

Article

Vision-Based Surface Inspection System for Bearing Rollers Using Convolutional Neural Networks

Shengping Wen ^{1,2}, Zhihong Chen ^{1,2,*} and Chaoxian Li ^{1,2}

¹ The Key Laboratory of Polymer Processing Engineering of Ministry of Education, School of Mechanical and Automotive Engineering, South China University of Technology, Wushan Road 381, Tianhe, Guangzhou 510640, China; shpwen@scut.edu.cn (S.W.); longtiao@foxmail.com (C.L.)

² National Engineering Research Center of Novel Equipment for Polymer Processing, South China University of Technology, Wushan Road 381, Tianhe, Guangzhou 510640, China

* Correspondence: jeremycz@foxmail.com

Received: 6 November 2018; Accepted: 7 December 2018; Published: 11 December 2018



Abstract: Bearings are commonly used machine elements and an important part of mechanical transmission. They are widely used in automobiles, airplanes, and various instruments and equipment. Bearing rollers are the most important components in a bearing and determine the performance, life, and stability of the bearing. In order to control the surface quality of the rollers, a machine vision system for bearing roller surface inspection is proposed. We briefly introduced the design of the machine vision system and then focused on the surface inspection algorithm. We proposed a multi-task convolutional neural network to detect defects. We extracted the features of the defects through a shared convolutional neural network, then classified the defects and calculated the position of the defects simultaneously. Finally, we determined if the bearing roller was qualified according to the position, category, and area of the defect. In addition, we explored various factors affecting performance and conducted a large number of experiments. We compared our method with the traditional methods and proved that our method had good stability and robustness.

Keywords: bearing roller; surface inspection; convolutional neural networks; machine vision

1. Introduction

Bearings are commonly used mechanical components. A bearing's main function is to support mechanical rotation and reduce the friction coefficient during its movement. Since the roller is the most important part of the bearing, its surface quality has a significant impact on the performance and even the life of the bearing, thus, the surface quality of the roller must be extremely high. Bearing rollers are shown in Figure 1 below.



Figure 1. Bearing rollers.

Rollers are the main pressure-bearing part in rolling bearings and are easily damaged due to defects and other factors. If there are defects on the roller surface, the stability of the bearing will be heavily reduced during use. Therefore, in mechanical design, the geometric accuracy and surface roughness of the roller are typically one level higher than that of the ferrules and raceways. Among the rolling bearings, deep groove ball bearings are mainly used in small- and medium-sized equipment, while roller bearings are widely used in medium- and large-sized machines. They are widely used in passenger transportation, aerospace, and other transportation fields, as well as agricultural machinery, industrial machinery, medical equipment, and other related machinery industries. The bearing roller is the main research object of this paper.

Defects inevitably occur on the surface of bearing rollers in the production process. The defects are mainly distributed on the cylindrical surface, chamfers, and end surfaces. Common defect categories include: Damage and scratches caused by mechanical collision; corrosion caused by mechanical aging; and material lacking, at the chamfer, and grind lacking caused in the production process. As can be seen from Figure 2 below.

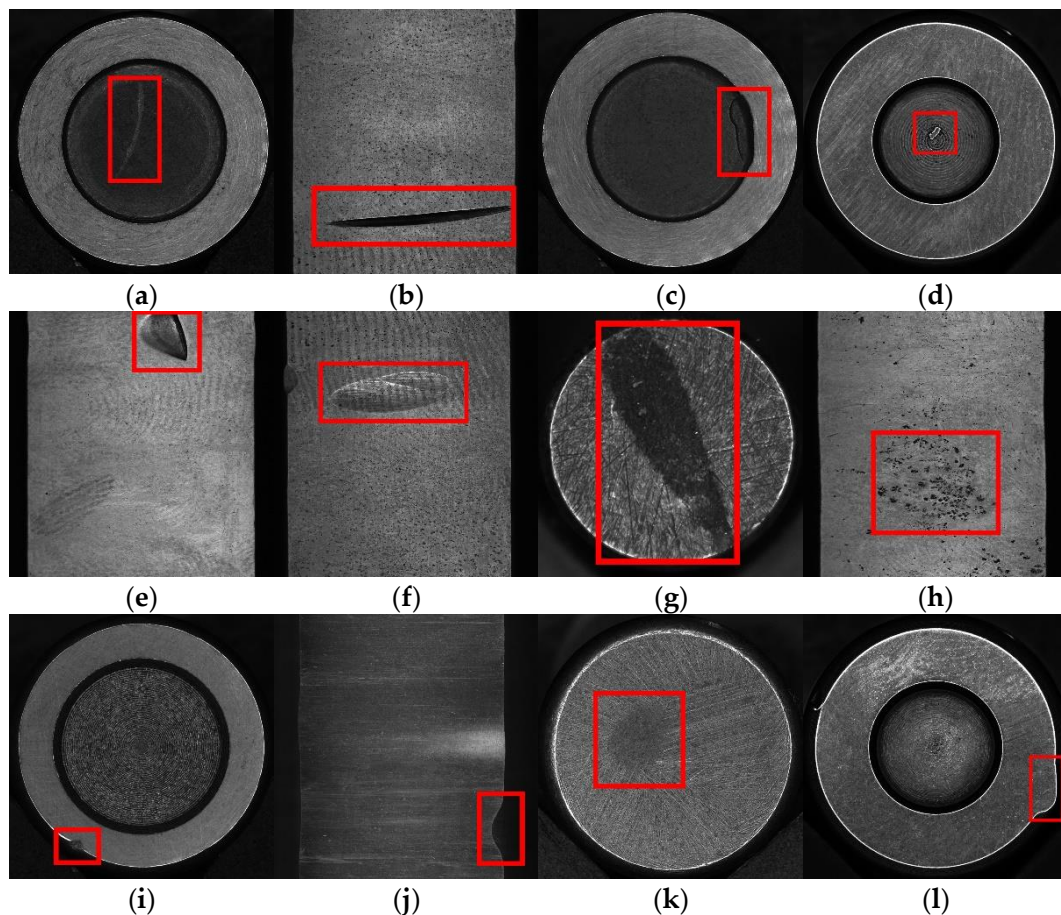


Figure 2. Common defects on the bearing roller: (a,b) Scratch; (c–f) damage; (g,h) corrosion; (i,j) material lacking at the chamfer; (k) grind lacking; and (l) stamp lacking.

