321

Tire Defect Detection with Limited Annotation

Chih-Hui Ho[1], Ziheng Huang[1], NaYeon Kim[2], YouSuk Bae[2], Nuno Vasconcelos[1]
[1] Department of Electrical and Computer Engineering, University of California San Diego, U.S.
[2] Department of Computer Engineering, Tech University of Korea, Korea

Abstract

Tire defect detection has significant industrial value and has been a research topic in both academia and industry. Despite its importance, prior works does not consider the practical manufacturing circumstances, where there is only limited annotation for the defect. Such limitation hinders the prior works from deploying to the real-world system. To address the problem of Tire Defect Detection with Limited Annotation (TTDLA), we proposed a novel framework, denoted as tire defect detection with Self-Supervision and Synthetic data (or S3). S3 first uses self-supervised learning to train the encoder without using any labeled data in the pretraining stage. The encoder is then adopted as the encoder of the Faster-RCNN detector in the fine-tuning stage. In addition, we proposed an algorithm to generate synthesized image by pasting defects randomly onto the regular image. Experiments demonstrate that both self-supervised learning and synthesized data boost the performance of the detector under TTDLA scenario.

Keyword: Tire Defect Detection, Anomaly Detection, Self-Supervised Learning, Industrial Defect Detection

Introduction

Industrial defect detection is an important problem for the advancement of automation in industry, which has been applied to multiple industrial products, typically involving steel or metal surfaces [24, 17, 9, 13, 20, 28, 27, 2]. Defect detection approaches are typically customized for different factory environments and downstream applications. In this work, we consider the problem of tire defect detection (TDD). This is an important problem due to the ubiquity of tire manufacturing, an ever-increasing demand for tires, and the essential role that tires play in aspects such as the security of both traditional and electric vehicles.

Despite the importance of TDD, only a few research works [26, 31, 7] have addressed this problem. These prior works mainly consider images that contain single defects, which is not always a realistic assumption for real world applications. In this paper, we instead consider the problem of TDD in the context of images containing one or more defects. Technically, while prior formulations of TDD resemble the "classification" task, the problems considered in this work are more like the "detection" task. The goal is to detect all defects in the input image, instead of deciding whether the image contains a defect.This difference makes the TDD problems now considered more challenging. Furthermore, while prior works demonstrate the possibility of applying deep neural networks to the TDD problem, most assume the existence of a large training set [26, 31]. However, tire defects are uncommon in practical tire manufacturing and the identification and annotation of these defects is time consuming. This makes it difficult to curate a large dataset for each defect type in practice.

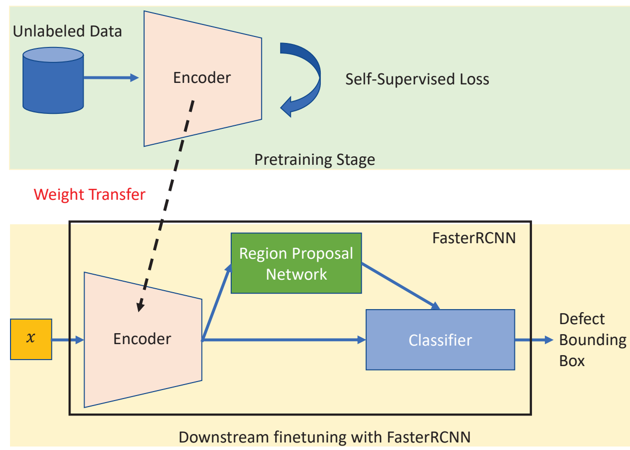


Figure 1. *Overview of the proposed TDDLA approach, using Self-Supervision and Synthetic data (S3). The top part of the figure reports to the pretraining stage, while the bottom part shows the finetuning stage. In the latter, the network encoder is initialized with the weights obtained at the end of pretraining.*

In this work, we specifically study the TDD problem with Limited Annotation (TDDLA), where there is a large imbalance between normal and abnormal tire patterns.

