

Research Article

Research on Deviation Detection of Belt Conveyor Based on Inspection Robot and Deep Learning

Yi Liu ,^{1,2} **Changyun Miao** ,³ **Xianguo Li** ,³ and **Guowei Xu** ,²

¹*School of Mechanical Engineering, Tiangong University, Tianjin 300387, China*

²*Center for Engineering Internship and Training, Tiangong University, Tianjin 300387, China*

³*School of Electronics and Information Engineering, Tiangong University, Tianjin 300387, China*

Correspondence should be addressed to Changyun Miao; miaochangyun@tiangong.edu.cn

Received 9 June 2020; Revised 30 November 2020; Accepted 6 February 2021; Published 25 February 2021

Academic Editor: Hang Su

Copyright © 2021 Yi Liu et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The deviation of the conveyor belt is a common failure that affects the safe operation of the belt conveyor. In this paper, a deviation detection method of the belt conveyor based on inspection robot and deep learning is proposed to detect the deviation at its any position. Firstly, the inspection robot captures the image and the region of interest (ROI) containing the conveyor belt edge and the exposed idler is extracted by the optimized MobileNet SSD (OM-SSD). Secondly, Hough line transform algorithm is used to detect the conveyor belt edge, and an elliptical arc detection algorithm based on template matching is proposed to detect the idler outer edge. Finally, a geometric correction algorithm based on homography transformation is proposed to correct the coordinates of the detected edge points, and the deviation degree (DD) of the conveyor belt is estimated based on the corrected coordinates. The experimental results show that the proposed method can detect the deviation of the conveyor belt continuously with an RMSE of 3.7 mm, an MAE of 4.4 mm, and an average time consumption of 135.5 ms. It improves the monitoring range, detection accuracy, reliability, robustness, and real-time performance of the deviation detection of the belt conveyor.

1. Introduction

Belt conveyor is continuous transportation equipment in modern production with the advantages of large capacity, being suitable for long distance, low freight, high efficiency, stable operation, convenient loading and unloading, being suitable for bulk material transportation, etc. It has become one of the three main industrial conveyances together with automobile and train and has been widely used in coal, mines, ports, electric power, metallurgy, chemical industry, and other fields [1]. The operating condition of the belt conveyor is harsh, and the deviation fault occurs due to improper installation and adjustment, manufacturing errors, bearing idler failure, uneven distribution of materials, adhesion of slime to the driving pulley and the bend pulley, etc. [2]. The deviation fault of the belt conveyor can lead to the tearing of the conveyor belt, the material spilling, and the belt conveyor damage. To ensure safe operation, it is required to detect the deviation of the conveyor belt.

At present, the main detection method of the deviation fault is to install two sets of deviation switches on the racks on both sides of the conveyor belt. Once deviation occurs, the conveyor belt pushes the action arm of one set of the deviation switch to deflect, and the deviation switch is triggered and sends an alarm signal. If the conveyor belt continues to deviate to the set stop position, it triggers another set of deviation switch and it will control the belt conveyor to stop. This method can only detect two fixed belt deviation positions and cannot estimate the deviation degree (DD). Meanwhile, it often gives false alarms to cause downtime and affect production. In addition, the deviation switch is prone to be damaged by the conveyor belt running at high speed, and its reliability and accuracy are poor. Since the deviation fault cannot be effectively detected, the longitudinal tearing, the belt breaking, and the belt conveyor damage occur frequently.

In order to detect the deviation of the conveyor belt effectively, in recent years, researchers have paid considerable attention to the machine vision-based methods, which

are contactless and more reliable. Yang et al. [3] proposed a deviation detection method based on the image segmentation algorithm. The linear array CCD was used to collect the images of the lower surface of the running conveyor belt, and the image segmentation algorithm based on the column threshold was used to detect the conveyor belt edge, according to which the deviation of the conveyor belt can be detected qualitatively. This method is simple and efficient, but the lens of the camera is easy to be polluted by materials and dusts, which affects the imaging quality. Mei et al. [4] proposed a deviation detection method based on the image enhancement algorithm, which used the adaptive threshold and the image enhancement algorithm to detect the conveyor belt edge, and calculated the offset and the distortion of the conveyor belt, according to which the DD was estimated and the deviation fault was predicted. This method is efficient and accurate, but still with the defect that the lens is susceptible to contamination. The neural network (NN) is with prominent nonlinear mapping and feature extraction capability. It has been applied to minimally invasive surgery, the control of robotic manipulators, mechanical fault detection, human activities monitoring, robot tool dynamics identification, medical images [5–11], etc. and has demonstrated superior performance. Liu et al. [12] proposed a deep learning-based deviation detection method of the belt conveyor, which has reliable object detection and anti-jamming ability, but the error of the estimated DD is large. The cameras of the abovementioned deviation detection methods are all installed in fixed positions and can only detect the fixed positions. However, the deviation fault may occur at any position of the belt conveyor. Therefore, it is more suitable to use an inspection robot to detect the deviation along the belt conveyor. An inspection robot is a great substitute for human beings to perform periodic inspection tasks in dangerous scenes and has been widely used in the power system [13] and construction [14]. In recent years, it has been implemented in coal mines [15], but the deviation detection along the belt conveyor is still unsolved.

To address this issue, a deviation detection method of the belt conveyor based on inspection robot and deep learning is proposed in this paper, which provides a more intelligent solution for the monitoring of the belt conveyor. The main idea is to combine the deep learning algorithm and the digital image processing technology to detect the deviation of the conveyor belt on an inspection robot. To summarize, our contributions are listed as follows:

- (1) The novel deviation detection method is proposed. The inspection robot is applied to detect the deviation at any position along the belt conveyor. In order to determine the relative position of the conveyor belt, the idler outer edge is used as a reference, and the deep learning algorithm is introduced to enhance the ability of the inspection robot to deal with the complex environment. In addition, the digital image processing algorithms are combined to detect edges and correct distortion.
- (2) The ROI detector based on MobileNet SSD (M-SSD) is designed to detect the conveyor belt edge and the

exposed idler in real time. The prediction source layers of M-SSD are customized to improve the detection accuracy and speed.

- (3) A conveyor belt edge detection method based on the Hough line transform is designed to detect the conveyor belt edge, and it is more robust than other methods. An elliptical arc detection algorithm based on template matching is proposed to detect the idler outer edge. The template elliptical arcs are generated by the linear transformation of a semicircle, and this method can detect the idler outer edge efficiently.
- (4) A novel geometric correction method is proposed. The geometric correction is simplified from three-dimensional (3D) projection transformation into a combination of plane homography transformation and plane geometric transformation. The DD is estimated by the proposed DD estimation method, and it is effective in different DDs.

The rest of this paper is structured as follows. Section 2 details the proposed deviation detection method and its components. Section 3 presents the experimental platform and the result analysis. Section 4 concludes our work and gives the further research directions.

2. Materials and Methods

2.1. Schematic of the Belt Conveyor Deviation Detection Method. The inspection robot runs on the rack track beside the belt conveyor, and it moves parallel to the central axis l_c of the belt conveyor. A mobile camera is fixed on the metal frame of the inspection robot, and its optical axis is perpendicular to l_c . The mobile camera captures the image containing the conveyor belt and the exposed idlers, and the lateral axis of the image is parallel to l_c . The proposed method combines the deep learning algorithm and the image processing technology, as shown in Figure 1. It can be summarized in two phases. In phase 1, the ROI containing the conveyor belt edge and the exposed idler is extracted. In phase 2, the conveyor belt edge and the idler outer edge are detected, and the DD is estimated based on the corrected coordinates of them. This method combines the antijamming ability of deep learning and the precision of digital image processing and detects the deviation in complex and changeable background environments.

Phase 1 is composed of three steps: (1) the algorithm captures the image continuously and crops the image to two subimages with the same size, named imgO and imgI, respectively. (2) We input the two subimages to the well-trained ROI detector OM-SSD to extract the ROI containing the conveyor belt edge and the exposed idler, named ROI_O and ROI_I, respectively, and they may contain the outer idler or the inner idler (the idler on the outer side of the conveyor belt appears in the upper half of the image, and the inner side one appears in the lower half, which are called the outer idler and the inner idler, respectively). (3) If the ROIs are extracted successfully, we step into phase 2; otherwise, the procedure will be aborted and we go with the next image. Phase 2 consists of the following four steps: (1) the

