

## Article

# A New Knowledge-Distillation-Based Method for Detecting Conveyor Belt Defects

Qi Yang <sup>1</sup>, Fang Li <sup>1</sup>, Hong Tian <sup>2</sup>, Hua Li <sup>1</sup>, Shuai Xu <sup>1</sup>, Jiyou Fei <sup>1,\*</sup>, Zhongkai Wu <sup>1</sup>, Qiang Feng <sup>1</sup> and Chang Lu <sup>1,3</sup>

<sup>1</sup> College of Locomotive and Rolling Stock Engineering, Dalian Jiaotong University, Dalian 116028, China

<sup>2</sup> Software Technology Institute of Dalian Jiaotong University, Dalian 116052, China

<sup>3</sup> PLA Army Academy of Artillery and Air Defense, Shenyang 110000, China

\* Correspondence: fjy@djtu.edu.cn

**Abstract:** Aiming to assess the problems of low detection accuracy, poor reliability, and high cost of the manual inspection method for conveyor-belt-surface defect detection, in this paper we propose a new method of conveyor-belt-surface defect detection based on knowledge distillation. First, a data enhancement method combining GAN and copy–pasting strategies is proposed to expand the dataset to solve the problem of insufficient and difficult-to-obtain samples of conveyor-belt-surface defects. Then, the target detection network, the YOLOv5 model, is pruned to generate a mini-network. A knowledge distillation method for fine-grained feature simulation is used to distill the lightweight detection network YOLOv5n and the pruned mini-network YOLOv5n-slim. The experiments show that our method significantly reduced the number of parameters and the inference time of the model, and significantly improves the detection accuracy, up to 97.33% accuracy, in the detection of conveyor belt defects.

**Keywords:** defect detection; data augmentation; model pruning; knowledge distillation



**Citation:** Yang, Q.; Li, F.; Tian, H.; Li, H.; Xu, S.; Fei, J.; Wu, Z.; Feng, Q.; Lu, C. A New Knowledge-Distillation-Based Method for Detecting Conveyor Belt Defects. *Appl. Sci.* **2022**, *12*, 10051. <https://doi.org/10.3390/app121910051>

Academic Editor: Rui Yao

Received: 31 August 2022

Accepted: 4 October 2022

Published: 6 October 2022

**Publisher's Note:** MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



**Copyright:** © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

The mineral transportation conveyor belt is an indispensable piece of equipment in the production and transportation of minerals. During operation, due to the friction and bumping with ore for a long time, which is also affected by factors such as the hardening and aging of the rubber, the surface of the conveyor belt will often appear to have edge loss, surface cracking, holes, covering rubber bulging or skinning, large area wear, deep scratches and other damages. Serious breakage is a precursor to the tearing of the conveyor belt, and if the breakage is not dealt with in time, the breakage will continue to expand with the increase in running time. Finally, destructive tearing can occur, which is a major hazard to the safety of personnel and property [1]. The current traditional manual inspection method requires regular inspection by running the machine at a low speed when the machine is empty, and this inspection method is limited by the time of inspection and the accuracy of the workers' inspection, which makes it difficult to detect damage on the surface of the conveyor belt in a timely, accurate and stable manner.

To solve the problems of traditional manual inspection methods and improve the efficiency, time and reliability of conveyor-belt-damage detection, Wang et al. [2] proposed a scheme for the nondestructive inspection of conveyor belts using X-rays. Yang et al. [3] proposed a method for detecting longitudinal tears in conveyor belts using infrared images, which performs ROI(region of interest) selection and the binarization of the images, and then determines whether an early warning should be issued based on the number of connected. Yang et al. [4] proposed an early warning method for longitudinal tear detection in conveyor belts based on infrared spectral analysis, where the spectral characteristics of the infrared radiation field are used to determine whether there is a risk of longitudinal tear on the conveyor belt. Qiao et al. [5] proposed a longitudinal tear detection method

based on visible charge-coupled devices (CCDs) and infrared CCDs. Combining two CCDs yields a more reliable conveyor belt tear detection method.

In recent years, with the continuous development of deep-learning technology and its great progress in the field of computer vision, target detection models based on deep-learning technology have also achieved better results in conveyor-belt-damage detection tasks. Unlike traditional computer vision methods that focus on processing images using fixed algorithms and processes to extract specific regions of the image, deep-learning-based computer vision techniques can achieve a more accurate, faster, and more intelligent detection by extracting deep image features of the target from a large number of data samples. At present, deep-learning-based target detection algorithms are mainly divided into two-stage detection algorithms represented by Mask R-CNN [6] and Faster R-CNN [7], and single-stage detection algorithms represented by SSD [8], RetinaNet [9] and YOLO [10]. The former is slower because it first generates the candidate frames of the target region, and then classifies each candidate frame, requiring two steps to complete the detection of the target. The latter can predict all the bounding boxes by feeding the images into the network only once, which makes it faster and is more often used in industrial scenarios.

As shown in Table 1, a number of studies have been conducted on conveyor belt detection using deep-learning-based methods. Zhang et al. [11] proposed a conveyor-belt-detection method using EfficientNet to replace YOLOv3's backbone network Darknet53, which achieved 97.26% detection accuracy. Wang et al. [12] proposed a detection model combining BTFPN and YOLOX to achieve 98.45% detection accuracy for conveyor belt damage. To improve the speed of detection of conveyor belt damage, Zhang et al. [13] proposed a lightweight network based on YOLOv4 to improve the detection speed of conveyor belt damage, but its detection accuracy was only 93.22%. Guo et al. [14] proposed a novel multiclassification conditional CycleGAN (MCC-CycleGAN) method for the detection of conveyor-belt-surface damage.

**Table 1.** Summary of the detection effect of some methods.

Methods	mAP@0.5/%	Inference Time/ms	Test Equipment
EfficientNet-YOLOv3 (Zhang et al., 2021)	97.26	23.81	NVIDIA RTX2060s
Yolox-BTFPN (Wang et al., 2022)	98.45	25.6	NVIDIA GTX1660Ti
MobilenetV3-YOLOv4 (Zhang et al., 2021)	93.08	22.71	NVIDIA RTX2060s
MCC-CycleGAN (Guo et al., 2022)	96.88	22.57	NVIDIA RTX3070

At present, deep-learning-based conveyor-belt-detection methods are generally improved in one of the aspects of detection accuracy or speed. When improving detection accuracy, it often leads to a slower detection speed and, conversely, increasing detection speed causes a decrease in detection accuracy. Therefore, improving detection accuracy while also improving detection speed is still a challenge.

Since there is no publicly available high-quality conveyor-belt-defect dataset, it is difficult to obtain a sufficient amount of data to train the deep-learning network, which makes it difficult to guarantee the detection and generalization performance of the model. In addition, due to the performance limitations of the equipment in actual production, how to balance and trade-off between detection accuracy and speed is also an important issue. In this paper, we focus on the above two problems in practical applications and propose a new conveyor-belt-defect detection method based on knowledge distillation.