We propose a novel detection framework for TDDLA, based on *Self-Supervision and Synthetic data* (S3). As shown in Fig. 1, S3 adopts the Faster-RCNN [23] architecture, but is trained in two stages: a pretraining and a fine-tuning stage. TPre-training aims to learn a general feature encoder using unlabeled data. This is performed by optimizing a self-supervised learning loss on an unlabeled dataset. The weights of the feature encoder learned in the first stage are then used to initialize the training of the Faster-RCNN detection model in the second stage, which has access to limited annotation of defect locations. Beyond proposing this multi-stage training scheme, we investigate the effectiveness of using synthetic data for training. Experiments demonstrate that both the introduction of the pretraining stage and the use of synthetic data boost TDDLA performance.

Overall, this work makes three contributions. First, we introduce the problem of Tire Defect Detection with Limited Annotation (TDDLA) and show that this problem can be applied to multiple industrial applications. Second, we propose a novel training framework, denoted as S3, to address the problem of TDDLA, where only limited tire defect annotation is available for training. S3 leverages self-supervised pertaining and data synthesis to overcome the limited data issue. Finally, extensive experiments and ablation studies are presented to validate the effectiveness of

S3 on the proposed TDD dataset.

Related Work

In this section, we discuss prior works on TTD and Self-supervised Learning.

Tire Defect Detection

TDD is a problem that has long been considered and has significant interest for industry. Prior works [26, 31] mainly consider the scenario where there is a single defect per image. This reduces the problem to one of image classification. Furthermore, in the datasets used in these works, the single defect occupies a large portion of the image. [26] adopts a fully convolutional network (CNN) for detecting the region that contains the defect and produces a segmentation map for it. [31] uses a CNN to extract deep features, which are then used to discriminate normal image patches from abnormal patches that contains defects. These works do not consider the low training data regime, where defect annotations are insufficient to train a model roubust enough for ptractical deployment. Similar to our work, [7] addresses the limited samples issue. This is done by using an ensembles of CNNs and taking their joint classification outputs to classify the defect image. Unlike [7], the detection task is considered in this work and self-supervised learning is leveraged to address the limited data issue.

Self-supervised Learning

Self-supervised learning [11] aims to learn a generic feature representation that can be applied to various downstream tasks. The learning of generic features is conducted during a pretraining stage, where no label from the downstream tasks is used. Self-supervised training, without data labels, is implemented by designing a pretext task to optimize the model [1, 22, 15, 29, 30, 14]. Various pretext tasks have been proposed over the past few years, ranging from solving image puzzles [12, 19], to predicting masked patches [22, 8, 18] or learning invariant features for different color channels of the same image [25]. One of the most popular pretext tasks is to learn an invariant representation for different augmentations (i.e. views) of an input image. Typically, these views are created by applying standard data transformations (random cropping, rotation, or color jittering) to the image. This idea has motivated multiple recent methods. For example, SimCLR [4, 5] first applies data augmentations to each input image and obtains two set of augmented views per image. Training is then based on a loss function that encourages the feature vectors extracted from the two views by a deep network to be similar. To further increase the diversity of the augmented features, Moco [10, 6] uses a dictionary that stores the features of augmented views from previous epochs. Both variants of the idea highlight the importance of learning invariant representation from diverse augmentations. In this work, we proposed an alternative augmentation method by using synthesized data that is tailored for TDD for self-supervised learning.

Tire Defect Detection with Limited Annotation

In this section, the task of Tire Defect Detection with Limited Annotation (TDDLA) is introduced. As shown in Fig. 2, we consider the problem where each tire image contains multiple "threads", which appear as white dots on the tire surface. While most of the threads are normal, those marked with blue rectangles are abnormal, indicating a defect in the manufacturing of the tire. These annotations are produced manually by a tire expert. As shown in the images of Fig. 2, the number of normal and abnormal threads is highly imbalanced,. Moreover, since the threads are densely distributed over the tire surface, it is challenging to annotate every thread. Hence, annotations are not produced for normal threads, only the bounding boxes of abnormal threads are available. The task of TDDLA is to localize these abnormal threads. This is similar to object detection, but more difficult than the standard object detection problem, since the abnormal threads to be detected are surrounded by many normal threads, for which annotations are not available. An analogous problem would be to train an object detector to detect female pedestrians, on street scenes heavily populated by male pedestrians, and no labels for the latter.