The main defect categories are as follows:

1. Scratch, as shown in Figure 2a,b. A defect caused by a roller being scratched by other hard objects.
2. Damage, as shown in Figure 2c–f. We describe defects with large areas and irregular shapes as damage.
3. Corrosion, as shown in Figure 2g,h. The defects caused by corrosion.
4. Material lacking at the chamfer, as shown in Figure 2i,j. The roller is sunken at the chamfer, making the contour not a circle.

5. Grind lacking, as shown in Figure 2k. The defects caused by insufficient grinding.
6. Stamp lacking, as shown in Figure 2l. The defects caused by insufficient stamping.

Figure 2a,c,d,g,i,k,l are images of the end surfaces of the roller, and the rest are images of the cylindrical surface of the roller. These defects have a great influence on the performance and stability of the bearing and must be detected. Visual inspection is a good solution because it can reduce a lot of manual detection. At present, visual inspection technology has been used in many scenarios, such as chip pin and circuit solder inspection [1], workpiece vision measurement [2], plastic bottle defect detection [3], metal product surface defect detection [4–6], equipment parts identification and classification [7], gear and bearing surface inspection and measurement [8], bearing defect inspection [9], optical character recognition [10], and agricultural product identification [11]. Despite being used in large numbers, there are still many problems with visual inspection in the application of roller surface inspection. In the actual production process, it still relies more on manual inspection, and the inspection efficiency and level are relatively low.

Traditional methods used in manufacturing, such as edge detection [12,13], segmentation [14], and line detection [15,16], can hardly extract the internal structures and accurately classify each defect category. Generally, a defect is regarded as a target without distinction, and the difference in reflection between the target and the background is used to separate the two, and then judge whether the bearing roller is qualified according to the position and area of the target. Internal features of defects are not utilized at all. For this reason, it is easy to treat some textures, marks, oil stains, etc. as defects, resulting in a low accuracy and low recall rate of the detection process. Sometimes we need to know exactly how many defect categories exist and calculate the frequency of each defect category in order to properly adjust the production process. And this is not possible for the traditional surface inspection method that is used in manufacturing.

The appearance of deep learning makes up for the disadvantages of traditional algorithms. Since deep learning algorithms have shown state-of-the-art performance in classification and object detection tasks [17], deep neural networks can be utilized to learn the difference between different categories of defects, and to learn the commonality between the same category of defect, from a large amount of data, so that accurate classification can be achieved.

For example, Daniel Weimer et al. explored how convolutional neural network architecture and different hyper-parameter settings affect the feature extraction in industrial inspection [18]. Yiting Li et al. conducted research on the surface defect detection algorithm based on MobileNet-SSD, which proved that defect detection can be achieved using lightweight networks [19]. Xian Tao et al. designed a cascaded autoencoder architecture for segmenting and localizing defects [20], and showed that their method meets the robustness and accuracy requirements for metallic defect detection. Jinhua Lin et al. used a deep convolution neural network to detect defects on castings. They established a convolutional neural network to extract defect features from a suspicious area and, finally, the accuracy of detection was more than 96% [21]. S. Nahavand et al. used intelligent algorithms to detect defects on a metal surface [22]; Xian Tao et al. developed a machine vision device to detect defects on an electrical connector using convolutional neural networks, and they discussed the effects of data augmentation on defect recognition [23]; Yuan et al. used a modified segmentation method and deep neural networks to detect defects on the cover glass of mobile phones, and used GAN to generate new data in order to overcome the drawbacks presented when a huge amount of data is unavailable [24].

This paper introduces a real-time machine vision system for bearing roller surface inspection, which can classify and locate the major categories of defects occurring on the surface of a bearing roller, and determine whether each bearing roller is qualified based on the position, category, and area of the defect. In order to meet industrial requirements, we propose a multi-task convolutional neural network framework for classifying and locating defects simultaneously. The simplified pipeline lays the foundation for future industrial applications. The system can replace the manual inspection, and its performance is better than the traditional algorithms.

Compared with the existing surface inspection research that is based on deep learning, our method can achieve real-time performance because we use a multi-task learning strategy. The classification task is performed simultaneously with the localization task, making the process of the entire model simpler and more efficient. Our system is an entire surface inspection system for bearing roller defect detection and quality evaluation, which has industrial application value.

The rest of the paper is organized as follows: Section 2 introduces the design of the visual inspection system, including the hardware system and software system; Section 3 elaborates on the defect detection method based on the convolutional neural network; Section 4 gives the implementation and results of the experiment; and the Section 5 summarizes the whole paper.

2. System Overview

The visual inspection system mainly consisted of two parts: a hardware system and a software system.