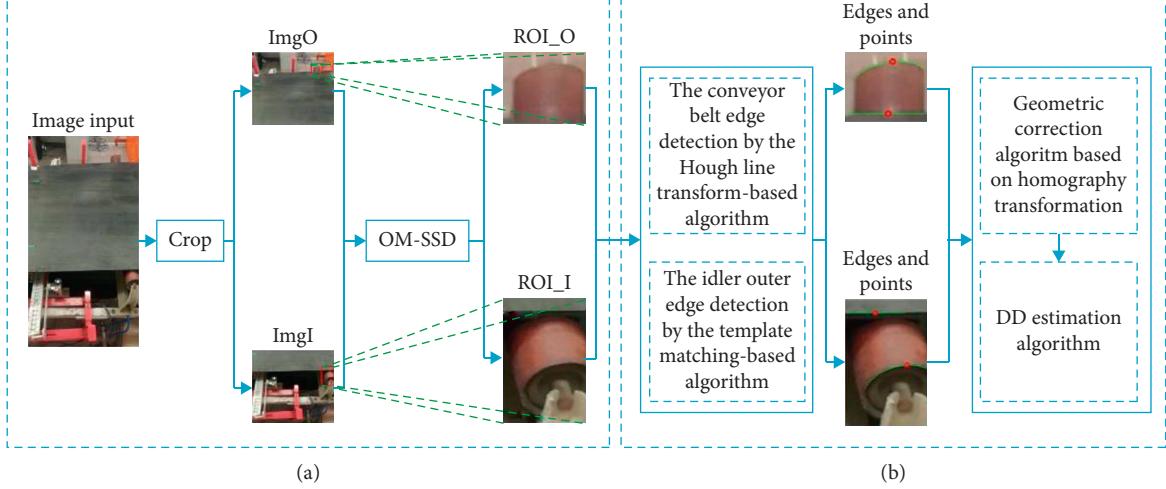


FIGURE 1: Schematic of the deep-learning-based belt conveyor deviation detection method. The inspection robot captures an image and divides it into imgO and imgI; then, they are input into the ROI detector (OM-SSD) to extract the ROI_O and ROI_I, respectively. The ROIs are converted into gray images, and then, the conveyor belt edge detection algorithm and the idler outer edge detection algorithm are implemented on them to get the edges. The points are the midpoints of the detected lines representing the conveyor belt edges and the vertices of the idler outer edges, respectively. The coordinates of the points are corrected by the geometric correction algorithm and then used to estimate the DD. The distances between the two points in ROI_O and ROI_I represent the lengths of the exposed outer and inner idlers, and the difference between the lengths indicates the DD. (a) Phase 1: ROI extraction. (b) Phase 2: DD estimation.

conveyor belt edge is detected by using the Hough line transform-based algorithm. (2) The idler outer edge is detected by the elliptical arc detection algorithm based on template matching. (3) The coordinates of the conveyor belt edge point and the idler outer edge point are corrected by implementing the geometric correction algorithm based on homography transformation; then, the length of the exposed idlers can be calculated. (4) The DD is estimated based on the length of the exposed idlers.

2.2. ROI Extraction Algorithm Based on M-SSD

2.2.1. The Framework of the ROI Extraction Algorithm. ROI extraction can be regarded as object detection. According to the detection stage, the methods of deep-learning-based object detection can be divided into 2 categories: two-stage and one-stage detection methods. The two-stage detection method is represented by faster RCNN [16], which is comprised of two stages. In stage 1, the proposal bounding boxes and the objectness scores are predicted simultaneously, and in stage 2, the region proposals are used for detection and regression. Faster RCNN can be considered as a combination of fast RCNN [17] and region proposal network (RPN), and it performs well in the public datasets. However, fast RCNN and RPN have to be trained alternately to share the parameters of the convolution layers and the prediction layers, which makes it inconvenient for integration and implementation. The one-stage detection method is represented by SSD [18] and YOLO [19], and they do not have an explicit generation stage of the proposal bounding boxes. Compared with YOLO and faster RCNN, SSD adopts pyramid architecture, imitates RPN to generate default boxes at each position from multiple source layers, and predicts each classification score of each

default box, thus making full use of each feature maps and avoiding the alternative training of the classification network and RPN, which makes it more flexible and widely used in different applications [18, 20].

2.2.2. Object Detect Network Based on MobileNet. The superior nonlinear mapping capability of the convolution neural network (CNN) is mostly derived from the depth and the capacity of the network, and the model goes deeper and larger for better performance in accuracy, such as AlexNet [21], VGGNet [22], GoogLeNet [23], and ResNet [24]. However, the training and deployment of them require higher hardware costs (memory and GPU), which limit their application in embedded and mobile devices. Therefore, streamlining the network has become an active research area in recent years [25–27]. The key point of the network pruning is to balance the tradeoff between accuracy and costs, that is, to reduce the parameters and computations dramatically with a slight degradation in accuracy. MobileNet [27] is an efficient CNN specified for mobile vision applications. The core is a streamlined architecture named depthwise separable convolution (DSC). Given an input feature map of size $D_W \times D_H \times M$, a convolution kernel of size $D_K \times D_K \times N$, the standard convolution assuming stride one and padding, has the computational cost of

$$\text{MAdds}_s = D_W D_H M D_K D_K N, \quad (1)$$

where MAdds is the multiadds [27], D_W , D_H , and M are the width, height, and number of channels of the input feature map, respectively, the width and the height of the convolution kernel are D_K , and the number of channels is N . The DSC divides the standard convolution into two

parts: a depthwise convolution and a pointwise convolution. The former is applied as a single filter per each input channel and outputs a feature map of size $D_W \times D_H \times M$, and the latter creates a linear combination of the output by using a simple convolution of size $1 \times 1 \times N$. Then, the computational cost of the DSC with stride one and padding can be calculated as

$$\text{MAdds}_D = D_W D_H M D_K D_K + D_W D_H M N. \quad (2)$$

Then, the computational cost ratio (CCR) of the DSC and the standard convolution can be written as

$$\frac{\text{MAdds}_D}{\text{MAdds}_s} = \frac{1}{N} + \frac{1}{D_K D_K}. \quad (3)$$

The state-of-art networks usually use small kernels [22], i.e., $D_K = 3$. When the output channel N gets larger, the CCR can be close to 1/9. In addition, the number of parameters P_s in the standard convolution is

$$P_s = D_W D_H M D_K D_K N, \quad (4)$$

while in the DSC, the number of parameters P_D is

$$P_D = M D_K D_K + M N, \quad (5)$$

and the ratio of them is as follows:

$$\frac{P_D}{P_s} = \frac{1}{D_W D_H N} + \frac{1}{D_K D_K D_W D_H}. \quad (6)$$

It can be proved that the larger the feature map and the number of the output channels get, the more the computations and the parameters decrease, and the experimental results show that the degradation in performance of the DSC is almost negligible [27]. With the lightweight architecture, deep CNNs and object detection frameworks can be deployed in the embedded devices successfully. The feature extraction network of M-SSD is shown in Table 1, with 22 layers, and all the convolution layers are followed by batchnorm layers and rectified linear unit (ReLU) layers, including the standard convolution (Conv) layers and the depthwise and pointwise convolution layers in the DSC.

2.2.3. ROI Detector. To detect objects of different sizes, M-SSD uses L11, L13, L15, L17, L19, and L21 as the prediction source layers in predicting the object bounding box. For the k -th of the 6 layers, the ratio S_k of the default box to the input image is determined as

$$S_k = S_{\min} + \frac{S_{\max} - S_{\min}}{m - 1} (k - 1), \quad k \in [1, m], \quad (7)$$

where S_{\min} and S_{\max} are the minimum and maximum of the ratio, which are 0.2 and 0.9, respectively. $m = 6$ is the number of the prediction source layers. To detect objects with different shapes, M-SSD sets different aspect ratios r_k for each default box at each position of the k -th layer, which can be determined as follows:

TABLE 1: The backbone of M-SSD.

Layer	Input size	Operator	Stride	Output channel
L0	$300 \times 300 \times 3$	Conv	2	32
L1	$150 \times 150 \times 32$	DSC	1	64
L2	$150 \times 150 \times 64$	DSC	2	128
L3	$75 \times 75 \times 128$	DSC	1	128
L4	$75 \times 75 \times 128$	DSC	2	256
L5	$38 \times 38 \times 256$	DSC	1	256
L6	$38 \times 38 \times 256$	DSC	2	512
L7	$19 \times 19 \times 512$	DSC	1	512
L8	$19 \times 19 \times 512$	DSC	1	512
L9	$19 \times 19 \times 512$	DSC	1	512
L10	$19 \times 19 \times 512$	DSC	1	512
L11	$19 \times 19 \times 512$	DSC	1	512
L12	$19 \times 19 \times 512$	DSC	2	1024
L13	$10 \times 10 \times 1024$	DSC	1	1024
L14	$10 \times 10 \times 1024$	Conv	1	256
L15	$10 \times 10 \times 256$	Conv	2	512
L16	$5 \times 5 \times 512$	Conv	1	128
L17	$5 \times 5 \times 128$	Conv	2	256
L18	$3 \times 3 \times 256$	Conv	1	128
L19	$3 \times 3 \times 128$	Conv	2	256
L20	$2 \times 2 \times 256$	Conv	1	64
L21	$1 \times 1 \times 64$	Conv	2	128

$$r_k = \begin{cases} \left\{1, 2, \frac{1}{2}\right\}, & k = 1, \\ \left\{1, 2, \frac{1}{2}, 3, \frac{1}{3}\right\}, & k \in [2, 6]. \end{cases} \quad (8)$$

In addition, a square default box with a scale of $\sqrt{S_k S_{k+1}}$ is added when $k \in [2, 6]$. Assuming an input image of size $300 \times 300 \times 3$, the number of default boxes is 1917.