Section 2 presents our proposed data enhancement method with the improvement of the YOLOv5 model using model pruning and knowledge distillation methods. Section 3 shows the experimental setting with the relevant parameters and the results of the proposed improved method in this paper. Finally, in Section 4, we conclude with a discussion and summary of the methods used in this paper.

## 2. Materials and Methods

In order to detect the conveyor belt defects, this paper designed a new detection method based on knowledge distillation. First, we designed a joint GAN and copy–pasting strategy data-augmentation method for the situation of insufficient samples of conveyor belt defects, which effectively increased the number of samples as well as the richness of sample morphology. For the limited computing performance of industrial equipment, we used the Network Sliming [15] algorithm to prune YOLOv5 and a knowledge distillation method based on fine-grained feature simulation [16] to detect two types of defects, namely conveyor belt scratches and edge defects.

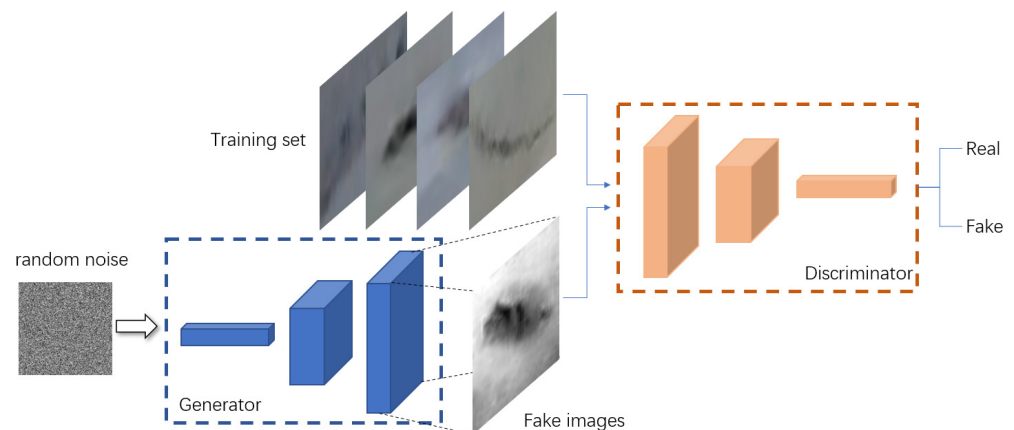
### 2.1. Generative Adversarial Networks

Deep-learning models require a large number of data samples to train the models, and there are few publicly available high-quality datasets in areas such as conveyor-belt-detection inspection, and it is difficult to obtain a sufficient number of samples from field collection. In 2014, Goodfellow et al. [17] proposed GANs (generative adversarial networks), a new solution to the problem of insufficient samples in the dataset. GANs can generate more samples, different from the existing data, to enhance the generalization performance of the model.

As shown in Figure 1, during training, a set of randomly generated noise vectors were first input to the generator to generate a set of false images. Then, the generated false images were input to the discriminator, together with the real images, to judge the authenticity of the images. After iteration, the discriminator could eventually not distinguish whether the images were from the real training samples or generated by the generator, at which time the GAN reached the optimal state. The objective function of the GAN is as follows:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x \sim p_{\text{data}}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))] \quad (1)$$

where  $G$  is the generator and  $D$  is the discriminator.



**Figure 1.** The process of scratch sample generation by GAN.

### 2.2. Knowledge Distillation

The concept of knowledge distillation was first proposed by Hinton [18] et al. However, it is only applicable to the classification task and does not work with the target detection task. Subsequently, Hou [19] et al. proposed a self-attentive distillation algorithm to improve the learning ability of the student network by guiding the shallow network to learn the features of the deep network. Gao [20] et al. proposed a residual knowledge distillation algorithm to introduce a residual module between the teacher network and the student network, but their algorithm is too complex and takes up a lot of computing resources in training. In addition, Chawla [21] et al. proposed a data-free knowledge

distillation method for target detection. Chen [22] et al. proposed a cross-level connection path for teacher and student networks to train the model with very little overhead.

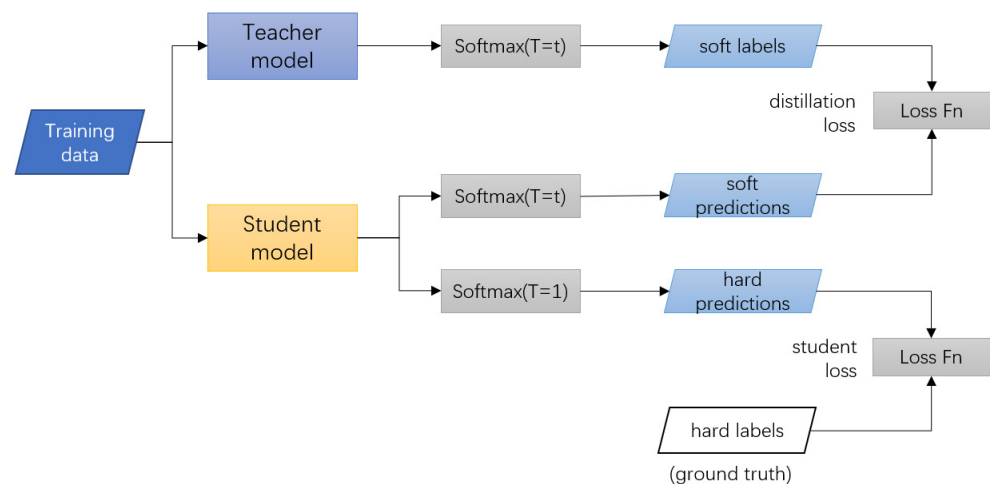
The algorithm flow of knowledge distillation is shown in Figure 2. For the same dataset, the output of the teacher network needed to be calculated first as the soft label, and the soft label contained the implicit relationship between each category. Then, the output of the student network was calculated to generate soft prediction and hard prediction, and distillation loss was calculated between soft prediction and soft label. Hard prediction and true label were calculated as the loss of the student network, and the weighted value of distillation loss and loss of the student network was finally calculated as the final loss function, which is calculated as follows:

$$L_{\text{soft}}(p_s, p_t) = - \sum p_s \log p_t \quad (2)$$

$$L_{\text{hard}}(p_s, p_t) = - \sum p_s \log y \quad (3)$$

$$L_{\text{final}} = \alpha L_{\text{soft}}(p_s, p_t) + (1 - \alpha) L_{\text{hard}}(p_s, y) \quad (4)$$

where  $p_t$  and  $p_s$  are the outputs of the teacher network and the student network,  $y$  is the ground truth and  $\alpha$  is the weighting factor.



**Figure 2.** Process of knowledge distillation algorithm.

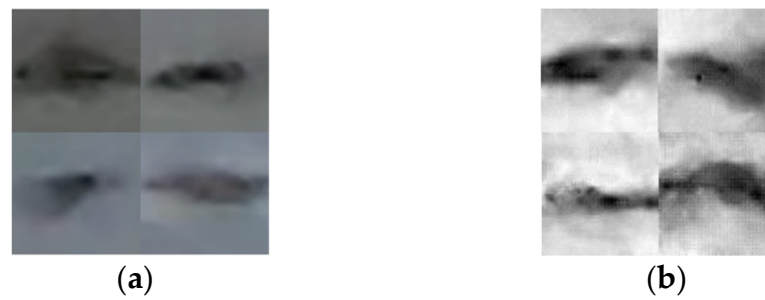
### 2.3. A New Method for Knowledge Distillation Detection of Conveyor Belt Defects Based on YOLOv5

#### 2.3.1. A Data Augmentation Method Combining GAN and Copy-Pasting Strategies

The copy-pasting strategy is a data augmentation method proposed by Ghiasi [23] for instance segmentation techniques, which generates new samples by pasting instance objects of different scales into a new background image for the purpose of data augmentation.