Method

In this section, the proposed defect S3 detection framework is introduced. Since TTDLA is a challenging task, we consider two possible solutions. The first is to augment the existing dataset with synthetic data. The second is to train a generalizable detector encoder by self-supervised learning from unlabeled data only. The entire framework is illustrated in Fig. 1 and more details will be elaborated below.

Architecture

A standard Faster-RCNN [23] with ResNet50 backbone is adopted as the detection model for thread detection. The Faster-RCNN contains three submodules: an encoder, a region proposal network, and a classifier (see bottom plot of Fig. 1). The encoder extracts features from the input image and its output contains the local feature vectors for each of the threads. The region proposal network identifies potential thread candidates among the encoder features. Finally, the classifier classifies each of the proposed candidates, predicting whether it is a defect.

Algorithm 1 Procedure to generate synthetic crops.

1. Save all images in a dictionary D.
2. For a given a synthetic/regular image ratio r, compute the number of synthesized images k.
3. Randomly select k images with no defects from dictionary D.
4. Paste 1 to 4 randomly selected defects in random locations of those k images. Save the locations as ground truth for defects in those k images.

Training with synthetic data

While prior works [26, 31] assume that sufficient defect data is available for training, this is usually unrealistic in the manufacturing setting. For example, only 43% of images in our dataset even contain thread defects. On average, each image contains more than 500 threads and there are less than 5 defects per image. This is a critical issue, since the deep learning model requires a large amount of training data. Insufficient data will cause the model to overfit to the training images and perform poorly on images unseen during training. Yet, curating a large dataset with

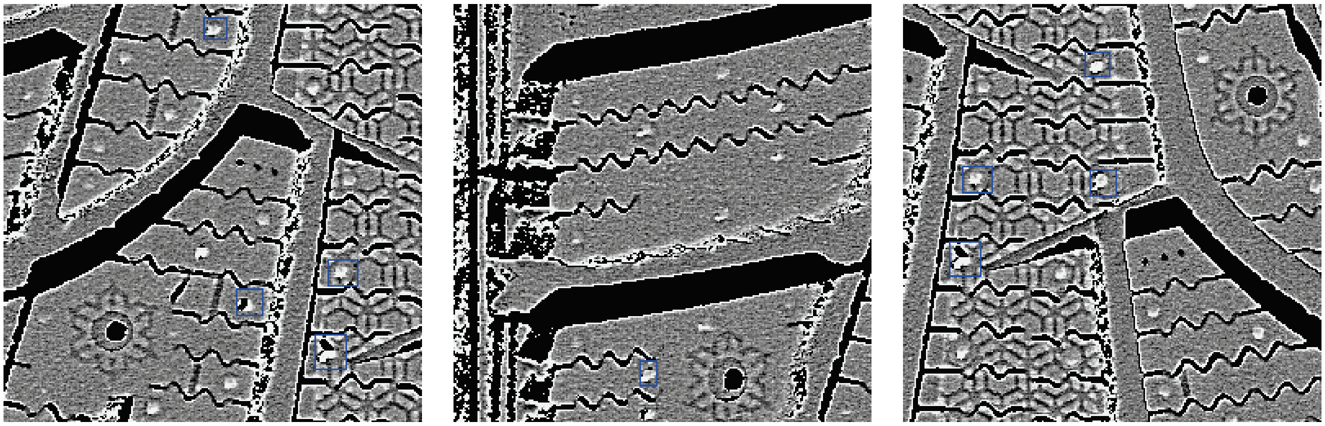
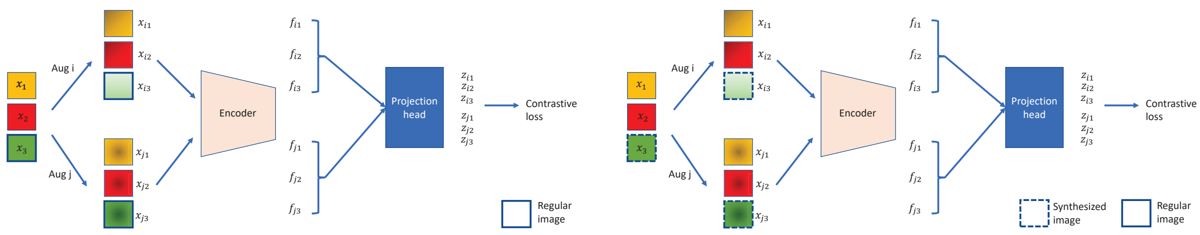