In this paper, OM-SSD algorithm for extracting the ROI containing the conveyor belt edge and the exposed idler is proposed. On the basis of M-SSD, OM-SSD sets the aspect ratios of L11's default boxes to the same ratios as those of other layers, and L14–L21 are removed; L11 and L13 are the only source layers for prediction, and the reason is as follows. For the extraction of the ROI, it is the same thing as detecting the image of the exposed idler with an extended small patch of the conveyor belt. Since the distance between the inspection robot and the idler does not change very much, the image size of the ROI varies in a small range, and the default boxes with unrelated scale contribute almost nothing to the detection but slowing down the speed [18], so the prediction source layers L14–L21, which are used to generate large scale default boxes in M-SSD, are removed. Since the idler is elongated, the aspect ratio of the exposed idler varies greatly, so the aspect ratios of L11 are expanded. Since most feature extraction layers in OM-SSD are DSCs, the number of the parameters in the model is small, and then, OM-SSD has less trouble with overfitting [27]. The original loss function of SSD [18] is adopted for training:

$$L(x, c, l, g) = \frac{(L_{\text{conf}}(x, c) + \alpha L_{\text{loc}}(x, l, g))}{N_b}, \quad (9)$$

where N_b is the number of matched default boxes, x is the match indicator of the default box and the ground truth, c is the prediction confidence of multiple classes, and l and g are the parameters of the predicted box and the ground truth box, respectively. Confidence loss $L_{\text{conf}}(x, c)$ and localization loss $L_{\text{loc}}(x, l, g)$ are the main components of the loss function and are weighted by weight term α , which is set to 1.

In order to extract the ROI efficiently, the input image is divided into two subimages: imgO and imgI, which may contain the outer idler and the inner idler, respectively. Then, they are resized to $300 \times 300 \times 3$ and input into OM-SSD. The detector outputs the label indicating the classification of the idler and the bounding box of the ROI.

2.3. Conveyor Belt Edge Detection Algorithm Based on the Hough Line Transform. The conveyor belt edge in the ROI can be regarded as a straight line. The conventional algorithms of line detection can be divided into two categories: Hough transform-based and line segment-based algorithms [28]. The latter has the advantages of high efficiency and accuracy; however, this method is based on region grow, and if a long line is blocked or blurred partially, it is often detected as multiple lines. The former does not suffer from this drawback and is more robust. The Hough line transform maps the edge point (x, y) in the image space to a sine-like curve in the parameter space by

$$\rho = x\cos\theta + y\sin\theta, \quad (10)$$

where ρ is the distance of the line and the origin and θ is the angle between the x -axis and the line. The curves corresponding to the edge points on the same line will intersect at a point (ρ_0, θ_0) . The plausible lines can be obtained by substituting the coordinates of the intersection points of the curves into equation (10).

The conveyor belt edge detection algorithm based on the Hough line transform consists of six steps and is described in Algorithm 1.

To reduce noise and keep sharpness, the kernel size of the Gaussian filter is set to 3, and the standard deviation is 1.0. The low and high threshold of the Canny edge detector are 0.2 and 0.4 of the maximum gradient, respectively, and the accumulator threshold for the plausible lines extraction is set to 30 empirically. The prior knowledge of the conveyor belt edge is as follows. The angle between the conveyor belt edge and the x -axis is no more than 15° , and then, the plausible lines with $|\theta| > 15^\circ$ are discarded. In addition, ROI_I may contain the edges of the upper and lower surface of the conveyor belt, and generally, the upper one is more prominent, so the uppermost proposal line is considered as the conveyor belt edge. ROI_O only contains the upper surface edge; thus, the longer proposal line at the bottom of the ROI_O is extracted.

2.4. Idler Outer Edge Detection Based on Template Matching

2.4.1. Features of the Idler Outer Edge in the Image. The general structure of the idler is a cylinder, and the shape of its outer edge is a standard circle. However, the image of

the idler outer edge obtained by the inspection robot is an elliptical arc with large gaps, varying scales, and interference with similar arcs, as shown in Figure 2; this is caused by the following reasons: (1) due to the shooting position, the angle between the plane of the idler outer edge and the image plane is not 0° , which results in the standard circle being compressed vertically, and only the upper half of the outer idler outer edge can be obtained and the lower part of the inner idler outer edge is blocked by the bracket. (2) The idler outer edge may be sheared horizontally with variations in scale, which is caused by the movement of the inspection robot. (3) The protruding edge on the idler outer edge will induce a pseudoarc, and the shadow of the conveyor belt edge on the idler will also appear as an interference.

2.4.2. Idler Outer Edge Detection Algorithm. The general arc detection algorithms can be classified into two categories: Hough transform-based [30] and gradient region growth-based algorithms [31]. The former suffers from tremendous computation, while the latter is more suitable for the images with simple content. More importantly, they do not perform well in detecting an elliptical arc.

The elliptical arc of the idler outer edge can be generated by horizontal scaling, vertical scaling, and horizontal shearing of a semicircle, according to which an idler outer edge detection algorithm based on template matching is proposed. Firstly, the template elliptical arcs are generated based on the prior transformation parameters, and then, the corresponding template elliptical arcs and the edge points in I_e are traversed to find the one with the highest matching confidence. The proposed algorithm is composed of the following three steps: (1) given the radius r_i and the vertical scaling factor s_j , the coordinate (w, z) of an arbitrary point on the base arc is determined as

$$z = s_j \sqrt{r_i^2 - (w - r_i)^2}, \quad (11)$$

where $w \in [0, 2r_i]$ and r_i and s_j are used to scale the base arc horizontally and vertically. Given the horizontal shearing factor c_k , the coordinate (x_a, y_a) of an arbitrary point on the template elliptical arc can be determined as

$$\begin{cases} x_a = w + c_k z, \\ y_a = z. \end{cases} \quad (12)$$

The images of the representative template elliptical arcs with different parameters are shown in Figure 3. (2) We consider $I_e(x, y)$ in I_e as the origin coordinate, count the number of the edge pixels $n_e(x, y)$ on the template elliptical arc route and the number of the valid pixels $n_v(x, y)$, and traverse all the template elliptical arcs determined by the combinations of $r_i = R_{\min} + iR_{\text{step}}$, $i \in [0, 1, 2, \dots, N_r - 1]$, $s_j = S_{\min} + jS_{\text{step}}$, $j \in [0, 1, 2, \dots, N_s - 1]$, and $c_k = C_{\min} + kC_{\text{step}}$, $k \in [0, 1, 2, \dots, N_c - 1]$ (R_{\min} , S_{\min} , and C_{\min} are the minimum values of the radius, the vertical scaling factor, and the horizontal shearing factor, R_{step} , S_{step} , and C_{step} are their steps, and N_r , N_s , and N_c are their numbers, respectively), and the matching confidence $p(x, y, i, j, k)$ that

Input: ROI image ROI_O or ROI_I.
Output: Coordinates of the start and end points of the proposal line.

- (1) Convert ROI_O or ROI_I into gray image;
- (2) Denoise the gray image by using Gaussian filter;
- (3) Obtain the edge map I_e by using Canny edge detector [29];
- (4) Map the edge points in I_e to the parameter space by using Hough line transform, then set the accumulators;
- (5) Extract the plausible lines by setting a proper accumulator threshold;
- (6) Extract the proposal line from the plausible lines by the prior knowledge of the conveyor belt edge.

ALGORITHM 1: Conveyor belt edge detection algorithm based on the Hough line transform.

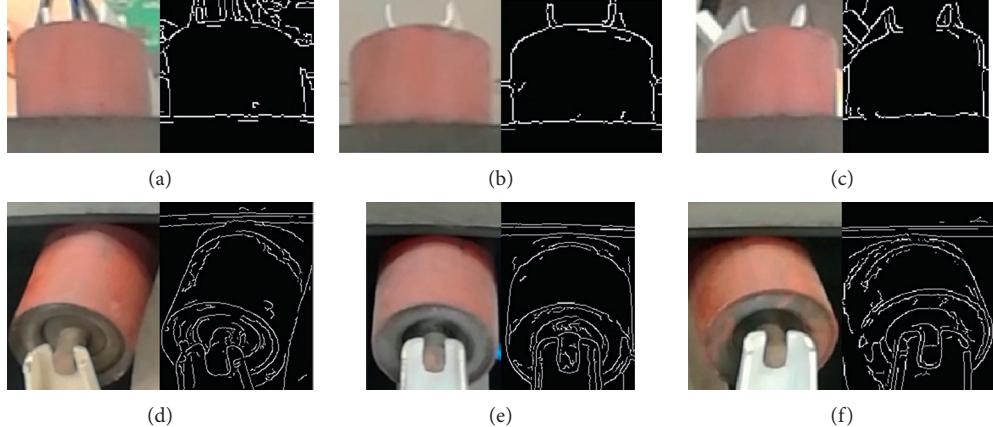


FIGURE 2: Typical images of the ROI and their edge maps. With the movement of the inspection robot, the position of the ROI in the image changes from left to right, the idler outer edge presents an elliptical arc, and the edge map of the ROI_I is disturbed by similar elliptical arcs. (a) ROI_O on the left, (b) ROI_O in the middle, (c) ROI_O on the right, (d) ROI_I on the left, (e) ROI_I in the middle, and (f) ROI_I on the right.