Inspired by this method, we propose a data augmentation method that combines the GAN and copy-pasting strategies. We used the copy-pasting strategy based on the scratch samples generated by the DCGAN [24] and pasted the scratch samples into the background image, using the conveyor belt image as the background image.

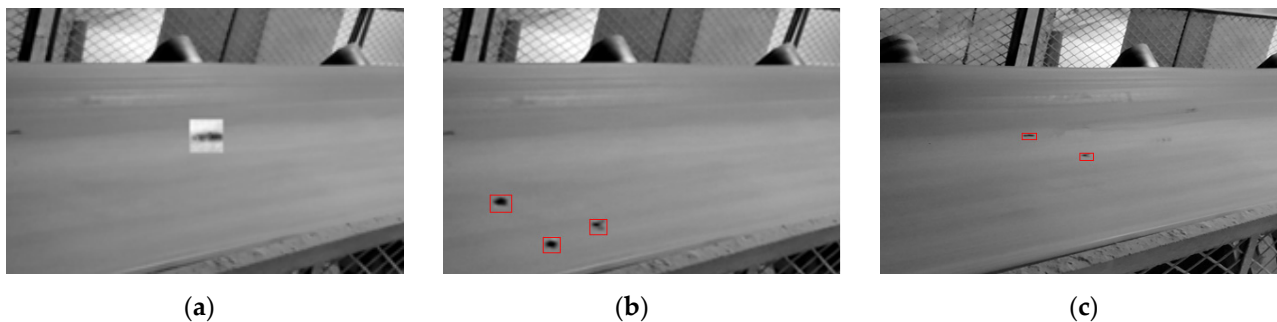
In order to generate higher-quality images of the conveyor belt scratches, and to avoid background information other than the conveyor belt scratches as noise to affect the generation effect, this paper adopts the method of generating only scratch samples to augment the data. Figure 3 shows the effect of scratch samples generated by the DCGAN. The samples generated by the DCGAN had different morphologies from the original samples, which increased the richness and diversity of scratch samples, enabling the model to learn more features of different morphologies and enhance the generalization performance of the model.



**Figure 3.** Comparison between DCGAN-generated scratch samples and original samples. (a) Real scratch samples. (b) Scratch samples generated by DCGAN.

As shown in Figure 4a, due to the difference in gray values between the generated scratch samples and the background image of the conveyor belt, the boundary between the foreground and the background was obvious when the samples were directly pasted into the background image, which had a negative impact on the learning of the network. To solve this problem, Poisson fusion was used to smooth the junction part between the foreground and background. The Poisson fusion approach was proposed by Pérez [25] et al., which is a method to solve the phenomenon of unnatural boundary transition that occurs when the scatter of the ROI and the background is reconstructed based on the Poisson equation. This phenomenon occurs because of the abrupt change in the gradient at the junction between the foreground and the background; therefore, the solution is to make the change at the boundary decrease, and the solution of this minimum change is the solution to Poisson's equation.

$$\min_f \iint_{\Omega} |\nabla f - \mathbf{v}|^2 \text{ with } f|_{\partial\Omega} = f^*|_{\partial\Omega} \quad (5)$$



**Figure 4.** Data enhancement method of joint GAN and Copy-Pasting strategy. (a) Paste directly. (b) Poisson fusion. (c) Poisson fusion + random scale zooming.

Where  $f$  denotes the resultant image after fusion,  $\nabla f$  is its gradient and  $\mathbf{v}$  is the gradient of the original image.  $f^*$  is the target image,  $\Omega$  is the original image and  $\partial\Omega$  is the image boundary.

The result of the image after using Poisson fusion is shown in Figure 4b, where there was no longer a clear boundary between the scratch sample and the background of the conveyor belt. Finally, the scratch sample was scaled at a random scale to simulate the real scratch situation, and the effect is shown in Figure 4c. After this data enhancement method, there was a significant increase in both the number and sample shape of the data.

### 2.3.2. Design of Lightweight Network Based on YOLOv5

In the object detection model, YOLOv5, according to the number of layers and channels of the network, the model can be divided into five different network structures of different sizes from YOLOv5x, YOLOv5l, YOLOv5m, YOLOv5s and YOLOv5n, in descending order. In this paper, the smallest model YOLOv5n was chosen as the benchmark and pruned on

this basis. The network structure of YOLOv5n is shown in Figure 5: its network structure mainly consists of the backbone, neck and head, where the backbone is responsible for extracting the information in the image, the neck is responsible for fusing and extracting the previously obtained features, and the head part detects the feature maps of different scales to generate anchor frames.

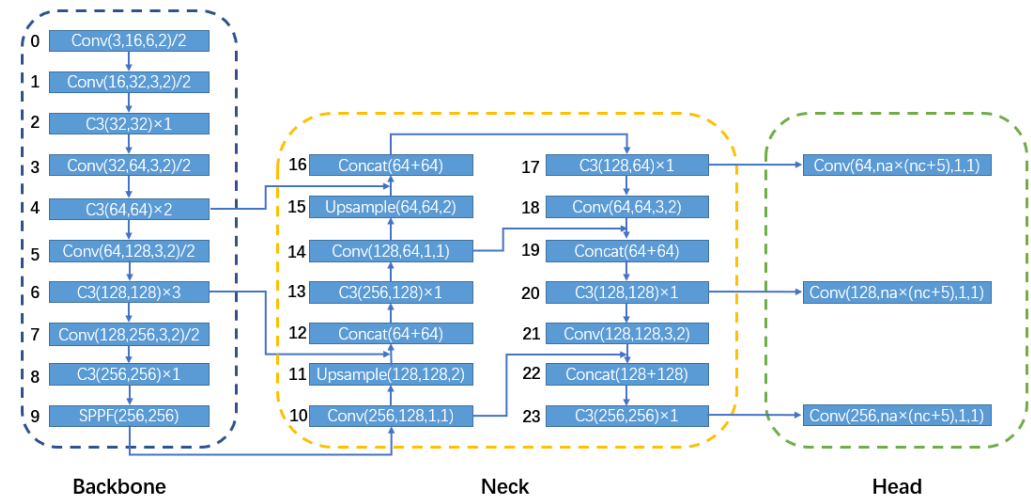


Figure 5. YOLOv5n network structure.