The electrical part of the system was mainly composed of the PLC (Programmable Logic Controller) and the industrial computer. The PLC implements motion control and digital I/O control. The industrial computer mainly implements image acquisition, image processing, image analysis, and output. The hardware of the industrial computer was Intel Core i7-6700k CPU, NVIDIA GTX-1080 GPU, 128GB RAM, and the operating system was Windows 10. The mechanical structure is shown in Figure 3 below. It mainly consisted of a feeding device, a feeding conveyor, a pushing mechanism, four cameras, four ring light sources, a strip light source, a cam, a receiving device, etc.

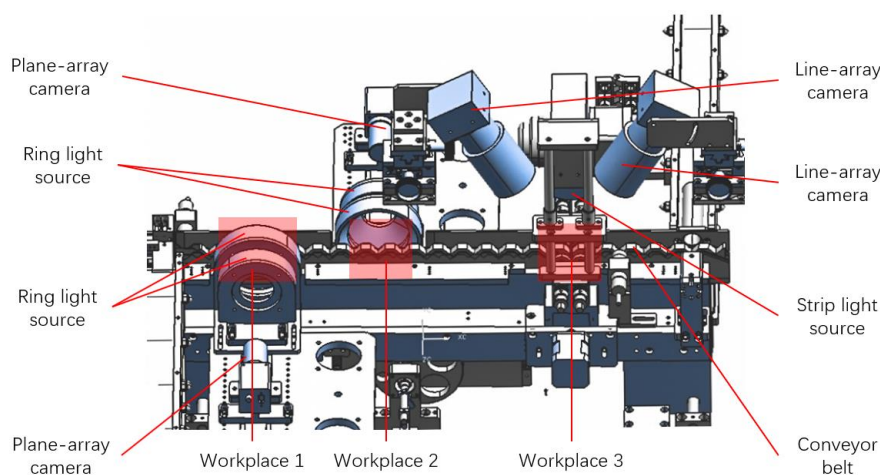


Figure 3. The surface inspection system.

The bearing roller has two end surfaces and a cylindrical surface, so three workplaces were required for image acquisition. The conveyor conveyed the rollers to workplace 1, workplace 2, and workplace 3 in sequence, and triggered the corresponding image acquisition function. At these three workplaces, we used a total of four industrial cameras. At workplace 1 and workplace 2, the roller was stationary. We use two plane-array cameras, with a resolution of 2448×2050 , to capture the two end surfaces of the roller. At workplace 3, the rollers began to roll under the action of the mechanism. We used two line-array cameras, with a resolution of 4K, to capture the cylindrical surface. As the cylindrical surface is the working surface of the bearing roller, we used two line-array cameras to prevent defects from being missed due to the rolling of the roller. The selection of the cameras was determined by the working distance and image definition requirements.

Visual inspection has strict requirements on illumination, and stable illumination can ensure the stability of the image quality. For defect features, it is important to choose a targeted light source. We set up two ring light sources, a high-angle light source, and a low-angle light source at workplace 1. The two ring light sources were arranged in front and rear. Since the end surface of the bearing roller

contains planes and chamfers, it is not possible to illuminate both parts with only one light source, so we used two light sources to simultaneously illuminate the chamfer and the plane of the roller. The low-angle light source in front was responsible for illuminating the chamfer, and the high-angle light source behind was responsible for illuminating the plane. The light source setting at workplace 2 was the same as at workplace 1. At workplace 3, we used a strip light source to illuminate the cylindrical surface.

The software system was programmed in C# and C++. C# writes the user interface, and C++ implements the underlying algorithm. The defect detection algorithm was developed using the PyTorch deep learning computing platform. Commonly used image processing algorithms, such as threshold segmentation and morphological processing, were implemented using OpenCV.

3. Surface Inspection Process

Bearing rollers have two end surfaces and a cylindrical surface. Since the cylindrical surface is the working surface of the bearing roller, a roller must be judged as unqualified if there are defects on it. If the defects occur on the outer circumference of the end surfaces, such as material lacking at the chamfer and stamp lacking, it will also affect the working surface, and the roller must also be judged as unqualified. For the defects inside the end surfaces, we can calculate the defect area to determine whether the roller is qualified.

Because the material lacking at the chamfer, represented by Figure 2i,j above, and the stamp lacking, represented by Figure 2l, can be first detected and excluded in the inspection process described below, our detection algorithm primarily detected and analyzed four categories of defects, which were damage, scratch, corrosion and grind lacking. Details of these defect categories are shown in Figure 4 below. Defects other than those mentioned above are not discussed because of their low frequency of occurrence.

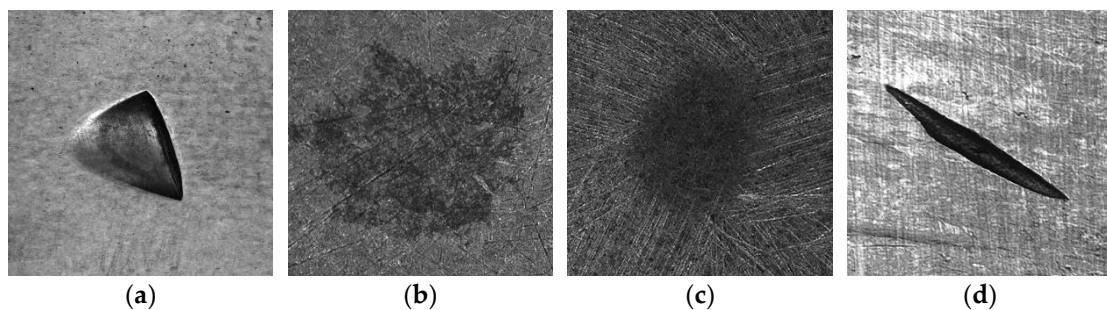


Figure 4. Details of common defect categories: (a) Damage; (b) corrosion; (c) grind lacking; and (d) scratch.