Figure 2. *Examples of the tire defects. For each image, the white dots are the threads on the tire surface. Some of the white dots (marked in blue rectangles) are abnormal and considered as defect, while the rest are normal threads.*



(a) (b)

Figure 3. *(a) SimCLR [4] architecture (b) SimCLR [4] architecture with synthetic data. Solid lines indicate regular images, while the dashed line indicate synthetic ones.*

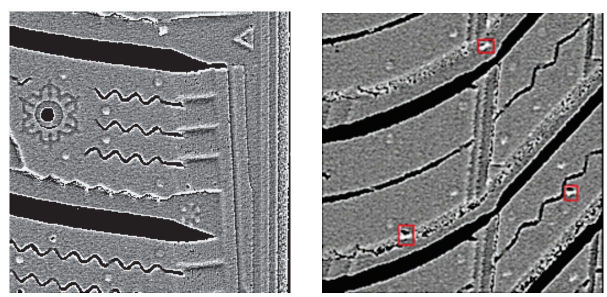
defect annotations is challenging, because (1) defects are rare in industrial manufacturing and (2) defect annotation requires domain expertise, which is not available in the public crowdsourcing platforms commonly leveraged to label large-scale datasets. The requirement of time and manual effort from experts makes the annotation of large datasets impractically costly.

Since it is challenging to curate a large dataset, we pursue the alternative strategy of augmenting the current dataset. For this, we propose to synthesize the input image. A similar approach [16] has shown promising results on the publicly available industrial defect dataset MVTec-AD [3]. Our method shares the spirit of [16] but, unlike this work, we rely on few annotated defects to start with. To ensure the quality of the synthesized images, we proposed to leverage these few annotated defects using the synthesized dataset generation procedure described in Algorithm 1.

After application of Algorithm 1, the synthesized images can contain multiple defects at random locations. As shown in Fig. (a), the synthesized output resembles a regular image, where the pasted defects cannot be distinguished with ease. Fig. (b) further shows bounding boxes of pasted defects. Note that since the defects are pasted on the synthesized image, the defect location is labeled automatically, no human effort is required.

Self-Supervised Pretraining

Self supervised learning (SSL) [11] is commonly used in circumstances where data labeling is challenging. The goal is to train a model using a large unlabeled training dataset with a proxy task. This pretrained model is then fine-tuned on the downstream task of interest, for which only little annotated data is available.



(a) (b)

Figure 4. *Example of (a) a synthesized crop and (b) a synthesized crop with bounding box annotations. Note that the annotations are obtained without human labeling effort.*

Motivated by the success of SSL, we propose to first pretrain the encoder (ResNet50 backbone) of the Faster-RCNN using state-of-the-art SSL methods. This encoder is then fine tuned on the downstream TDD task.

Following recent advances in SSL, we select SimCLR [4] as SSL approach. Take 3 input images x_1, x_2 and x_3 for example. As depicted in Fig. 3(a), SimCLR takes these images and applies two different data augmentations to each, resulting in two sets of augmented images $\{x_{i1}, x_{i2}, x_{i3}\}$ and $\{x_{j1}, x_{j2}, x_{j3}\}$. These two augmented image sets are then forwarded into the encoder (i.e. ResNet50) for extracting the corresponding feature vectors $\{f_{i1}, f_{i2}, f_{i3}\}$ and $\{f_{j1}, f_{j2}, f_{j3}\}$, respectively. Finally, these vectors are passed through a projection head, composed by a series

SSL	Syn. data for SSL	Syn. data for finetuning	Avg. Precision	Avg. Recall	Avg. F1	Best F1
			0.7972	0.7927	0.7949	0.8212
✓			0.7957	0.8070	0.8013	0.8188
✓	✓		0.7875	0.8276	0.8071	0.8216
✓	✓	✓	**0.8432**	**0.8347**	**0.8389**	**0.8552**

Table 1. Quantitative results of different TDDLA approaches. Checkmarks indicate the components used by each approach, beyond the plain Faster-RCNN baseline (top row). Results in bold highlight the best performance.

of fully connected layers.