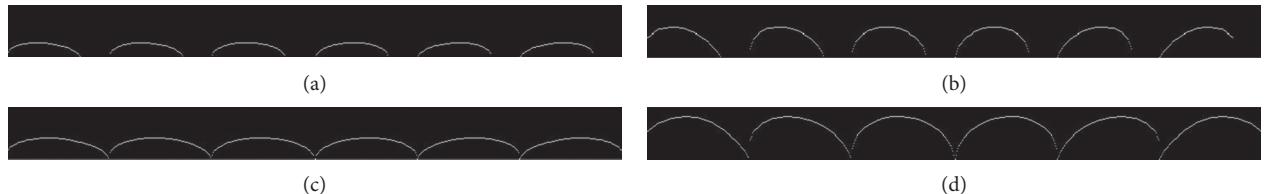


FIGURE 3: Typical template elliptical arcs with different parameters. (a) $r_i = 43$, $s_j = 0.4$, and $c_k = -0.5, -0.3, -0.1, 0.1, 0.3$, and 0.5 . (b) $r_i = 43$, $s_j = 0.8$, and $c_k = -0.5, -0.3, -0.1, 0.1, 0.3$, and 0.5 . (c) $r_i = 60$, $s_j = 0.4$, and $c_k = -0.5, -0.3, -0.1, 0.1, 0.3$, and 0.5 . (d) $r_i = 60$, $s_j = 0.8$, and $c_k = -0.5, -0.3, -0.1, 0.1, 0.3$, and 0.5 .

indicates the probability of a true idler outer edge can be defined as

$$p(x, y, i, j, k) = \max \frac{n_e(x, y)}{n_v(x, y)}. \quad (13)$$

C_{\min} varies with the horizontal location of the ROI in the image. Assuming that the abscissa value of the left boundary of ROI_O or ROI_I is x_l and its ratio to the width of imgO or imgI is r_x , C_{\min} can be determined as

$$C_{\min} = C_{\text{step}} \text{ floor}\left(\frac{r_x(C_r - C_l) + C_l}{C_{\text{step}}}\right), \quad (14)$$

by which C_{\min} is aligned to an integral multiple of C_{step} within $[C_l, C_r]$. (3) We traverse all of the edge points in I_e , repeat step (2), calculate all the $p(x, y, i, j, k)$, and then, sort them in descending order and output the parameters (x, y, i, j, k) of top K. The origin coordinate of the idler outer edge is located in the left half of I_e ; thus, only the edge points on the left half are traversed to speed up. Generally, matching confidence of the true idler outer edge is higher than that of the pseudoarc; then, the parameters with the highest matching confidence are used to describe the idler outer edge.

We summarize the idler outer edge detection algorithm based on template matching in Algorithm 2.

Input: Edge map I_e , prior transformation parameters R_{\min} , R_{step} , N_r , S_{\min} , S_{step} , N_s , C_l , C_r , C_{step} and N_c .
Output: Idler outer edge parameters (x, y, i, j, k) .

- (1) Compute the template elliptical arcs by using equations (11) and (12);
- (2) Consider the edge point $I_e(x, y)$ in the left half of I_e as the origin coordinate, compute the matching confidence $p(x, y, i, j, k)$ by using equation (13);
- (3) Traverse all the edge points $I_e(x, y)$ in I_e and repeat step 2, then output the parameters (x, y, i, j, k) with the highest matching confidence.

ALGORITHM 2: Idler outer edge detection algorithm based on template matching.

2.5. Geometric Correction Based on Homography Transformation

2.5.1. Equivalence of Conveyor Belt Imaging. Geometric correction is required for the measurement with a monocular camera. The surface of the trough conveyor belt is not a plane, and distortion will occur when the surface is projected onto the image plane.

To correct this distortion, the imaging process is analyzed as follows. The deviation of the conveyor belt is defined in its cross-section plane, and the imaging model is shown in Figure 4. a'' and d'' are the vertices of the outer edges of the inner and outer idlers, b_1 and c_1 are the edge points of the conveyor belt upper surface, and their projection points on the image plane I are a , d , b , and c , respectively. The thickness of the conveyor belt should not be ignored and is marked as $b_1 b_2$ or $c_1 c_2$, where b_2 and c_2 are the edge points of the lower surface of the conveyor belt. According to the geometry, the length of the exposed idlers $a''b_2$ and c_2d'' can be obtained by the internal and external parameters of the camera as well as the pixel distance of ab and cd , and they can be used to estimate the DD of the conveyor belt. However, to obtain these parameters directly, 3D perspective transformation is needed, which requires more reference points that are difficult to get in engineering.

To address this issue, the imaging process is equivalent to two projections. We suppose there is a horizontal auxiliary plane I' with a'' and d'' on it. b' and c' are the projection points of b_1 and c_1 on I' with O_c as the perspective center, and the first equivalent projection is that the image of the 3D objects such as the conveyor belt and the exposed idlers are projected to plane I' . The second one is that the image on I' is projected to I . The process of the measurement by a monocular camera can be regarded as the inverse imaging process described above.

2.5.2. Geometric Correction Algorithm. As the imaging process is equivalent to two projections, the geometric correction is simplified from 3D projection transformation

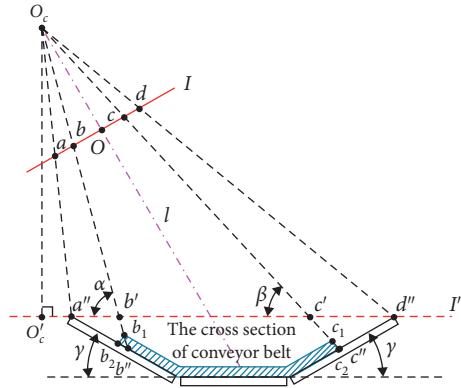


FIGURE 4: The imaging model of the conveyor belt in the cross-section plane.

into a combination of plane homography transformation and plane geometric transformation.

For any point (u, v) on the image plane I , there is a nonsingular matrix H of size 3×3 and a scale factor ρ that can map (u, v) to (x', y') on I' linearly. The homogeneous expression is as follows:

$$\begin{bmatrix} x' \\ \frac{y'}{\rho} \\ \frac{1}{\rho} \end{bmatrix} = \begin{bmatrix} x'_a \\ y'_a \\ 1 \end{bmatrix} = H \begin{bmatrix} u \\ v \\ 1 \end{bmatrix}, \quad (15)$$

$$H = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{bmatrix}.$$

There are eight degrees of freedom in H , and four pairs of calibration points, at least, are needed to solve equation (15). The calibration points can be obtained by a rectangular calibration board placed on I' . The width and aspect ratio of the calibration board are W_a and R_a , and the calibration points

(u_i, v_i) ($i = 1, 2, 3, 4$) are the coordinates of the upper left, upper right, lower left, and lower right corner of the calibration board on the image plane I , respectively. The corrected coordinates (u_{ai}, v_{ai}) ($i = 1, 2, 3, 4$) can be defined as

$$\left\{ \begin{array}{l} u_{a1} = u_1, \\ v_{a1} = v_1, \\ u_{a2} = u_1 + \sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2}, \\ v_{a2} = v_1, \\ u_{a3} = u_1, \\ v_{a3} = v_1 + \frac{\sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2}}{R_a}, \\ u_{a4} = u_{a2}, \\ v_{a4} = v_{a3}, \end{array} \right. \quad (16)$$

and then, ρ is calculated by

$$\rho = \frac{W_a}{\sqrt{(u_2 - u_1)^2 + (v_2 - v_1)^2}}, \quad (17)$$

and the elements in H can be obtained by solving

$$\left\{ \begin{array}{l} u_{ai} = \frac{h_{11}u_i + h_{12}v_i + h_{13}}{h_{31}u_i + h_{32}v_i + 1}, \\ v_{ai} = \frac{h_{21}u_i + h_{22}v_i + h_{23}}{h_{31}u_i + h_{32}v_i + 1}, \end{array} \right. \quad i = 1, 2, 3, 4. \quad (18)$$

When the coordinates of a , d , b , and c are detected, they can be transformed to a'' , d'' , b' , and c' by equation (15). The vertical projection point of the camera optical center O_C in plane I' is O'_C , the vertical and horizontal distance between O_C and a'' in the cross section plane are $O_C O'_C$ and $O'_C a''$, and the angle between the idler center axis and the horizontal plane is γ ; then, the length of the exposed idlers can be calculated by the plane geometric transformation as follows:

$$\left\{ \begin{array}{l} a''b_2 = a''b' \cos\gamma + \frac{a''b' \sin\gamma - b_1 b_2}{\tan(\arctan(O_C O'_C / O'_C a'') + a''b') - \gamma}, \\ c_2 d'' = c' d'' \cos\gamma - \frac{c' d'' \sin\gamma - b_1 b_2}{\tan(\arctan(O_C O'_C / O'_C a'') + a''c') + \gamma}. \end{array} \right. \quad (19)$$

2.6. DD Estimation. The slope of the conveyor belt edge is gentle, and the midpoints of the lines detected in Section 2.3 can be regarded as the edge points b and c in the image. Meanwhile, the vertices of the detected elliptical arcs in

Section 2.4 are used as the vertices a and d in the image. Then, $a''b_2$ and c_2d'' can be calculated by equation (15) and (19). According to the Chinese industry standard (GB/T 10595-2017), the deviation of the conveyor belt is defined as the deviation between the centerlines of the conveyor belt and the belt conveyor, and the conveyor belt is considered to be in normal operation when the deviation is within 5% of the belt width, and early warning or shutdown command should be issued when that exceeds 5% consistently [4].