In this paper, we pruned the network's BN layer, which mainly existed in the Conv module, consisting of a convolution operation, a BN layer and the SiLU activation function. From Formulas (6) and (7) of the BN layer, it can be seen that when the value of the coefficient  $\gamma$  is small, the output value  $Z_{out}$  of the BN layer will also be small, so these channels with small activation values can be pruned out.

$$\hat{Z} = \frac{Z_{in} - \mu_{\beta}}{\sqrt{\sigma_{\beta}^2 + \epsilon}} \quad (6)$$

$$Z_{out} = \gamma \hat{Z} + \beta \quad (7)$$

Sparing the parameter  $\gamma$  by adding an L1 regularization constraint on  $\gamma$  to the loss function:

$$L = \sum_{(x,y)} l(f(x,W),y) + \lambda \sum_{\gamma \in \tau} g(\gamma) \quad (8)$$

where  $l(\cdot)$  is the original loss function,  $\lambda$  is the regularization factor and  $g(x) = |x|$ .

Figure 6 shows that the pruning of the model is achieved by removing the parameters with small activation values after sparsification training.

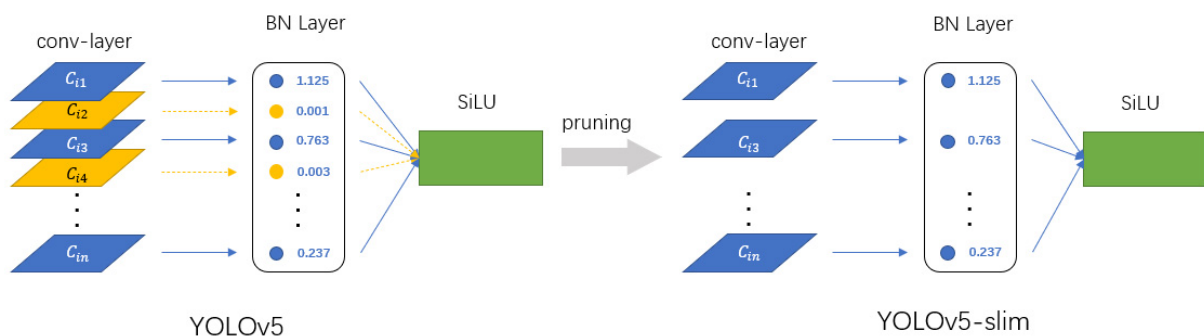
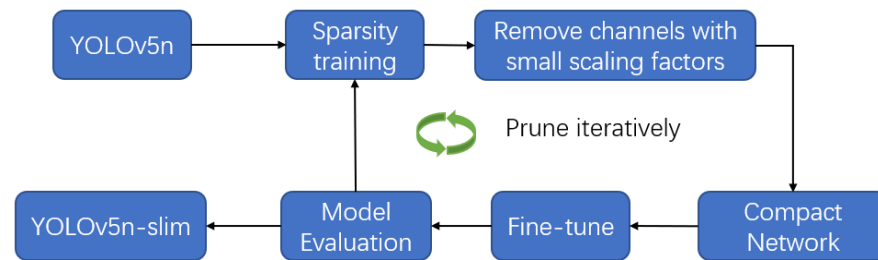


Figure 6. Pruning of YOLOv5 models.



The process of model pruning is shown in Figure 7. In order to remove the small-scale parameters, we sparsified the BN layer parameters  $\gamma$ . The model first needed to be trained sparsely, then the small-scale factors 2343 removed from it to generate a compact network, then the model was fine-tuned to recover its accuracy, the performance of the fine-tuned model was evaluated, and the above pruning operation was performed iteratively, according to the specific requirements. Finally, the mini-network YOLOv5n-slim was generated.



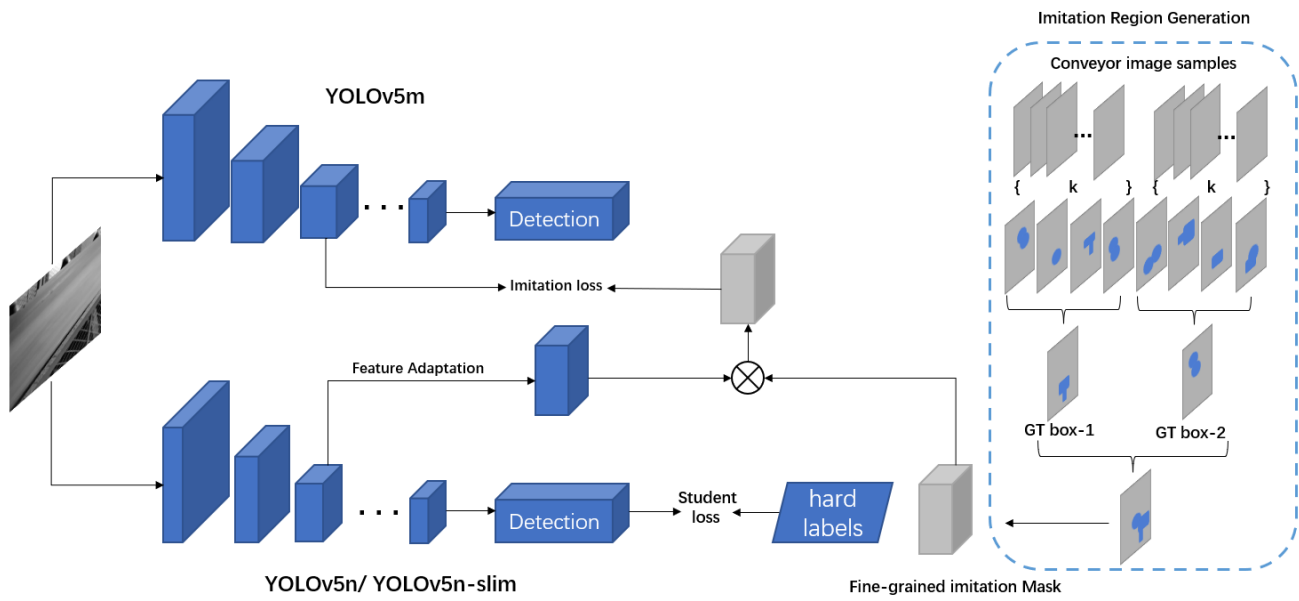
**Figure 7.** Model-pruning algorithm process.

### 2.3.3. Design of Knowledge Distillation Network Based on YOLOv5

In the field of object detection, the traditional method of knowledge distillation is to extract the feature map in the middle layer of the teacher network and the student network and calculate the distillation loss for the whole feature map. However, the network tends to learn the features of the target to be detected during training, while in the task of conveyor-belt-damage detection, the conveyor belt scratches and defects only account for a small part of the whole picture. At this time, a large amount of background information is regarded as a noisy part relative to the damage area, which negatively affects the training of the network. To address this problem, this paper chose to adopt a knowledge distillation strategy based on fine-grained feature simulation. The student network only simulates the teacher network for the area around the anchor frame instead of learning the features of the teacher network on the whole feature map, thus avoiding the impact of a large amount of useless background information on the network training.