The projected embeddings $\{z_{i1}, z_{i2}, z_{i3}\}$ and $\{z_{j1}, z_{j2}, z_{j3}\}$ are used to compute the contrastive loss

$$L_{contrastive} = \frac{z_{i1}^T z_{j1}/\tau}{\sum_{k=1}^{3} z_{ik}^T z_{jk}/\tau}, \tag{1}$$

where τ is a smoothing factor. This loss encourages pairs of feature vectors extracted from the same image to be close in the embedding space, while those extracted from different images should be far away. This endows the embedding with a metric structure (similar images are mapped to neighboring feature vectors, different images to vectors that are far apart). The downstream task can then be learned in this space using little training data, since the metric structure of the space is already learned. In many cases, the downstream task can even be implemented by a simple nearest neighbor feature classifier. While the application of self-supervised learning to the limited data regime has shown to be successful mostly for classification tasks, we demonstrate that the technique can be equally applied to detection tasks such as TDD.

Beyond this, we further investigate the benefits of adding synthesized data to the self-supervised learning pipeline. As illustrated in Fig 3(b), regular images can be combined with synthetic images, marked with a dashed line in the figure. Experiments demonstrate that adding synthesized data to the SSL pipeline boosts TDD performance.

Experiment

In this section, the details of the dataset are discussed and experimental results of the TDD task are presented.

Dataset and Metric

The dataset contains 567 images, only 324 of which contains defects. To train the detection model, we discard images without defects and use an 80 to 20 split for train and test set. This means that 243 images are used as training set. All the remaining images are used as test set. Evaluation is based on the precision, recall and F1 score metrics popular in the detection literature, using IOU@0.5. Since the test set is relatively small, results have high variance. We report the average result over at least five training trials.

Implementation Details

All experiments are conducted using Pytorch [21] on a Nvidia Titan Xp GPU with Intel Xeon CPU E5-2630. For the self-supervised pretraining stage, the encoder is trained for 200 epochs, batch size is set 128 and τ is set to 0.5. We use random resized crop, random horizontal flip, color jitter and random gray scale for the data augmentations. For the fine-tuning stage, we set the batch size to 4 and train the Faster-RCNN for 100 epochs. In both pretraining and finetunning stages, an SGD optimizer is used with learning rate 0.002, momentum 0.9 and weight decay 0.0005.

Quantitative Results

Table 1 summarizes the benefits of the proposed TTDLA approach. The table presents the average precision, recall, and F1-score of several approaches plus the best F1-score over the five training runs. The top row of the table summarizes the result of simply training the Faster-RCNN on the TDD dataset. This is our baseline and achieves the average F1 score of 0.7949. By performing SLL pretraining, the average F1 score increases to 0.8013. By adding the synthetic data to the pretraining stage, we observe another 0.58% gain in terms of F1 score. Finally, when synthetic data is used during both pretraining and fine tuning, the gain over the baseline is around 3.8%. Note that the best F1 score achieved by S3 with both SSL and synthetic data is 0.8552. These results validate the hypothesis that both SSL pretraining and the addition of synthetic examples can help mitigate the challenges of limited training data and are useful for the TDDLA task.

Ablation of Synthetic/Regular Ratio

To further investigate the effects of synthetic data, we first train a detector detector without synthetic defects. As shown in Table 2, this has an average F1 score of 0.7949. We then ablate the synthetic/regular ratio r and find that $r = 2$ leads to the best F1 score. Compared to the results from the first row of Table 2, adding synthetic data improves the F1 score by 3.5 points on average. This indicates the use of synthetic data can potentially alleviate the issue of insufficient training data and improve the detection results. Note that synthetic data is only used during the training phase, not the testing phase.

Qualitative Results

Fig. 5 shows some quantitative results of the proposed framework on three inputs image. It can be observed that the dataset now used is unlike previous ones, where images only contain a single defects. The first row shows an example where all detected defects appears in the ground truth. The second row shows an example with a false positive detection. The last row shows another example, where the ground truth and the prediction mismatches, differ by both a false positive and a false negative (missed) detection.