Image acquisition was performed at a suitable working distance. For each bearing roller, a total of two images were captured on both end surfaces, and the image was cropped to a resolution of 416×416 . For the cylindrical surface, of each bearing roller, two images were captured and the resolution was also 416×416 after cropping.

We note that, although the shapes of the same defect category are different, there are similarities in features that can be extracted and classified by convolutional neural networks. In this section, we will describe in detail the method for identifying various defects on bearing rollers. The completed process pipeline is shown in Figure 5 below.

The process consisted of the following three stages: First, contour detection. It is used to determine if the outer contour of the end surface is a standard circle and exclude the roller with a non-circular contour. Second, defect detection. It uses a multi-task learning convolutional neural network to classify and locate defects. Third, roller quality evaluation. It is used to determine whether the bearing roller is qualified according to the position, category, and area of the defect.

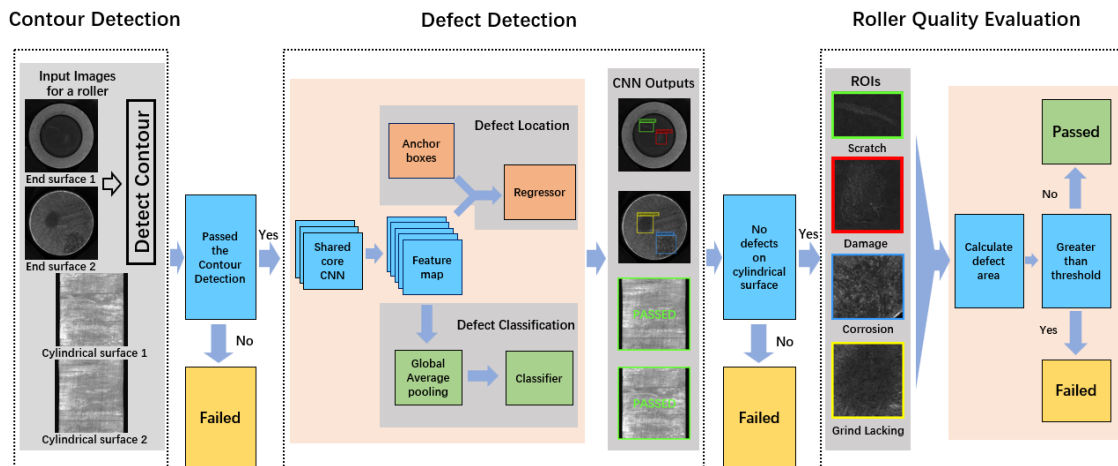


Figure 5. Surface inspection process. CNN: Convolutional Neural Networks; ROI: Region of Interest.

3.1. Contour Detection

In this part, we fitted the outer contour of the end surfaces of the roller by using the Hough transform [25]. The pipeline can be seen from Figure 6 below.

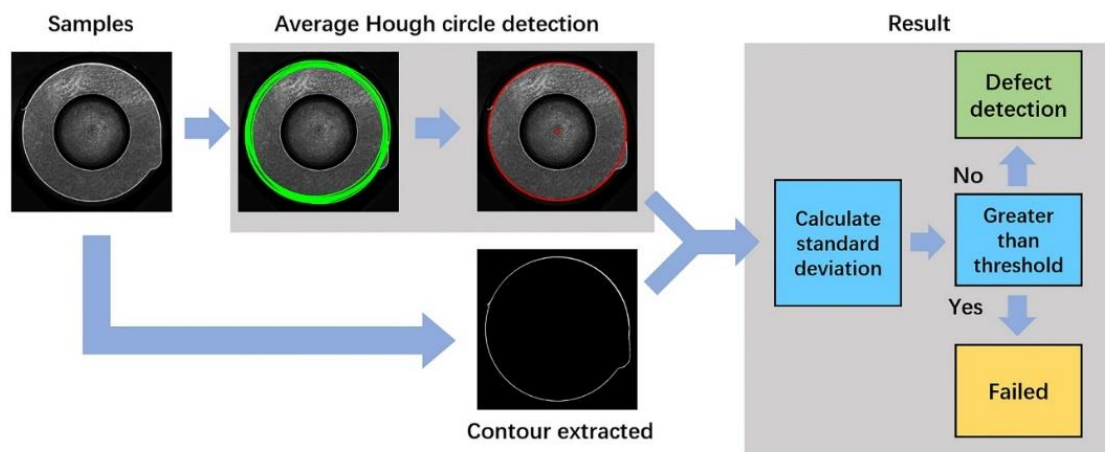


Figure 6. Contour detection process.

We performed the Hough circle detection 10 times for each end surface, and then took the average of the radius and the average of the center coordinates as the actual radius and center coordinates of the outer contour of the end surface. Then we used the Canny algorithm to extract the outer contour and calculated the standard deviation of the distance between the actual center coordinates and all points on the contour. The formulas were as follows:

$$\begin{aligned}
 x_s &= \sum_{i=1}^{10} x_{si} \\
 y_s &= \sum_{i=1}^{10} y_{si} \\
 r_s &= \sum_{i=1}^{10} r_{si} \\
 std &= \sqrt{\frac{1}{N} \sum_{i=1}^N (d_j - r_s)^2}
 \end{aligned} \tag{1}$$

where (x_{si}, y_{si}) and r_{si} are the center coordinates and the radius of the i -th circle detected by Hough circle detection. (x_s, y_s) and r_s are the actual center coordinates and the actual radius of the outer contour of

the end surface, d_j is the distance between the j -th point on the contour and the coordinate (x_s, y_s) , and std is the standard deviation of d .