The flow chart of the knowledge distillation method based on the fine-grained feature simulation is shown in Figure 8. In this paper, we used YOLOv5m as the teacher network, and we used YOLOv5n and its pruned model, YOLOv5n-slim, as the student model. To localize the distillation targets, for each GT box, the IOU between it and all anchor boxes was first calculated to form a  $W \times H \times K$  IOU mapping graph, where  $W$  and  $H$  are the width and height of the feature map, respectively, and  $K$  denotes  $K$  pre-defined anchor boxes. Then, we found the maximum value of IOU  $M = \max(M)$ , multiplied it by the threshold factor  $\psi$ , and obtained the filter  $F = \psi * M$ . Finally, we used  $F$  to filter all IOU mappings, keeping only the positions larger than  $F$  and using the OR operation to obtain the feature map of size  $W \times H$ . The final fine-grained simulation mask  $I$  is obtained by merging all the GT boxes. When  $\psi = 0$ , the generated mask included all the positions on the feature map, and when  $\psi = 1$ , no position was retained. Different masks can be obtained by changing the value of  $\psi$ .

During the training process, to ensure the distillation effect, it is necessary to keep the student network and teacher network feature map sizes consistent, to which a feature adaptation layer is added after the feature map of the student network to solve the problem of mismatch between the student network and the teacher network feature map channels. Finally, the deviations of these anchor frames in the student network and the teacher network were obtained by combining the mask information as the simulated losses and were added to the loss function of the distillation training.



**Figure 8.** Structure of knowledge distillation network based on YOLOv5.

We define  $s$  as the bootstrap feature map of the student network and  $t$  as the feature map of the corresponding teacher network. For each approaching target anchor location  $(i, j)$  on the feature map with width  $W$  and height  $H$ , the loss function between the student network and the teacher network is as follows:

$$l = \sum_{c=1}^C (f_{\text{adap}}(s)_{ijc} - t_{ijc})^2 \quad (9)$$

combined with the mask information  $I$  to obtain the simulated loss function:

$$L_{\text{imitation}} = \frac{1}{2N_p} \sum_{i=1}^W \sum_{j=1}^H \sum_{c=1}^C I_{ij} (f_{\text{adap}}(s)_{ijc} - t_{ijc})^2 \quad (10)$$

$$N_p = \sum_{i=1}^W \sum_{j=1}^H I_{ij} \quad (11)$$

where  $N_p$  is the number of positive points in the mask and  $f_{\text{adap}}(\cdot)$  is the adaptation function that unifies the size of the feature map. The loss function of the final student model is:

$$L = L_{gt} + \lambda L_{\text{imitation}} \quad (12)$$

where  $L_{gt}$  is the detection loss of the student network on real data and  $\lambda$  is the weighting factor of the simulated loss. In the process of training the network, we used larger values of  $\lambda$  to make the student network learned the teacher network features as soon as possible, and then gradually decreased the value of  $\lambda$  to make the student network learn the target features based on real data and enhance the detection effect.

We use the medium-sized network YOLOv5m as the teacher network and YOLOv5n with the pruned network, YOLOv5-slim, as the student network for knowledge distillation. The comparison of backbone network parameters of the teacher network and the student network is shown in Figure 9. Because channel pruning does not change the model finishing structure, the pruned YOLOv5n had the same structure as it did before pruning.



Layer	Teacher:YOLOv5m	Student:YOLOv5n/YOLOv5n-slim
Conv	3,48,6×6, stride 2	3,16,6×6, stride 2
Conv	48,96,3×3, stride 2	16,32,3×3, stride 2
C3	(96,96, stride 2)×2	(32,32, stride 2)×1
Conv	96,192,3×3, stride 2	32,64,3×3, stride 2
C3	(192,192, stride 4)×4	(64,64, stride 2)×2
Conv	192,384,3×3, stride 2	64,128,3×3, stride 2
C3	(384,384, stride 6)×6	(128,128, stride 3)×3
Conv	384,768,3×3, stride 2	128,256,3×3, stride 2
C3	(768,768, stride 2)×2	(256,256, stride 2)×2
SPPF	768,768	256,256

**Figure 9.** Comparison of teacher model and student model.

### 3. Result and Discussion

#### 3.1. Experiment Environment and Parameter Settings

All experiments in this paper were implemented on the open-source framework Py-Torch 1.11.0, using a computer with Intel Xeon E5-2690 v4 CPU configuration, a Tesla P100-PCIE-16GB graphics card model and an Ubuntu 20.04 operating system. The initialized learning rate  $lr = 0.01$ , number of iterations epoch = 100, and other specific parameters are shown in Table 2.

**Table 2.** Relevant training parameters.

Parameters	Value
Batch_size	60
Img_size	$640 \times 640$
Momentum	0.937
IOU_thresh	0.20

#### 3.2. Dataset and Evaluation Indicators

Since the conveyor belt defect samples were different from the scratch samples in terms of morphology and location, and were more difficult to obtain than the scratches, they could not be enhanced using GANs. Therefore, in order to solve the problem of insufficient conveyor belt defect samples and the imbalance between the number of samples and scratch samples, this paper adopted an oversampling method to enhance the conveyor belt defect samples. The samples were oversampled by repeatedly adding an image to the training several times during the network training process. We divided the original defective samples into two parts: training set and test set, and then performed conventional data enhancements, such as rotation and cropping. Finally, the data-enhanced training set samples were three-times oversampled to generate the final conveyor belt's defective samples dataset.

In this paper, we used a total of 1533 conveyor belt images taken at the production site and manually annotated them; then, the dataset was enhanced by rotating, cropping and adding noise to generate 3066 images. A total of 1500 images were generated by the joint GAN and copy-pasting strategy; 1177 samples were generated by oversampling the

defective conveyor belt samples, then, we obtained a dataset containing 7276 conveyor belt samples. The training set and test set were divided with a ratio of 8:2 and, finally, 5821 samples were obtained from the training set and 1455 samples were obtained from the test set.

To have a more scientific evaluation standard for the model's performance, the evaluation indexes used in this paper included classification accuracy (Precision, P), recall rate (Recall, R), mean average accuracy (mean Average Precision, mAP@0.5), number of model parameters (Params) and inference time (Inference/ms), which are formulated as follows:

$$Precision = \frac{TP}{TP + FP} \quad (13)$$

$$Recall = \frac{TP}{TP + FN} \quad (14)$$

$$AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) P_{inter}(r_i + 1) \quad (15)$$

$$mAP = \frac{\sum_{i=1}^k AP_i}{k} \quad (16)$$

The TP (true positive) indicates the number of correctly predicted positive samples, FP (false positive) indicates the number of incorrectly predicted positive samples, and FN (false negative) indicates the number of incorrectly predicted negative samples.

### 3.3. Data Augmentation Strategy Ablation Experiments

To prove the effectiveness of our data augmentation strategy, we used different datasets to train YOLOv5n, and compared the detection results of different datasets when using the same detection model.

From Table 3, we can see that, when we train the model using the original data, it is difficult to train the detection model effectively due to the small number of samples; thus, the detection accuracy of the model is relatively poor. Then, we used traditional data augmentation methods, such as rotation, translation and cropping, to augment the data: the detection accuracy was improved to a certain extent. After that, we used our data augmentation method combining GAN and copy-pasting strategies to augment the dataset, and the detection accuracy was further improved. However, since the number of conveyor belt defect samples was still too small, in order to improve the detection accuracy of the defect parts, we oversampled the conveyor belt defect samples in the training set to obtain the final dataset. Finally, we obtained 7276 conveyor belt images, which contained about 12,000 scratch samples and about 1200 conveyor-belt-edge defect samples.