Conclusion and Future Work

In this work, we consider the problem of tire defect detection with limited annotation (TDDLA) scenario. To address this issue, we proposed a novel training framework suitable for the TDDLA problem. The proposed framework contains 2 stages, which are (1) the pretraining stage and (2) the finetuning stage. The former assists the feature encoder to learn a more generic feature, while the later uses the limited annotation to predict defect

Syn./Regular Ratio r	Precision	Recall	F1	F1 (Best)
No syn. image	0.7972	0.7927	0.7949	0.8212
2	0.8204	**0.8379**	**0.8291**	**0.8533**
1.5	0.8147	0.7968	0.8057	0.8271
1	0.8283	0.8197	0.8240	0.8444
0.5	**0.8369**	0.8037	0.8200	0.8392
1/3	0.8291	0.7740	0.8006	0.8235
1/5	0.8138	0.8322	0.8229	0.8345

Table 2. Ablation study of the synthetic/regular image ratio r. Results in bold highlight the best performance. It can be observed that r=2 leads to better F1 result.

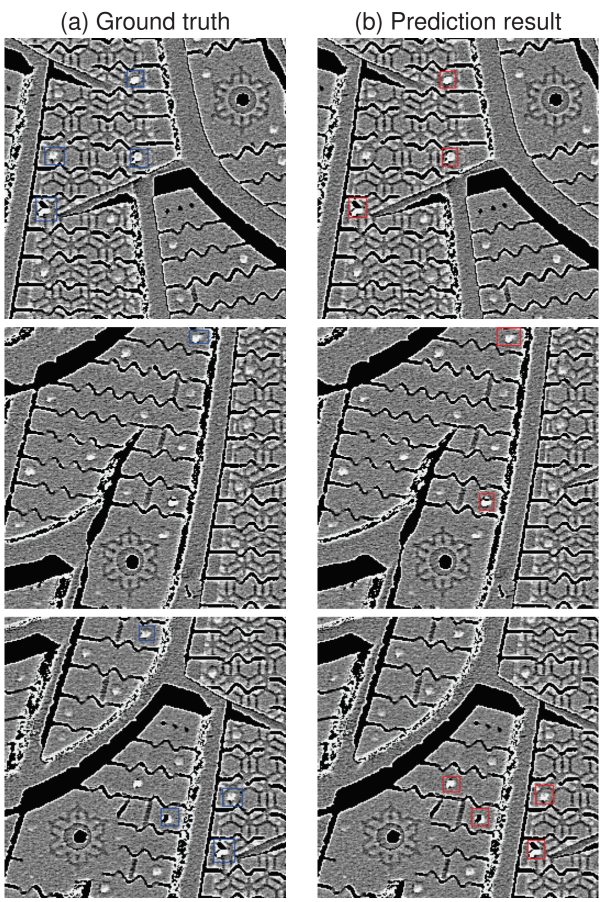


Figure 5. *(a) Ground truth location of the defects. (b) Prediction result of the proposed model.*

location based on the initial weight from the former pretraining stage. To further augment the limited training data, an algorithm is proposed to produce synthesized defect on the image. Extensive experiments demonstrate the performance of the proposed framework and the ablation study validates the effectiveness of each proposed techniques. We hope this work could contribute a new research direction to the literature of tire defect detection with limited annotation and inspire future works that investigate more defect types.

Acknowledgement

This research was financially supported by the Ministry of Science and ICT and Institute for Information & Communication Technology Planning & Evaluation (IITP) through the Grand ICT program.

References

[1] Pulkit Agrawal, João Carreira, and Jitendra Malik. Learning to see by moving. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 37–45, 2015.

[2] Xiaolong Bai, Yuming Fang, Weisi Lin, Lipo Wang, and Bing-Feng Ju. Saliency-based defect detection in industrial images by using phase spectrum. *IEEE Transactions on Industrial Informatics*, 10(4):2135–2145, 2014.

[3] Paul Bergmann, Michael Fauser, David Sattlegger, and Carsten Steger. Mvtec ad — a comprehensive real-world dataset for unsupervised anomaly detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9584–9592, 2019.

[4] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

[5] Ting Chen, Simon Kornblith, Kevin Swersky, Mohammad Norouzi, and Geoffrey Hinton. Big self-supervised models are strong semi-supervised learners. *arXiv preprint arXiv:2006.10029*, 2020.

[6] Xinlei Chen, Haoqi Fan, Ross Girshick, and Kaiming He. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020.