If the std was less than the set threshold (set to 0.4 by experiment), it meant that the outer contour of the current end surface was a circle, and the sample would be sent into the shared convolutional neural network to extract a feature map for defect classification and localization. On the contrary, if there was a defect at the contour of the end surface, and the outer contour was not a circle, then the bearing roller would be judged as unqualified.

3.2. Features Extraction Using CNN

In this part, we designed a 26-layer convolutional neural network for feature extraction. The design reference for this network comes from the VGG [26] and the Resnet [27]. Firstly, we used small convolution kernels, instead of large convolution kernels, in order to reduce the computation and increase the network depth as well as the nonlinear mapping, so that the model's data-fitting ability would be stronger. Secondly, we also used the 1×1 convolution kernel to compress parameters that were output from the 3×3 convolution kernel to reduce the computation of the network. Finally, we referred to Resnet to add shortcuts to the network in order to alleviate the gradient disappearance during training. The structure is shown in Table 1 below. We pre-trained the network on the ImageNet dataset [28] to improve the generalization capabilities.

Table 1. Network structure.

Layer Type	Kernel Size/Stride	Output Size
Convolutional	$3 \times 3 \times 32$	416×416
Max Pooling	2×2	208×208
Convolutional Residual	$\begin{bmatrix} 3 \times 3 \times 64 \\ 3 \times 3 \times 64 \end{bmatrix} \times 2$	208×208
Max Pooling	2×2	104×104
Convolutional Residual	$\begin{bmatrix} 3 \times 3 \times 128 \\ 3 \times 3 \times 128 \end{bmatrix} \times 2$	104×104
Convolutional	$3 \times 3/2, 256$	52×52
Convolutional Residual	$\begin{bmatrix} 1 \times 1 \times 128 \\ 3 \times 3 \times 256 \end{bmatrix} \times 2$	52×52
Convolutional	$3 \times 3/2, 512$	26×26
Convolutional Residual	$\begin{bmatrix} 1 \times 1 \times 256 \\ 3 \times 3 \times 512 \end{bmatrix} \times 3$	26×26
Convolutional	$3 \times 3/2, 1024$	13×13
Convolutional Residual	$\begin{bmatrix} 1 \times 1 \times 512 \\ 3 \times 3 \times 1024 \end{bmatrix} \times 2$	13×13
Avgpool Softmax	Global, 4	

3.3. Defect Classification and Localization

We classified the defects and calculated the position of the defects based on the feature map extracted by the CNN. We used a multi-task CNN architecture to unify classification and localization in order to simplify the entire inspection process. The loss function of the entire CNN was linearly weighted by the loss function of the classification task and the loss function of the localization task, as shown below:

$$L_{total} = L_{cls} + \alpha L_{loc} \tag{2}$$

where L_{cls} is the loss function of the classification task, L_{loc} is the loss function of the localization task, and α is the weight of L_{loc} .

3.3.1. Classification

The feature map was extracted by the convolutional neural network, and the dimension of the feature map was $13 \times 13 \times 1024$. Each position of 13×13 represented a specific area in the original

image. We followed the Single Shot MultiBox Detector (SSD) [29] and Faster R-CNN [30] to associate 6 anchor boxes at each location of the feature map. Each anchor box was responsible for predicting whether there was a defect at the position or not. If there was a defect, it would then predict the defect category and calculate the probability of the defect belonging to a certain defect category. In this paper, there were four categories of defects. The loss function of the classification task was defined as follows:

$$L_{cls}(x, c) = -\frac{1}{N} \sum_{i \in Pos} x_{ij}^p \log(\hat{c}_i^p) - \frac{1}{N} \sum_{i \in Neg} \log(\hat{c}_i^0) \tag{3}$$

$$\hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

where N is the total number of anchor boxes, i refers to the anchor box index, j refers to the ground-truth box index, p refers to the category index, and 0 represents the background. $x_{ij}^p = 1$ when category p of i -th anchor box and category p of j -th ground-truth box match, otherwise $x_{ij}^p = 0$. c_i^p indicates the predicted probability of the category p corresponding to the i -th anchor box.

3.3.2. Localization

If there was a defect in the current position, we calculated the *IoU* of each anchor box with the ground-truth box, and removed the anchor boxes whose *IoU* was smaller than the set threshold by non-maximum suppression, leaving the anchor box whose *IoU* was larger than the set threshold. The boxes left were our predicted boxes. *IoU* was defined as:

$$IoU(G_T, P_B) = \frac{Area(G_T \cap P_B)}{Area(G_T \cup P_B)} \tag{4}$$

where G_T is the ground-truth box and P_B is the predicted box.

Each predicted box contained four predicted values, which were the center coordinates (x, y) of the box, and the length and width of the box. Through continuous iteration, the loss was gradually reduced, and the position of the predicted box was constantly approaching the ground-truth box. The loss function was as follows:

$$L_{loc}(x, t, t^*) = \frac{1}{N} \sum_{i \in Pos} x_{ij}^p L_{reg}(t_i, t_i^*) \tag{5}$$

where L_{reg} is Smooth L1 loss, N is the total number of anchor boxes, and $x_{ij}^p = 1$ when category p of i -th anchor box and category p of j -th ground-truth box match, otherwise $x_{ij}^p = 0$. t_i is a four-dimensional vector that represents the position of the predicted box. t_i^* is a four-dimensional vector that represents the position of the ground-truth box.

$$\begin{aligned} t_x &= (x - x_a) / w_a, & t_y &= (y - y_a) / h_a, \\ t_w &= \log(w / w_a), & t_h &= (h / h_a), \\ t_x^* &= (x^* - x_a) / w_a, & t_y^* &= (y^* - y_a) / h_a, \\ t_w^* &= \log(w^* / w_a), & t_h^* &= (h^* / h_a), \end{aligned} \tag{6}$$

where x, y , denote the box's center coordinates and w, h , denote its width and height, respectively. Variables x, x_a , and x^* are for the predicted box, anchor box, and ground-truth box, respectively (likewise for y, w , and h).