**Table 3.** Detection results with different datasets.

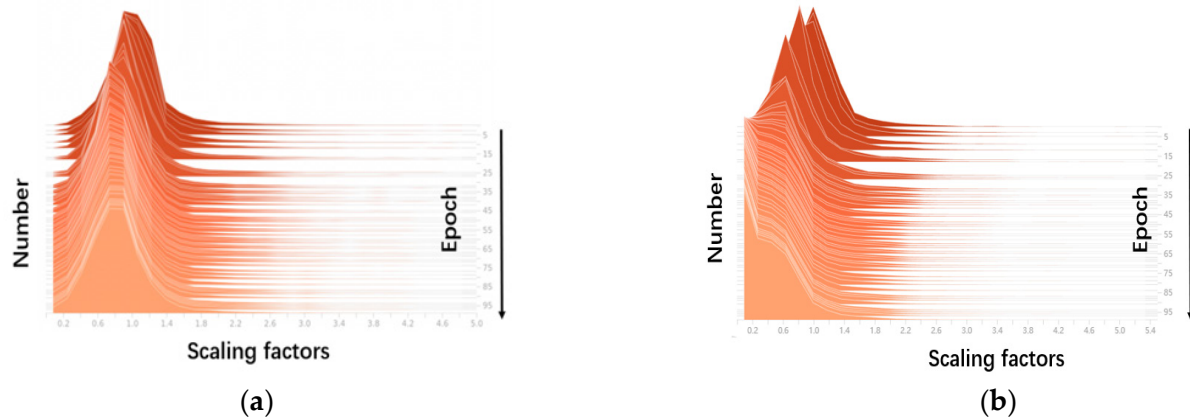
Dataset	Number of Images	Defect mAP@0.5/%	Scratch mAP@0.5/%	All mAP/%
Ori	1533	23.80	83.38	53.59
Ori+aug	4599	26.42	89.90	58.16
Ori+aug+GAN	6099	26.40	94.32	60.36
Ori+aug+GAN+oversampling	7276	96.21	94.93	95.57

After data augmentation, the number and diversity of conveyor-belt-damage samples in the dataset were significantly improved, increasing the number of samples while enriching the diversity of samples and improving the generalization ability of the detection model. The mAP@0.5 can reach 95.57% accuracy in detecting conveyor belt damage, which meets the requirements for detection accuracy in industrial applications.

### 3.4. Results of Model-Pruning Experiments

To prune the model, it is first necessary to train the model sparsely, as shown in Figure 10. We visualized the scaling factors of the BN layer. It can be seen from the figure

that the BN layer scaling factor is normally distributed before the model is sparse trained, and after the sparse training, the distribution of the BN layer scaling factor gradually converges to near zero due to the L1 regularization of the parameters. Then, the channels close to zero are removed to achieve the pruning of the model.



**Figure 10.** Distribution of BN layer scaling factors (a) Original training; (b) Sparse training.

In this paper, we pruned and fine-tuned the lightweight network YOLOv5n, and compared the performance of various aspects of the model with different pruning ratios. The results are shown in Table 4.

**Table 4.** Results of model pruning.

Model	Pruning Rate/%	P/%	R/%	mAP@0.5/%	Params/M	Inference Time/ms
YOLOv5n	0	86.02	93.84	95.57	1.76	1.7
YOLOv5n-slim-0.2	20	87.70	88.50	92.92	1.49	1.5
YOLOv5n-slim-0.3	30	89.28	84.91	92.75	1.22	1.4
YOLOv5n-slim-0.4	40	89.14	88.90	92.64	1.01	1.3
YOLOv5n-slim-0.5	50	85.43	86.65	89.58	0.86	1.3
YOLOv5n-slim-0.6	60	83.17	87.40	89.10	0.68	1.2
YOLOv5n-slim-0.7	70	82.22	88.70	90.74	0.49	1.2
YOLOv5n-slim-0.8	80	81.72	85.51	86.68	0.32	1.1
YOLOv5n-slim-0.9	90	83.70	78.21	83.50	0.17	1.0

From Table 2, it can be seen that, with the increase in pruning rate, the number of model parameters and inference time decreased, but at the same time, the detection accuracy of the model decreased, even after fine-tuning. When the pruning ratio exceeded 70%, the model performance decreased more obviously. Therefore, considering both the accuracy and speed of the model, this paper used a network with a pruning ratio of 70% as the final mini-network and named it YOLOv5n-slim, based on which a knowledge distillation algorithm was used to distill it to improve the performance of the model.

### 3.5. Experimental Results of Knowledge Distillation Algorithm

To verify the effectiveness of the knowledge distillation strategy used in this paper, YOLOv5m was used as the teacher network, and YOLOv5n and YOLOv5n-slim were used as the student network with mini-networks. In this paper, the network after YOLOv5n distillation was named YOLOv5n(KD), and the network after YOLOv5n-slim distillation was named YOLOv5n-slim(KD). The test results of the model are shown in Table 5.

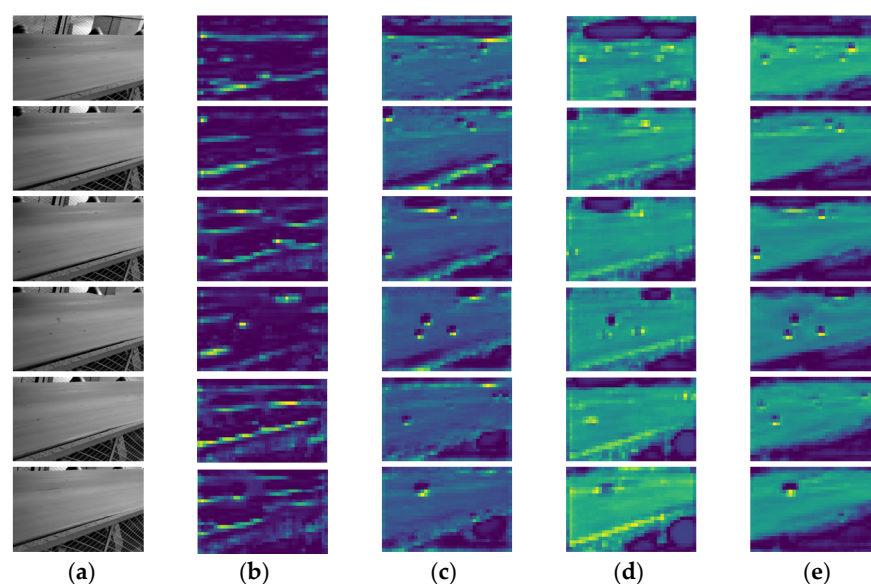
**Table 5.** Comparison of detection performance of each model.