The DD is defined as the percentage of the conveyor belt deviation and its width W_b . When both of $a''b_2$ and c_2d'' can be obtained, the DD is calculated as follows:

$$\text{DD} = \frac{a''b_2 - c_2d''}{2W_b} \times 100\%. \quad (20)$$

However, $a''b_2$ and c_2d'' are not always available. When there is only $a''b_2$, the DD should be calculated by

$$\text{DD} = \frac{a''b_2 - 0.5(W_i - W_b)}{W_b} \times 100\%, \quad (21)$$

where W_i is the groove length of the idler set, which is longer than $d''d''$. $(W_i - W_b)$ is the groove length of the exposed idler set, which can also be obtained from the previous normal detection results:

$$W_i - W_b = a''b_2 + c_2d''. \quad (22)$$

When there is only c_2d'' , the DD should be calculated by

$$\text{DD} = \frac{0.5(W_i - W_b) - c_2d''}{W_b} \times 100\%. \quad (23)$$

If $a''b_2$ and c_2d'' are both unavailable, this detection process will be aborted. The DD is a signed variable, a positive one indicates that the conveyor belt deviates to the outer side, and a negative one indicates that of the inner side.

3. Results and Discussion

An experimental setup was built to validate the proposed method. To illustrate the advantages of the proposed ROI detector, OM-SSD is compared with M-SSD in the self-built dataset. Then, the performance of the deviation detection algorithm is verified at different standard DDs and shooting heights, and the preferred shooting height is given through the comparative experiments. Furthermore, the time consumption of the proposed algorithms is examined on a NVIDIA Jetson TX2.

3.1. Experimental Setup. The inspection robot and belt conveyor experimental platform is shown in Figure 5. The inspection robot runs on the rack track beside the belt conveyor and is equipped with a source light to make up the poor illumination. The mobile camera (camera sensor: SONY IMX298, resolution: 1080×1920 , aperture: f/2.0) is fixed on the metal frame, which moves along with the inspection robot and is height adjustable. The belt conveyor is 7.7 m in length, 1 m in width, and 1.55 m in height (from the

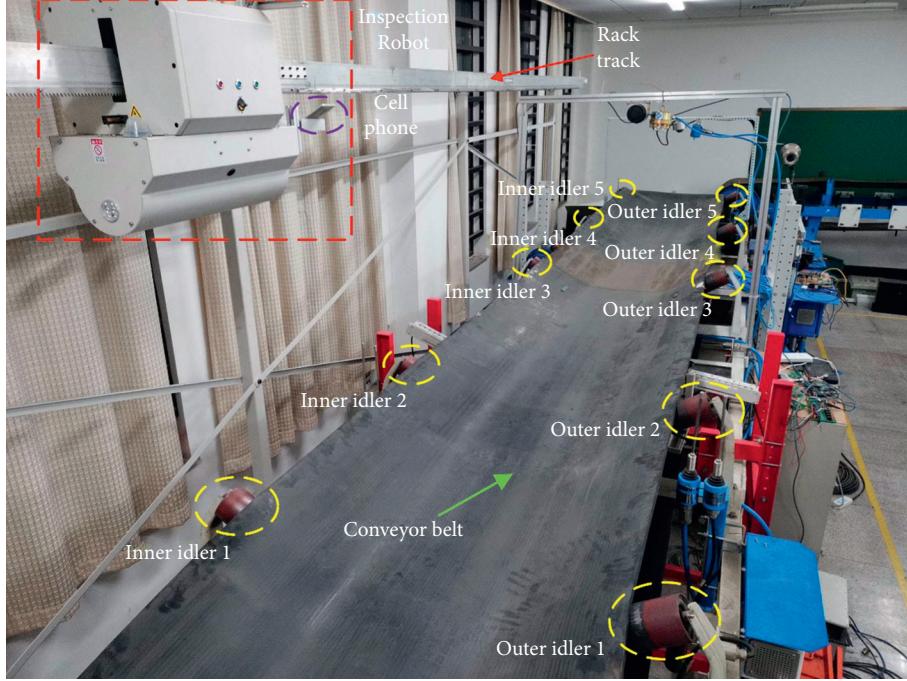


FIGURE 5: The inspection robot and belt conveyor experimental platform.

vertex of the side idler outer edge to the ground), and the belt is 11 mm in thickness. The upper belt is supported by 5 sets of idlers (diameter: 89 mm, $\gamma: 30^\circ$) and only the side idlers of each set could be captured, which are named as inner idler 1–5 and outer idler 1–5, respectively.

The belt conveyor in mine roadway is close to the bearing wall, so the horizontal distance $O_C'a'$ cannot vary too much and is set to 0.35 m empirically. Due to the limitation of the tunnel height and the integrity of the imaging content, the shooting height from O_c to the ground varies between 2.4 m and 3.0 m, and $O_C O'_C$ can be obtained by subtracting the belt conveyor height from the shooting height. To simulate the effect of the shooting height, it is set to 2.4 m, 2.6 m, 2.8 m, and 3.0 m, respectively. To simulate the deviation of the belt conveyor, the standard DD is adjusted to +5%, 0%, and -5%, respectively. It should be noted that only one set of the idlers and the corresponding conveyor belt edges can be used to indicate the standard DD due to the twist of the conveyor belt when it deviates. In these 12 combinations, the horizontal and vertical perspective angles are set to -60° and 0° , respectively, the inspection robot captures an image every 0.06 m along the conveyor belt from the head to tail, and then, 12 sets of images are obtained and the images without idlers are discarded. The validation dataset of the proposed deviation detection method is composed of 184 images containing the specific idlers, which are picked out from each set, when the standard DD is +5% or 0%, the images containing the idler 2 are picked out, and when the standard DD is -5%, the images containing the idler 4 are picked out; the typical images are shown in Figure 6. Each of the remaining images is cropped to two subimages of 1080 × 960 × 3 pixel for the fine tuning of the ROI detector. The cropped subimages are randomly sorted and divided into

three sets for training, validation, and testing, and the number of them is 551, 237, and 388, respectively.

The desktop computer configured as Intel i7-7820X 3.6 GHz CPU, 16 GB memory, NVIDIA RTX 2080Ti, and Ubuntu 18.04 is used for the fine tuning of the ROI detector, and the performance validation of the proposed deviation detection method is conducted on a NVIDIA Jetson TX2, which is configured as Ubuntu 16.04, Python 2.7, OpenCV 4.11, and Caffe. The algorithm proposed in Section 2.4 is written in C++ and embedded in the source code of OpenCV 4.11 to compile. The integration of all the proposed algorithms is based on Python 2.7.

3.2. Training Result of the ROI Detector. The backbone network parameters of OM-SSD adopt the parameters of M-SSD (<https://github.com/chuanqi305/MobileNet-SSD>) with a mean average precision (mAP) of 0.727, which is pretrained on PASCAL VOC0712. OM-SSD is fine tuned on the dataset mentioned in Section 3.1. The optimization algorithm is the root mean square prop (RMSPProp) with a base learning rate of 0.0002, which is reduced by half at the 5000th and 20000th iteration. The fine tuning ends after 30000 iterations. Before deploying the trained model, the batchnorm layers and the scale layers are merged with the previous convolution layers, and the compute unified device architecture (CUDA) units are used in the convolution computation to further speed up the forward inference. To investigate the influence of the different prediction source layers on the model, a comparative experiment of OM-SSD with different prediction source layers was implemented, and the configuration is shown in Table 2, where the selected layers are used as the prediction source layers, and all the feature extraction layers behind the last selected one are

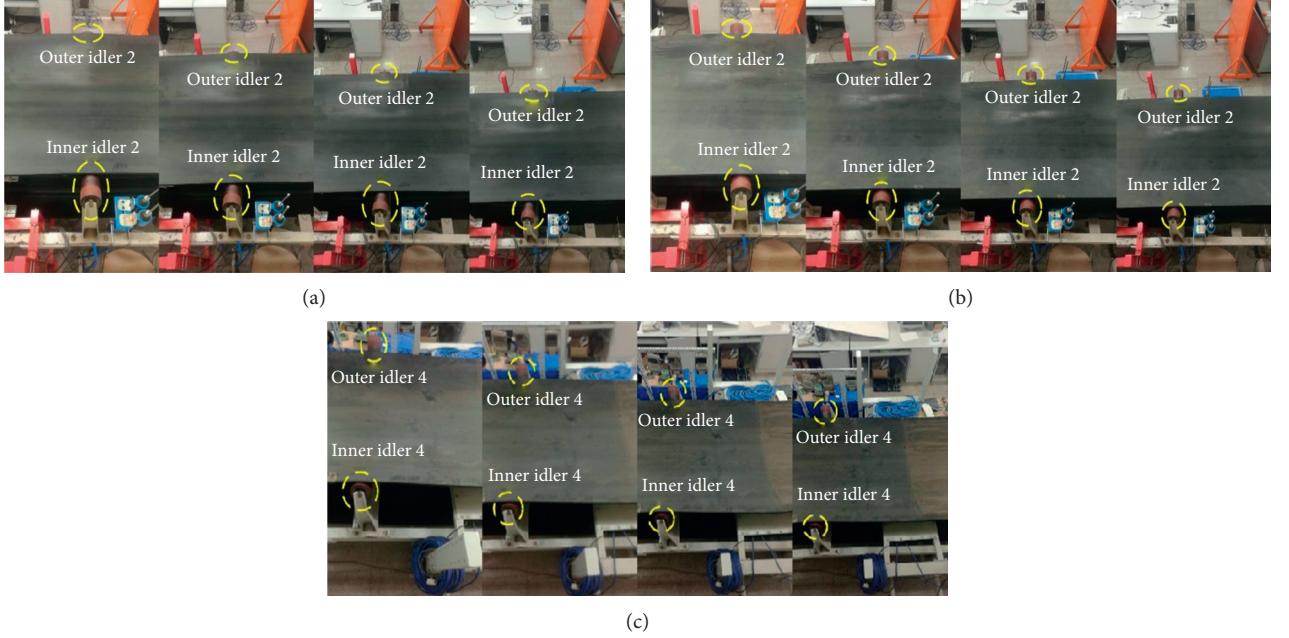


FIGURE 6: Conveyor belt images captured at different DDs and heights. (a) DD = +5%, shooting height from left to right: 2.4 m, 2.6 m, 2.8 m, and 3.0 m. (b) DD = 0%, shooting height from left to right: 2.4 m, 2.6 m, 2.8 m, and 3.0 m. (c) DD = -5%, shooting height from left to right: 2.4 m, 2.6 m, 2.8 m, and 3.0 m.