3.4. Roller Quality Evaluation

For defects that occurred on the cylindrical surface, no matter which kind of defect it was and what the defect area was, the bearing roller was judged as unqualified. For defects that occurred on the end surfaces, step 3.1, described above, had already excluded defects, such as material lacking at

the chamfer and stamp lacking, that caused the outer contour to not be circular in shape. For corrosion, scratch, damage, and grind lacking defects, the bearing roller was judged based on the defect area. The defects with bounding boxes were equivalent to the ROIs (Region of Interest), and the ROIs were analyzed separately using the image processing method. Accordingly, we calculated the defect area on each end surface separately. The process is shown in Figure 7 below.

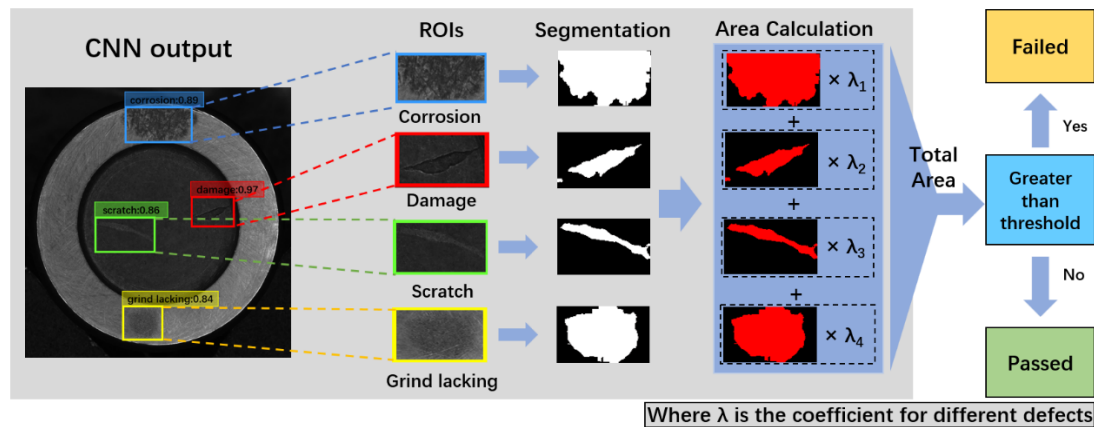


Figure 7. Roller quality determination process.

Different defects have different impacts on the performance of the roller. Damage has the greatest impact on the performance, followed by scratch, corrosion, and grind lacking. Our surface inspection system had different tolerances for different defects and; therefore, we defined four coefficients for the four defects. When calculating the total defect area, it was necessary to multiply the area of the different defects by the corresponding coefficient. For damage, scratch, corrosion, and grind lacking, the coefficients were defined as 3, 1.5, 1, and 0.8, respectively. The coefficients were defined by multiple experiments based on the inspection effect, and different coefficients could be defined according to different situations.

After performing median filtering, Otsu thresholding [31], and morphological processing on the ROIs, defects were segmented from the background, and then we calculated the total area of all the defects. If the total defect area was greater than the set threshold, which was about 5% of the end surface area, the bearing roller would be judged as unqualified. The roller would be judged as qualified only when the defect area of each end surface was smaller than the threshold.

3.5. Data Augmentation

Both classification and localization depend on the CNN model, and the deep CNN model is easily over-fitting due to its powerful fitting ability, especially when the amount of data is not large. For bearing rollers, the probability of occurrence of defects is relatively low, and the amount of data that can be collected is relatively small, so it is necessary to appropriately augment the original data. We adopted several commonly used augmentation methods, including image rotation, image flipping, image cropping, adding blur, and adding noise. The augmentation results are shown in Figure 8 below.

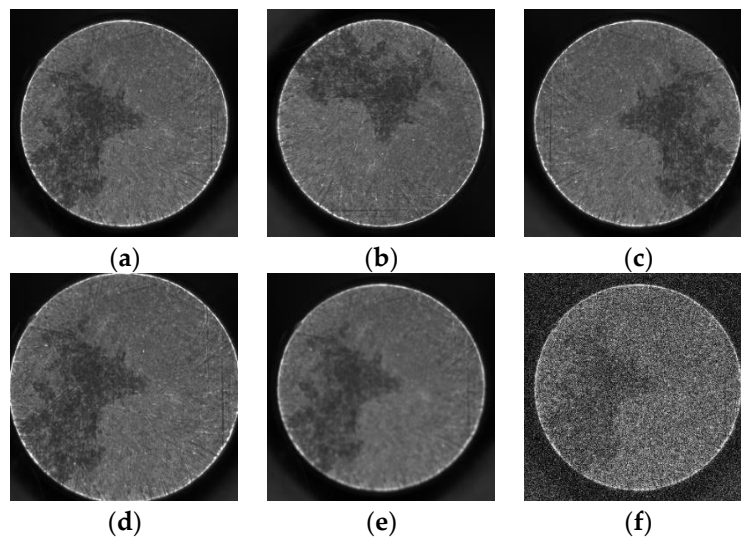


Figure 8. Data augmentation: (a) Original image; (b) rotation; (c) image flipping; (d) center cropping; (e) adding blur, and (f) adding gaussian noise.