Model	P/%	R/%	mAP@0.5/%	Params/M	Inference Time/ms
YOLOv5n	86.02	93.84	95.57	1.76	1.7
YOLOv5s	91.88	95.84	97.53	7.23	4.7
YOLOv5m	93.89	96.98	98.44	20.86	10.1
YOLOv5n-slim	82.22	88.70	90.74	0.49	1.2
YOLOv5n(KD)(ours)	90.75	93.47	97.33	1.76	1.7
YOLOv5n-slim(KD)(ours)	84.60	88.60	94.83	0.49	1.2

To compare the performance of the models after distillation, three models of different sizes of YOLOv5 were tested in this paper, as can be seen from Table 3: (1) In the YOLOv5 series of models, the corresponding detection accuracy and inference time increased as the model increased from YOLOv5n to YOLOv5m. (2) The pruned network YOLOv5n-slim decreased by 4.83% in mAP@0.5, 72.16% in the number of parameters, and 29.41% in the inference time compared with that before pruning. (3) After the introduction of knowledge distillation, the detection accuracy of the model was relatively significantly improved compared to the previous one, where YOLOv5n(KD) improved by 1.76% compared to mAP@0.5 before distillation, and mAP@0.5 decreased by 1.11%, parameter volume decreased by 91.56% and inference time decreased by 83.17%, compared to the teacher network YOLOv5m. Compared with the network YOLOv5s, which is larger than it, a 75.66% reduction in parameter volume and a 63.83% reduction in inference time were achieved at the cost of a 0.2% reduction in mAP@0.5. (4) The pruned student network YOLOv5n-slim showed a 4.09% improvement in mAP@0.5 after distillation, compared with the teacher network YOLOv5m mAP@0.5 which decreased by 2.92%. Additionally, the amount of network parameters decreased by 97.65%, and the inference time decreased by 88.12%. This shows that the knowledge distillation algorithm used in this paper can effectively improve the accuracy of the detection model without increasing the complexity of the model.

### 3.6. Feature Map Comparison Analysis

We choose to visualize the feature maps output by the network detection head, and the results are shown in Figure 11.



**Figure 11.** Comparison of output feature maps of different network detection heads. (a) Initial input images. (b) Feature map of YOLOv5m. (c) Feature map of YOLOv5n. (d) Feature map of YOLOv5n(KD). (e) Feature map of YOLOv5n-slim(KD).

From the comparison of the feature maps, it can be seen that the distilled network has higher activation values in the target region with clearer boundaries and is less influenced by background information. Therefore, the fine-grained feature-simulation distillation method used in this paper is a good guide for improving the feature-learning performance of the network.

### 3.7. Comparison with Other Models

To show the effectiveness of the knowledge distillation algorithm, we compared it with other YOLO lightweight networks, and the experimental results are shown in Table 6.

**Table 6.** Comparison of YOLO Lightweight Network Performance.

Model	P/%	R/%	mAP@0.5/%	Params/M	Inference Time/ms
YOLOv3-tiny	92.46	93.47	96.74	8.67	2.1
YOLOv4-tiny	67.39	86.68	84.30	6.06	4.9
YOLOv7-tiny	72.46	90.92	85.49	6.02	5.0
YOLOv5n(KD)(ours)	90.75	93.47	97.33	1.76	1.7
YOLOv5n-slim(KD)(ours)	84.60	88.60	94.83	0.49	1.2

From the table, it can be seen that, compared to the lightweight network YOLOv3-tiny, YOLOv5n(KD), after using the distillation algorithm in this paper, was improved by 0.59% in mAP@0.5, reduced 79.7% in parameters, and reduced by 19.05% in inference time. The pruned model YOLOv5n-slim(KD) was 1.91% lower in mAP and had 94.35% less parameters and 42.86% less inference time.

Compared with YOLOv4-tiny, the model after using knowledge distillation in this paper had a 13.03% and 10.53% improvement in mAP@0.5, a 70.46% and 91.91% reduction in the number of parameters, and a 65.31% and 75.51% reduction in inference time, respectively.

Compared with the lightweight model YOLOv7-tiny in the latest YOLO version YOLOv7, the distilled model mAP@0.5 was improved by 11.84% and 9.34%, the number of parameters was reduced by 70.76% and 91.86%, and the inference time was reduced by 66% and 76%, respectively. The knowledge distillation algorithm used in this paper can assist the student network to better learn the features of the target to be detected and effectively improve the detection accuracy of the model by distilling the features of the target region without increasing the complexity of the model.

As shown in Table 7, we also compared our research methods with those of others in the same area. Although the experimental environment and equipment were different, our method achieved a detection accuracy similar to other algorithms on devices with small differences in performance, and it achieved a significant improvement in detection speed.

**Table 7.** Comparison with other model detection results.

Methods	mAP@0.5/%	Inference Time/ms
EfficientNet-YOLOv3	97.26	23.8
Yolox-BTFPN	98.45	25.6
MobilenetV3-YOLOv4	93.08	22.7
MCC-CycleGAN	96.88	22.6
YOLOv5n(KD)(ours)	97.33	1.7
YOLOv5n-slim(KD)(ours)	94.83	1.2

In conclusion, we compared all the algorithms mentioned in this paper, and the results are shown in Figure 12:

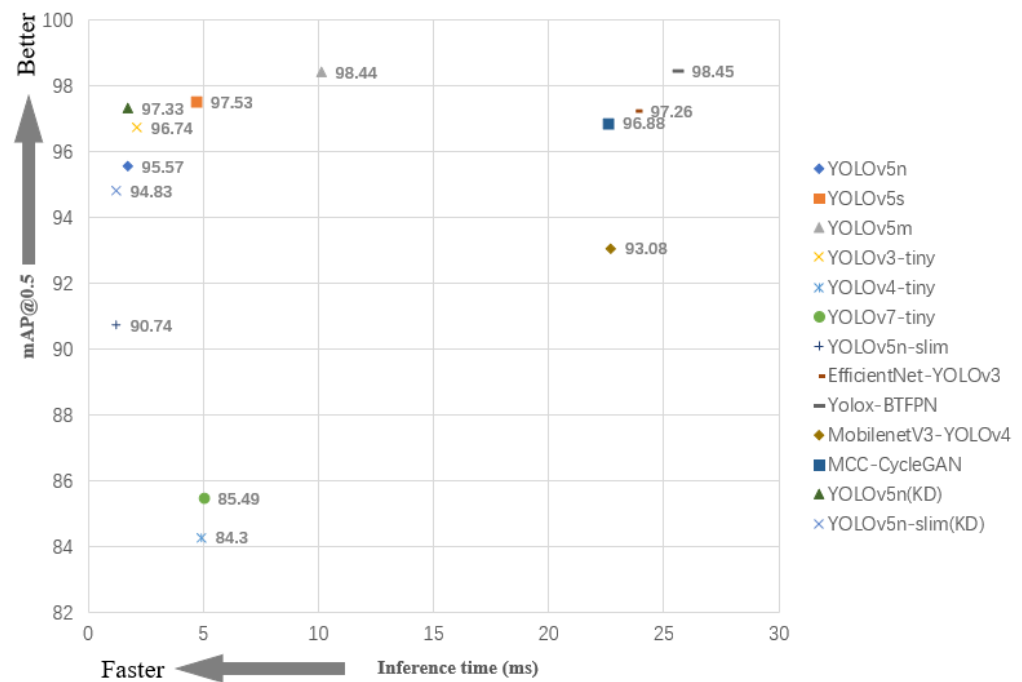


Figure 12. Performance comparison of the models.