TABLE 2: Configuration of OM-SSD with different prediction source layers.

Layer model	L11	L13	L15	L17	L19	L21
OM-SSD-1	✓					
OM-SSD-2	✓		✓			
OM-SSD-3	✓	✓		✓		
OM-SSD-4	✓	✓	✓		✓	
OM-SSD-5	✓	✓	✓		✓	✓
OM-SSD-6	✓	✓	✓	✓	✓	✓

removed. The training loss and the mAP of each OM-SSD- x ($x = 1, 2, \dots, 6$) and M-SSD are shown in Figures 7 and 8. The results show that the final training loss of each model is within 2.5, and the mAP is over 0.98, but the subtle difference may affect the detection accuracy. Among the models, the training loss of OM-SSD-2 is less than 2.0 in the late stage and finally converges to within 1.0, while that of other models is more than 1.0. The mAP of OM-SSD-2 is over 0.995 in the late stage and ends up with 0.9995, which is better than that of others, and this shows that the features used to detect the ROI are mainly derived from L11 and L13. The mAP curve of M-SSD shows a negative spike in the late stage, indicating that the model is not robust. Therefore, it can be concluded that the proposed OM-SSD-2 is more suitable for the detection of the ROI containing the conveyor belt edge and the exposed idler.

3.3. Results of the Deviation Detection

3.3.1. Accuracy of the ROI Detector. The well-trained OM-SSD-2 is used to extract ROI from the images in the validation dataset of the proposed deviation detection method. According to the shooting sequence, the images in each set

are numbered as 0, 1, 2, …, respectively. Due to the different shooting heights, the number of the images in each set varies from 11 to 20, and the higher the shooting height, the larger the number of the images.

The detection confidences of ROI_Is and ROI_Os are depicted in Figure 9. When the ROI is detected, the confidence varies in [0.2, 1.0], and a higher confidence indicates a more accurate result. The confidence is 0 when the ROI is missed. From the perspective of the ROI, the number of the missed ROI_I and ROI_O is 7 and 19, respectively, that is, the missed detection rate of ROI_I (3.8%) is lower than that of ROI_O (10.9%), and the former shows a higher average confidence (77.6%) than the latter (44.9%). The reason is that more textures and details of the inner idler can be captured, and the features are more distinguishable. From the perspective of the shooting height, the missed detection rates at the four shooting heights are 5.7%, 6.0%, 15.3%, and 2.6%, respectively, and the lower ones are at 2.4 m and 3.0 m. When the shooting height is 2.4 m, all the ROI_I are detected, and the confidences are close to 1.0, while the ROI_Os are missed 4 times. When the shooting height is

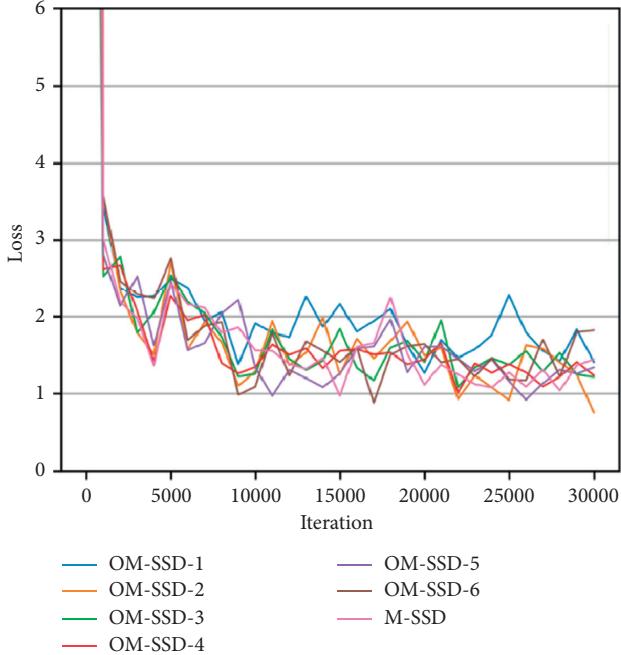


FIGURE 7: The training losses of different models.

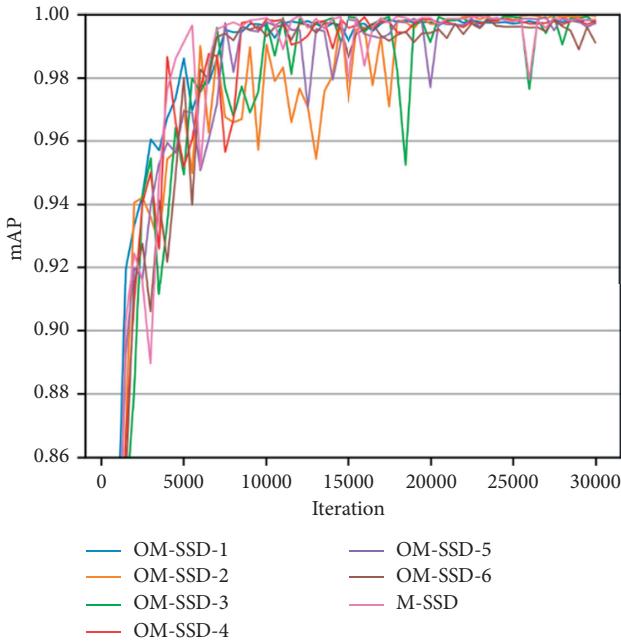


FIGURE 8: The mAPs of different models during training.

3.0 m, the number of the missed ROI_I s increases to 2, while that of ROI_O s decreases to 1. Part of the reason is that when the shooting height is lower, the camera is closer to the inner idler and more textures and details can be captured, and when the shooting height is higher, more content of the outer idler can be captured, while some details of the inner idler are lost. It can be concluded that, to achieve better ROI detection accuracy, the shooting height of 2.4 m and 3.0 m is preferred.

3.3.2. Accuracy of the DD Estimation. When ROIs are detected, the DD can be estimated by performing the conveyor belt edge detection, the idler outer edge detection, the geometric correction, and the DD estimation algorithm.

The empirical values of the prior transformation parameters for the idler outer edge detection are shown in Tables 3 and 4. The typical results of the conveyor belt edge detection and the idler outer edge detection are shown in Figure 10. It can be seen that the conveyor belt edges and the idler outer edges are detected correctly. Meanwhile, the conveyor belt edge points and the vertices of the idler outer edges are marked with circles.

The estimated DDs at each standard DD and shooting height are shown in Figure 11. It can be revealed that the fluctuation of the detection results increased along with the decrease of the standard DD from +5% to -5%. This can be explained by the fact that a larger size of ROI can lead to higher detection accuracy of the edges, and the image of the inner idler is bigger than that of the outer idler. As the standard DD decreases, so does the length of the inner exposed idler, and the detection accuracy degrades. Another reason is that when only one of ROI_O and ROI_I is detected, errors will be introduced by using the previous normal detection results. When a false conveyor belt edge or a false idler outer edge is detected, the estimated DD deviates from the ground truth. It can also be revealed that the image number affects the result of the DD estimation irregularly; that is to say, the horizontal shearing of the idler image is not the major factor that affects the detection accuracy, which verifies the validation of the proposed idler outer edge detection algorithm indirectly.

For the estimated DDs at each standard DD and shooting height, the mean and standard deviation (SD) are used to assess the accuracy and precision, as shown in Table 5. It shows that the estimated DDs at 2.4 m are closer to the ground truth than that at other shooting heights. When the standard DDs are +5% and -5%, the means of the estimated DDs are 5.04% and -5.03% at 2.4 m, which are the best in all shooting heights, and the SDs are within 0.20%. When the standard DD is 0%, the mean of the estimated DDs at each shooting height is within 0.3%. In addition, RMSE and MAE [32] are used to assess the average accuracy at each shooting height. As is shown in Table 5, the values of RMSE and MAE at 2.4 m are 0.37% and 0.44%, respectively, which are the smallest. Considering the width of the conveyor belt in the experiment is 1 m, the RMSE and MAE are 3.7 mm and 4.4 mm, respectively, which is much more accurate than the error of 16 mm in [12]. Meanwhile, 2.4 m is also the preferred shooting height for the ROI detection. Therefore, it can be concluded that 2.4 m is the preferred shooting height for the deviation detection of the belt conveyor.