[7] Xuehong Cui, Yun Liu, Yan Zhang, and Chuanxu Wang. Tire defects classification with multi-contrast convolutional neural networks. *Int. J. Pattern Recognit. Artif. Intell.*, 32:1850011:1–1850011:17, 2018.

[8] Carl Doersch, Abhinav Kumar Gupta, and Alexei A. Efros. Unsupervised visual representation learning by context prediction. *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 1422–1430, 2015.

[9] Hao Feng, Zhiguo Jiang, Fengying Xie, Ping Yang, Jun Shi, and Long Chen. Automatic fastener classification and defect detection in vision-based railway inspection systems. *IEEE Transactions on*

Instrumentation and Measurement, 63(4):877–888, 2014.

[10] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

[11] Longlong Jing and Yingli Tian. Self-supervised visual feature learning with deep neural networks: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43:4037–4058, 2021.

[12] Dahun Kim, Donghyeon Cho, Donggeun Yoo, and In So Kweon. Learning image representations by completing damaged jigsaw puzzles. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 793–802, 2018.

[13] A. Kumar and G.K.H. Pang. Defect detection in textured materials using gabor filters. *IEEE Transactions on Industry Applications*, 38(2):425–440, 2002.

[14] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. Learning representations for automatic colorization. In *ECCV*, 2016.

[15] Hsin-Ying Lee, Jia-Bin Huang, Maneesh Singh, and Ming-Hsuan Yang. Unsupervised representation learning by sorting sequences. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 667–676, 2017.

[16] Chun-Liang Li, Kihyuk Sohn, Jinsung Yoon, and Tomas Pfister. Cutpaste: Self-supervised learning for anomaly detection and localization. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9659–9669, 2021.

[17] Xiaoli Li, S. K. Tso, Xin-Ping Guan, and Qian Huang. Improving automatic detection of defects in castings by applying wavelet technique. *IEEE Transactions on Industrial Electronics*, 53(6):1927–1934, 2006.

[18] Terrell N. Mundhenk, Daniel Ho, and Barry Y. Chen. Improvements to context based self-supervised learning. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9339–9348, 2018.

[19] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

[20] Jong-Hwan Oh, Woo-Seob Kim, Chan-Ho Han, and Kil-Houm Park. Defect detection of tft-lcd image using adapted contrast sensitivity function and wavelet transform. *IEICE Trans. Electron.*, 90-C:2131–2135, 2007.

[21] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*, 2019.

[22] Deepak Pathak, Philipp Krähenbühl, Jeff Donahue, Trevor Darrell, and Alexei A. Efros. Context encoders: Feature learning by inpainting. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2536–2544, 2016.

[23] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[24] Yi Sun, Peng Bai, Hong yu Sun, and Ping Zhou. Real-time automatic detection of weld defects in steel pipe. *NDT & E International*, 38(7):522–528, 2005.

[25] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XI*, page 776–794, Berlin, Heidelberg, 2020. Springer-Verlag.

[26] Ren Wang, Qiang Guo, Shanmei Lu, and Caiming Zhang. Tire defect detection using fully convolutional network. *IEEE Access*, 7:43502–43510, 2019.

[27] Moe Win, A. R. Bushroa, M. A. Hassan, N. M. Hilman, and Ari Ide-Ektessabi. A contrast adjustment thresholding method for surface defect detection based on mesoscopy. *IEEE Transactions on Industrial Informatics*, 11(3):642–649, 2015.

[28] Lijuan Xu and Qiang Huang. Modeling the interactions among neighboring nanostructures for local feature characterization and defect detection. *IEEE Transactions on Automation Science and Engineering*, 9(4):745–754, 2012.

[29] Mang Ye, Xu Zhang, PongChi Yuen, and Shih-Fu Chang. Unsupervised embedding learning via invariant and spreading instance feature. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6203–6212, 2019.

[30] Richard Zhang, Phillip Isola, and Alexei A Efros. Colorful image colorization. In *ECCV*, 2016.

[31] Yan Zhang, Xuehong Cui, Yun Liu, and Bin Yu. Tire defects classification using convolution architecture for fast feature embedding. *International Journal of Computational Intelligence Systems*, 11:1056, 05 2018.