4. Experiment

4.1. Experimental Configuration

4.1.1. Dataset Description

Our dataset was collected from the bearing rollers with different sizes on the production line. There were 3200 images in the dataset. There were one or more defects on each sample. The specific quantities are shown in Table 2 below. The images were down sampled to match the input size of 416×416 . We shuffled the data and then divided the data into three parts: 70%, as the training set; 15%, as the validation set; and 15%, as the test set. We made sure that all three parts of the dataset had the same data distribution by way of shuffling. The training set was used for model training, the validation set was used for selecting the model hyper-parameters, and the test set was used for evaluating the model performance. The training set, validation set, and the test set were strictly labeled manually.

Table 2. Defect data statistics.

Defect Categories	Training Set	Validation Set	Test Set	Total
Damage	692	145	149	986
Grind lacking	492	108	107	707
Corrosion	630	138	136	904
Scratches	620	132	137	889

4.1.2. Evaluation Indicators

In the following experiments, we quantitatively evaluated the performance of the defect detection algorithm and the performance of the entire surface inspection system. For the defect detection algorithm, we used mAP (Mean Average Precision) to evaluate its performance, and we used detection time to evaluate the efficiency of the algorithm. We also compared the multi-class classification performance of our algorithm and the pattern recognition algorithms, and we used the micro F1 score to evaluate the performance of the different methods. For the entire surface inspection system, we used the F1 score to evaluate its performance. The formulas for calculating the F1 score were as follows:

$$\begin{aligned}
 precision &= \frac{TP}{TP+FP} \\
 recall &= \frac{TP}{TP+FN} \\
 F1_score &= \frac{2 \times precision \times recall}{precision+recall}
 \end{aligned}
 \tag{7}$$

where TP represents the number of positive samples that are judged to be positive samples, FP represents the number of negative samples that are judged to be positive samples, and FN represents the number of positive samples that are judged to be negative samples. The formulas for calculating the micro F1 score were as follows:

$$\begin{aligned}
 Micro_P &= \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FP_i} \\
 Micro_R &= \frac{\sum_{i=1}^n TP_i}{\sum_{i=1}^n TP_i + \sum_{i=1}^n FN_i} \\
 Micro_F1_score &= \frac{2 \times Micro_P \times Micro_R}{Micro_P + Micro_R}
 \end{aligned}
 \tag{8}$$

where i represents the i -th category of defect, TP represents the number of positive samples that are judged to be positive samples, FP represents the number of negative samples that are judged to be positive samples, and FN represents the number of positive samples that are judged to be negative samples.

4.2. Performance of the Defect Detection Algorithm under Different Settings

The defect detection results are shown in Figure 9 below. The red box belongs to damage, the green box belongs to scratch, the yellow box belongs to grind lacking, and the blue box belongs to corrosion. The category of the defect and the probability are displayed above the box.

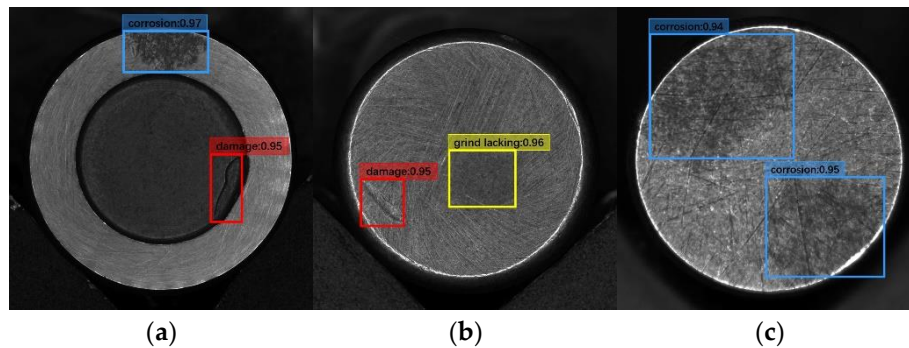


Figure 9. Cont.

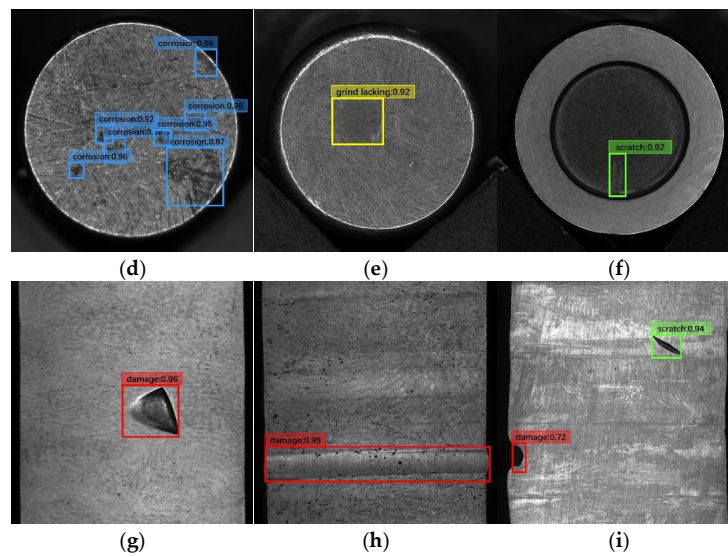


Figure 9. Defect detection results: (a) Corrosion and damage; (b) grind lacking and damage; (c,d) corrosion; (e) grind lacking; (f) scratch; (g,h) damage; (i) damage and scratch. The red box belongs to damage, the green box belongs to scratch, the yellow box belongs to grind lacking, and the blue box belongs to corrosion.

4.2.1. Influence of Different α on Performance

We used cross-validation to select the appropriate α . Table 3 gives the results of the task under different α . It can be seen from the table that the best score was achieved when $\alpha = 1.05$.