3.3.3. Time Consumption. The time consumption of the proposed algorithm is shown in Table 6. The average time consumption of the deviation detection algorithm is 135.5 ms and is faster than that of 260 ms in [12]. The time

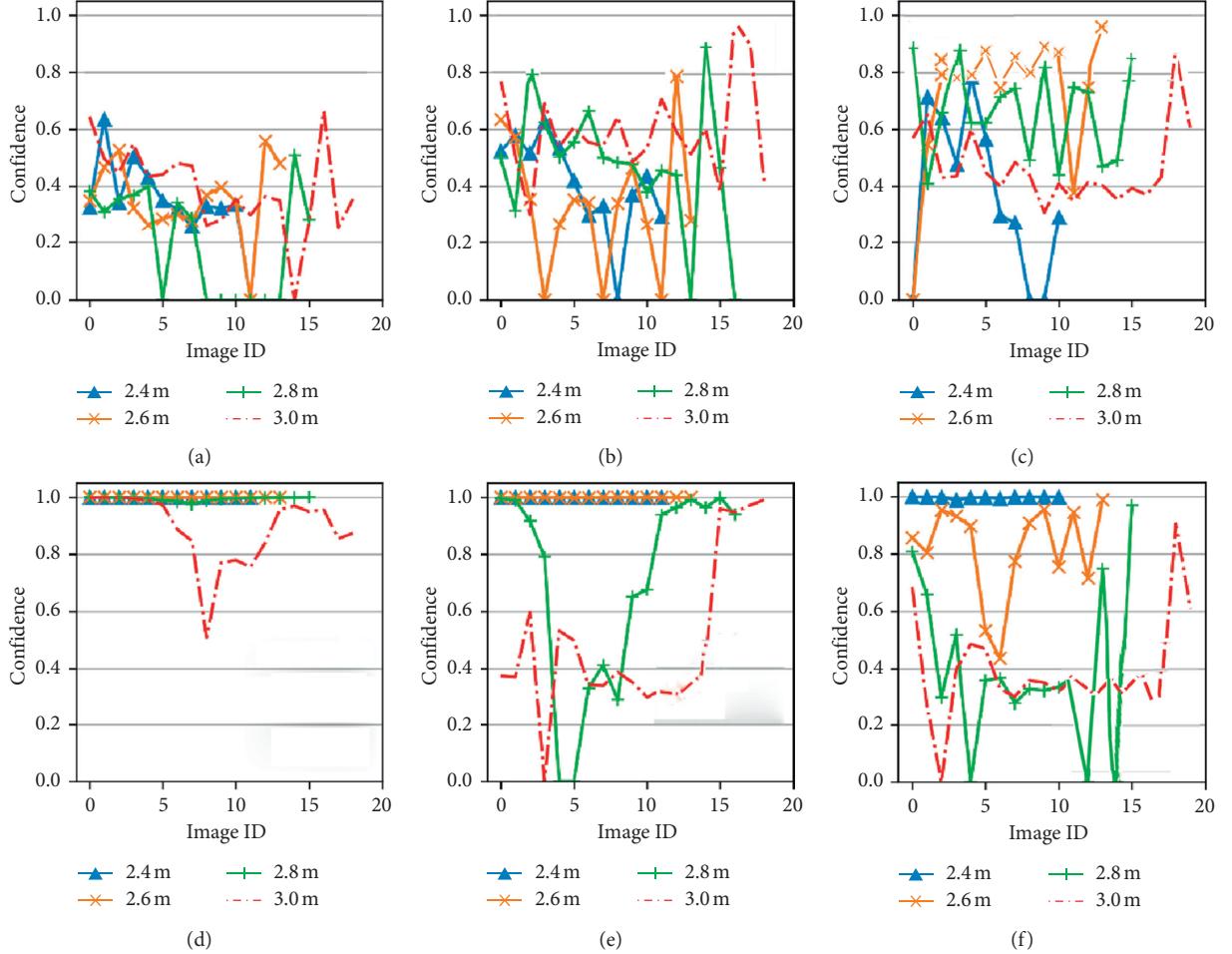


FIGURE 9: Detection confidences at different standard DDs and heights. (a) ROI_O, DD: +5%. (b) ROI_O, DD: 0%. (c) ROI_O, DD: -5%. (d) ROI_I, DD: +5%. (e) ROI_I, DD: 0%. (f) ROI_I, DD: -5%.

TABLE 3: The prior transformation parameters for the inner idler outer edge detection.

Height (m)	Parameter										
	R_{\min} (pixel)	R_{step} (pixel)	N_r	S_{\min}	S_{step}	N_s	C_l	C_r	C_{step}	N_c	
2.4	65	1	8	0.6	0.05	4	-0.6	0.6	0.05	4	
2.6	52	1	8	0.6	0.05	4	-0.6	0.6	0.05	4	
2.8	47	1	8	0.5	0.05	4	-0.6	0.6	0.05	4	
3.0	43	1	8	0.5	0.05	4	-0.6	0.6	0.05	4	

TABLE 4: The prior transformation parameters for the outer idler outer edge detection.

Height (m)	Parameter										
	R_{\min} (pixel)	R_{step} (pixel)	N_r	S_{\min}	S_{step}	N_s	C_l	C_r	C_{step}	N_c	
2.4	41	1	8	0.35	0.05	4	-0.5	0.5	0.05	4	
2.6	37	1	8	0.25	0.05	4	-0.5	0.5	0.05	4	
2.8	36	1	8	0.2	0.05	4	-0.5	0.5	0.05	4	
3.0	34	1	8	0.2	0.05	4	-0.5	0.5	0.05	4	

consumption is mainly composed of three components: the time consumption of OM-SSD-2, the conveyor belt edge detection, and the idler outer edge detection algorithm,

and the time consumption of the geometric correction and the DD estimation can be ignored. During each deviation detection process, OM-SSD-2 is called twice; if

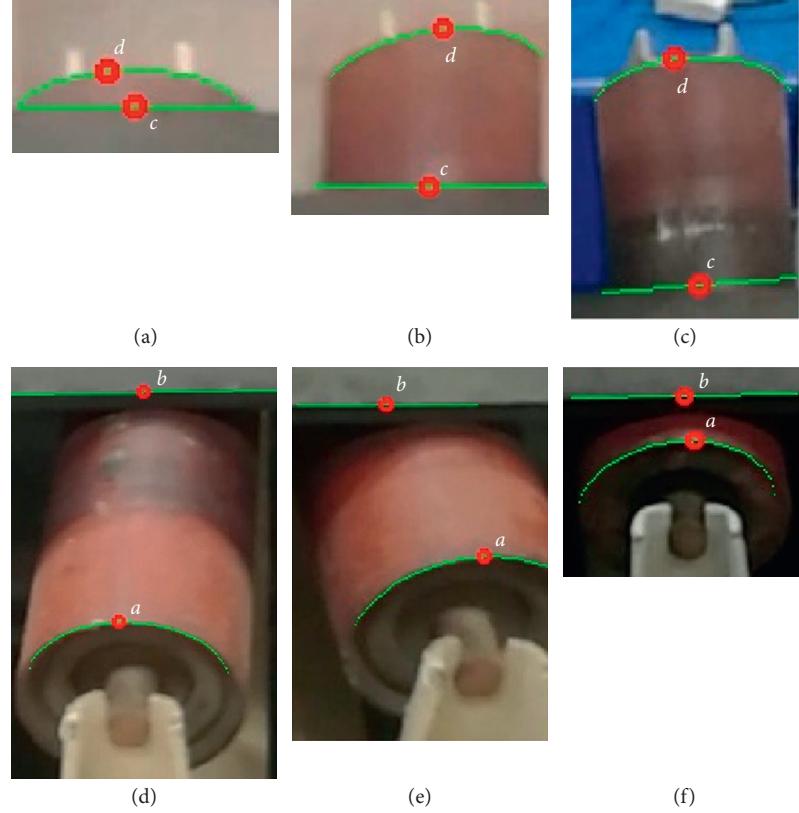


FIGURE 10: Detection results of the idlers, lines, and arcs. (a) Outer idler, DD: +5%. (b) Outer idler, DD: 0%. (c) Outer idler, DD: -5%. (d) Inner idler, DD: +5%. (e) Inner idler, DD: 0%. (f) Inner idler, DD: -5%.

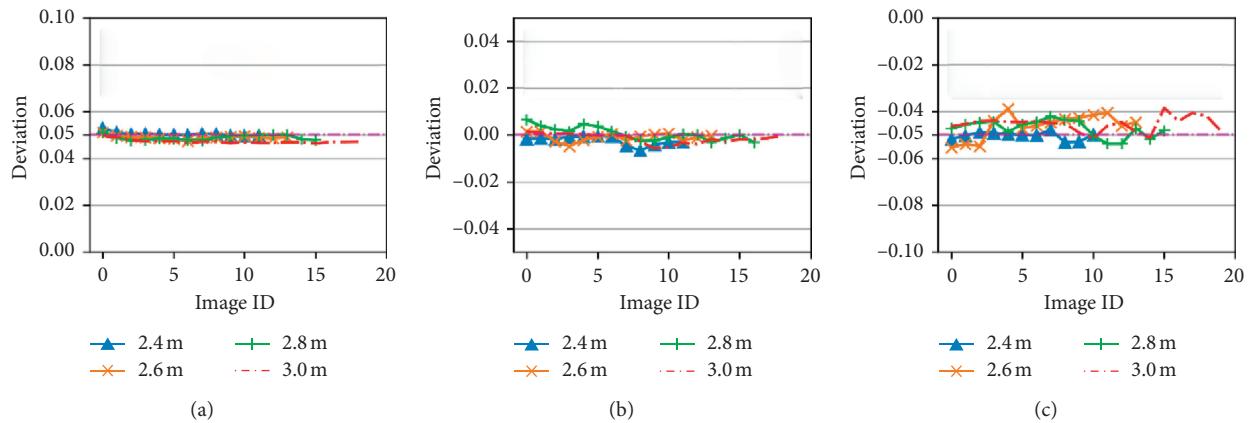


FIGURE 11: Estimated DDs at each standard DD and shooting height. (a) DD: +5%. (b) DD: 0%. (c) DD: -5%.