#### 4. Conclusions and Future Work

We proposed a new detection method based on knowledge distillation in this paper to address the problem that defective samples are difficult to obtain in the damage detections of the mineral-transportation conveyor belt. Firstly, the DCGAN was used to generate conveyor belt scratch samples, and a data enhancement method combining GAN and copy-pasting strategies was proposed to effectively solve the problem of insufficient samples to train the neural network model. In addition, to address the requirements of model accuracy and speed in production environments, this paper pruned the model of YOLOv5n, investigated the model performance under different pruning ratios, and finally generated a smaller miniature network, before adopting a knowledge distillation method based on fine-grained feature simulation for the lightweight network YOLOv5n and the miniature network YOLOv5n-slim for knowledge distillation. For the task of detecting conveyor belt damage, the mAP@0.5 can reach 97.33% after knowledge distillation for the lightweight network YOLOv5n and 94.83% for the micro network YOLOv5n-slim, both of which meet the requirements of detection accuracy and speed in industrial applications.

Compared with other research, our study innovatively introduced model pruning and knowledge distillation to the task of conveyor-belt-damage detection. The model-pruning and knowledge distillation method designed in this paper significantly reduces the size of the model and improves the detection accuracy of the model without increasing the complexity of the model, enabling it to approach the detection accuracy of medium and large models with a smaller size and a faster inference speed. This allows our detection model to be deployed on lower-performance devices.

In following research, we will continue to work on the lightweighting of the model and the improvement of the detection accuracy. Additionally, we will study the influence of environmental factors, such as light and dust, on the detection results.

**Author Contributions:** Conceptualization, methodology, software and writing—original draft preparation Q.Y.; validation and formal analysis, F.L.; investigation, H.L. and Z.W.; resources, S.X.; writing—review and editing, H.T.; visualization, Q.F.; supervision, C.L.; data curation and funding acquisition, J.F. All authors have read and agreed to the published version of the manuscript.



**Funding:** The authors sincerely thank for the support by the Key Laboratory of Modern Measurement and Control Technology, Beijing University of Information Science and Technology, Ministry of Education (OMTIKF2021001).

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** The data in this study are available on request from the corresponding.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. He, D.; Pang, Y.; Lodewijks, G. Green operations of belt conveyors by means of speed control. *Appl. Energy* **2017**, *188*, 330–341. [\[CrossRef\]](#)
2. Wang, J.; Miao, C.; Cui, Y.; Wang, W.; Zhou, L. Research of X-ray Nondestructive Detection System for High-speed running Conveyor Belt with Steel Wire Ropes. *Mod. Appl. Sci.* **2007**, *1*, 47. [\[CrossRef\]](#)
3. Yang, Y.; Hou, C.; Qiao, T.; Zhang, H.; Ma, L. Longitudinal tear early-warning method for conveyor belt based on infrared vision. *Measurement* **2019**, *147*, 106817. [\[CrossRef\]](#)
4. Yang, R.; Qiao, T.; Pang, Y.; Yang, Y.; Zhang, H.; Yan, G. Infrared spectrum analysis method for detection and early warning of longitudinal tear of mine conveyor belt. *Measurement* **2020**, *165*, 107856. [\[CrossRef\]](#)
5. Qiao, T.; Liu, W.; Pang, Y.; Yan, G. Research on visible light and infrared vision real-time detection system for conveyor belt longitudinal tear. *IET Sci. Meas. Technol.* **2016**, *10*, 577–584. [\[CrossRef\]](#)
6. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask r-cnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2961–2969.
7. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2015; Volume 28.
8. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single shot multiBox detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 21–37.
9. Lin, T.Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal loss for dense object detection. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
10. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
11. Zhang, M.; Shi, H.; Zhang, Y.; Yu, Y.; Zhou, M. Deep learning-based damage detection of mining conveyor belt. *Measurement* **2021**, *175*, 109130. [\[CrossRef\]](#)
12. Wang, G.; Liu, Z.; Sun, H.; Zhu, C.; Yang, Z. Yolox-BTFPN: An anchor-free conveyor belt damage detector with a biased feature extraction network. *Measurement* **2022**, *200*, 111675. [\[CrossRef\]](#)
13. Zhang, M.; Zhang, Y.; Zhou, M.; Jiang, K.; Shi, H.; Yu, Y.; Hao, N. Application of Lightweight Convolutional Neural Network for Damage Detection of Conveyor Belt. *Appl. Sci.* **2021**, *11*, 7282. [\[CrossRef\]](#)
14. Guo, X.; Liu, X.; Królczyk, G.; Sulowicz, M.; Glowacz, A.; Gardoni, P.; Li, Z. Damage Detection for Conveyor Belt Surface Based on Conditional Cycle Generative Adversarial Network. *Sensors* **2022**, *22*, 3485. [\[CrossRef\]](#) [\[PubMed\]](#)
15. Liu, Z.; Li, J.; Shen, Z.; Huang, G.; Yan, S.; Zhang, C. Learning efficient convolutional networks through network slimming. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2736–2744.
16. Wang, T.; Yuan, L.; Zhang, X.; Feng, J. Distilling object detectors with fine-grained feature imitation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 4933–4942.
17. Goodfellow, I.J.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial nets. In *Advances in Neural Information Processing Systems*; Curran Associates, Inc.: Red Hook, NY, USA, 2014; pp. 2672–2680.
18. Hinton, G.; Vinyals, O.; Dean, J. Distilling the knowledge in a neural network. *arXiv* **2015**, arXiv:1503.02531.
19. Hou, Y.; Ma, Z.; Liu, C.; Loy, C.C. Learning Lightweight Lane Detection CNNs by Self Attention Distillation. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Korea, 27 October–2 November 2019; pp. 1013–1021. [\[CrossRef\]](#)
20. Gao, M.; Shen, Y.; Li, Q.; Loy, C.C. Residual knowledge distillation. *arXiv* **2020**, arXiv:2002.09168.
21. Rahnamoun, R.; Rawassizadeh, R.; Maskooki, A. Learning mobile app usage routine through learning automata. *arXiv* **2016**, arXiv:1608.03507.
22. Chen, P.; Liu, S.; Zhao, H.; Jia, J. Distilling knowledge via knowledge review. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 5008–5017.
23. Ghiasi, G.; Cui, Y.; Srinivas, A.; Qian, R.; Lin, T.Y.; Cubuk, E.D.; Le, Q.V.; Zoph, B. Simple copy-paste is a strong data augmentation method for instance segmentation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Virtual, 19–25 June 2021; pp. 2918–2928.
24. Radford, A.; Metz, L.; Chintala, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv* **2015**, arXiv:1511.06434.
25. Pérez, P.; Gangnet, M.; Blake, A. Poisson image editing. In *ACM SIGGRAPH 2003 Papers*; Association for Computing Machinery: New York, NY, USA, 2003; pp. 313–318.