TABLE 5: Deviation detection results of the belt conveyor.

Height (m)	DD: +5%		DD: 0%		DD: +5%		RMSE (%)	MAE (%)
	Mean (%)	SD	Mean (%)	SD	Mean (%)	SD (%)		
2.4	5.04	0.08	-0.24	0.19	-5.03	0.16	0.37	0.44
2.6	4.90	0.07	-0.10	0.17	-4.60	0.50	0.69	0.87
2.8	4.89	0.10	-0.06	0.28	-4.72	0.35	0.56	0.76
3.0	4.72	0.08	-0.15	0.20	-4.54	0.32	0.67	0.93

TABLE 6: Time consumption of the proposed algorithm.

Algorithm	Time consumption (ms)
OM-SSD-2	41.6
Conveyor belt edge detection	4.7
Idler outer edge detection	23
Geometric correction and DD estimation	0.1
Belt conveyor deviation detection	135.5

one of the inner and outer idler is blocked by the conveyor belt and not detected, the remaining components will only be called once.

It is time consuming to perform the homography transformation on a high-resolution color image (i.e., $1080 \times 1920 \times 3$). In this paper, all the detection algorithms are performed on the original image, and the geometric correction and the DD estimation are implemented on only four points, and it improves the real-time performance significantly.

4. Conclusions

In this paper, a deviation detection method of the belt conveyor based on inspection robot and deep learning is proposed to detect the deviation at its any position. Firstly, the image of the belt conveyor is captured by the inspection robot continuously, and the ROI is extracted by the proposed ROI detector based on M-SSD. Secondly, the Hough line transform algorithm is used to detect the conveyor belt edge, and the elliptical arc detection algorithm based on template matching is proposed to detect the idler outer edge. Finally, a geometric correction algorithm based on homography transformation is proposed to correct the coordinates of the conveyor belt edge point and the idler outer edge point, based on which the DD of the conveyor belt is estimated. From the experimental results, the following concluding remarks can be drawn:

- (1) The proposed method can detect the deviation at any position along the belt conveyor, and with the lens down, it overcomes the drawback that the lens is susceptible to contamination in the existing machine vision-based method. In addition, this method provides a new intelligent solution for the monitoring of the belt conveyor.
- (2) The accuracy of the proposed OM-SSD-2 on the self-built dataset is 0.9995, and the conveyor belt edge detection method and the idler outer edge detection algorithm can detect the edges efficiently.
- (3) The experimental results show that the proposed method can detect the deviation of the conveyor belt continuously with an RMSE of 3.7 mm, an MAE of 4.4 mm, and average time consumption of 135.5 ms, which is better than that of the existing method, and the preferred shooting height is 2.4 m.

In order to further improve the performance of the deviation detection method of belt conveyor, the following research can be studied in the future:

- (1) The self-built dataset should be expanded to enhance the adaptability of the proposed method.
- (2) To improve the accuracy of the edge detection and object detection, complementary information should be fused into the image, and the thermal infrared information may be the best choice in mechanical fault detection.
- (3) Due to the powerful nonlinear mapping ability of deep learning and the convenient access to multispectral data, the deviation detection method based on an end-to-end deep learning framework and multispectral information may be the next research direction, and the CNN-based image registration, multispectral information fusion, object detection, and deviation estimator should be investigated, improved, and integrated.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest regarding the publication of this paper.

Acknowledgments

This work was partially supported by the National Natural Science Foundation of China (grant no. 51274150), the Key Projects of Science and Technology Support of Tianjin, China (grant no. 18YFZCGX00930), and the Key R&D Projects Supported by Achievements Transformation Relay of Tianjin, China (grant no. 18YFJLCG00060).

References

- [1] J. Li and C. Miao, "The conveyor belt longitudinal tear on-line detection based on improved SSR algorithm," *Optik*, vol. 127, no. 19, pp. 7395–8202, 2016.
- [2] L. Zhao and Y. Lin, "Typical failure analysis and processing of belt conveyor," *Procedia Engineering*, vol. 26, pp. 942–946, 2011.
- [3] Y. Yang, C. Miao, X. Li, and X. Mei, "On-line conveyor belts inspection based on machine vision," *Optik*, vol. 125, no. 19, pp. 5803–5807, 2014.
- [4] X. Mei, C. Miao, Y. Yang, and X. Li, "Rapid inspection technique for conveyor belt deviation," *Journal of Mechanical Engineering Research and Developments*, vol. 39, no. 3, pp. 653–662, 2016.

- [5] H. Su, C. Yang, G. Ferrigno, and E. De Momi, “Improved human-robot collaborative control of redundant robot for teleoperated minimally invasive surgery,” *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 1447–1453, 2019.
- [6] Q. Zhou, S. Zhao, H. Li, R. Lu, and C. Wu, “Adaptive neural network tracking control for robotic manipulators with dead zone,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 12, pp. 3611–3620, 2019.
- [7] H. Su, Y. Hu, H. R. Karimi, A. Knoll, G. Ferrigno, and E. De Momi, “Improved recurrent neural network-based manipulator control with remote center of motion constraints: experimental results,” *Neural Networks*, vol. 131, pp. 291–299, 2020.
- [8] S. R. Saufi, Z. A. B. Ahmad, M. S. Leong, and M. H. Lim, “Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: a review,” *IEEE Access*, vol. 7, pp. 122644–122662, 2019.
- [9] W. Qi, H. Su, and A. Andrea, “A smartphone-based adaptive recognition and real-time monitoring system for human activities,” *IEEE Transactions on Human-Machine Systems*, vol. 2020, 2020.
- [10] H. Su, W. Qi, C. Yang, J. Sandoval, G. Ferrigno, and E. D. Momi, “Deep neural network approach in robot tool dynamics identification for bilateral teleoperation,” *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2943–2949, 2020.
- [11] J. Li, Y. Luo, L. Shi et al., “Automatic fetal brain extraction from 2D in utero fetal MRI slices using deep neural network,” *Neurocomputing*, vol. 378, pp. 335–349, 2020.
- [12] Y. Liu, Y. Wang, C. Zeng, W. Zhang, and J. Li, “Edge detection for conveyor belt based on the deep convolutional network,” 2018.
- [13] A. B. Alhassan, X. Zhang, H. Shen, and H. Xu, “Power transmission line inspection robots: a review, trends and challenges for future research,” *International Journal of Electrical Power & Energy Systems*, vol. 118, 2020.
- [14] S.-N. Yu, J.-H. Jang, and C.-S. Han, “Auto inspection system using a mobile robot for detecting concrete cracks in a tunnel,” *Automation in Construction*, vol. 16, no. 3, pp. 255–261, 2007.
- [15] W. Yang, X. Zhang, and H. Ma, “An inspection robot using infrared thermography for belt conveyor,” 2016.
- [16] S. Ren, K. He, R. Girshick, and J. Sun, “Faster R-CNN towards real-time object detection with region proposal networks,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137–1149, 2015.
- [17] R. Girshick, “Fast R-CNN,” in *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1440–1448, London, UK, 2015.
- [18] W. Liu, D. Anguelov, D. Erhan et al., “SSD: single shot MultiBox detector,” *Computer Vision-ECCV 2016*, vol. 9905, pp. 21–37, 2016.
- [19] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” 2016.
- [20] V. Osin, A. Cichocki, and E. Burnaev, “Fast multispectral deep fusion networks,” *Bulletin of the Polish Academy of Sciences-Technical Sciences*, vol. 66, no. 6, pp. 875–889, 2018.
- [21] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “Imagenet classification with deep convolutional neural networks,” *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.
- [22] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” 2014.
- [23] C. Szegedy, W. Liu, Y. Jia et al., “Going deeper with convolutions,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1–9, London, UK, 2015.
- [24] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015.
- [25] S. Han, J. Pool, J. Tran, and W. J. Dally, “Learning both weights and connections for efficient neural network,” 2015.
- [26] X. Zhang, X. Zhou, M. Lin, and J. Sun, “Shufflenet: An extremely efficient convolutional neural network for mobile devices,” 2018.
- [27] A. G. Howard, M. Zhu, B. Chen et al., “Mobileneets: efficient convolutional neural networks for mobile vision applications,” 2017.
- [28] H. Li, X. Zhang, and Z. Chen, “Automatic corner detection of checkerboard based on LSD,” *Optics and Precision Engineering*, vol. 23, no. 12, pp. 3480–3489, 2015.
- [29] J. Canny, “A computational approach to edge detection,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.
- [30] D. H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.
- [31] J. Cai, P. Huang, and B. Zhang, “Fast circle detector based on region-growing of gradient and histogram of euclidean distance,” *Acta Optica Sinica*, vol. 35, no. 3, pp. 193–202, 2015.
- [32] Y. Gao, T. Qiao, H. Zhang, Y. Yang, Y. Pang, and H. Wei, “A contactless measuring speed system of belt conveyor based on machine vision and machine learning,” *Measurement*, vol. 139, pp. 127–133, 2019.