Table 3. Influence of α on performance.

α	0.8	0.85	0.9	0.95	1	1.05	1.1	1.15	1.2
mAP (%)	80.65	81.21	82.02	83.19	83.68	84.24	83.25	82.42	81.97

The AP (Average Precision) of each category when $\alpha = 1.05$ is shown in Table 4:

Table 4. AP of each defect category.

Defect Categories	Damage	Corrosion	Grind Lacking	Scratch
AP (%)	82.85	84.18	85.07	84.86

The detection results of different α are shown in Figure 10 below. The yellow boxes represent the ground-truth label. The Figure only shows the detection results when $\alpha = 0.8$, $\alpha = 0.9$, $\alpha = 1.05$, and $\alpha = 1.2$. It can be seen from the Figure that when $\alpha = 0.8$, $\alpha = 0.9$, and $\alpha = 1.2$, the detection results deviated from the ground-truth label, especially when $\alpha = 0.8$, and the result was more accurate when $\alpha = 1.05$.

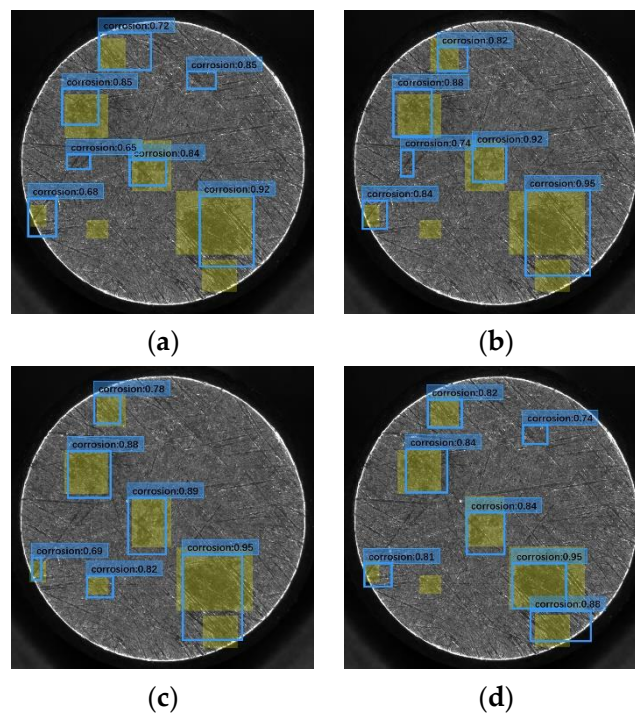


Figure 10. Detection results under different α : (a) $\alpha = 0.8$; (b) $\alpha = 0.9$; (c) $\alpha = 1.05$; and (d) $\alpha = 1.2$.

4.2.2. Influence of Data Augmentation on Performance

We used a variety of data augmentation strategies and ended up using the following methods to get the best results:

Each sample had a 20% chance of performing a specified angular rotation (60° , 120° , 180° , 240° , and 300°), with a 50% chance of flipping, a 5% chance of adding gaussian noise, a 5% chance to add blur, and a 30% chance of performing center cropping. The results are shown in Table 5 below.

Table 5. Influence of data augmentation on performance.

Augmentation	mAP (%)
No augmentation	74.41
Rotation + flipping + center cropping	82.07
Rotation + flipping + center cropping + add noise	83.62
Rotation + flipping + center cropping + blur	83.44
Rotation + flipping + center cropping + add noise + blur	84.18

When using the best data augmentation method, the APs for each defect category are shown in Table 6 below.

Table 6. AP of each defect category.

Defect Categories	Damage	Corrosion	Grind Lacking	Scratch
AP (%)	81.92	84.23	84.75	85.02

4.2.3. Influence of Different Resolutions on Performance

We compared the influence of different resolutions on the detection performance. The results are shown in Table 7 below.

Table 7. Influence of different resolutions on performance.

Resolution	mAP(%)	Detection Time
288 × 288	74.06	13.9 ms
320 × 320	77.25	16.9 ms
352 × 352	79.79	21.7 ms
384 × 384	82.14	25.6 ms
416 × 416	84.89	29.4 ms

It can be seen from the results that increasing the resolution had a significant impact on the mAP and detection time. As the resolution increased, the mAP increased but the detection time decreased. That was because the increase in resolution lead to an increase in computation. Therefore, it is necessary to select an appropriate resolution according to actual needs.

4.2.4. Influence of Model Pre-Training on Performance

Inspired by transfer learning [32–35], we pre-trained our CNN model on the ImageNet data set and compared the same model without pre-training. The results are shown in Table 8 below.

Table 8. Influence of model pre-training on performance.

	Pre-Trained	Not Pre-Trained
mAP (%)	84.42	76.78

It can be concluded from the results that the pre-trained model had a better generalization ability and had a positive effect on improving the mAP.

4.2.5. Influence of Different Base Networks on Performance

We compared our network with Resnet-50, VGG-19, and MobileNet [36]. The results are shown in Table 9 below, and the detection results are shown in Figure 10 below.

Table 9. Influence of different base networks on performance.

Base networks	Resnet-50	VGG-19	MobileNet	Our CNN Model
mAP (%)	85.65	83.86	78.45	84.19
Detection Time	83.3 ms	142.9 ms	11.2 ms	28.6 ms

As can be seen from the table, the best mAP was achieved using Resnet-50, but processing an image was more time consuming. VGG-19 achieved a mAP of 83.86% but took even longer to process a single image. MobileNet had a fairly high processing efficiency, but the mAP was the lowest among all the base networks. Our network achieved a better balance between the mAP and data processing efficiency due to less parameters and computation. Our mAP was close to that of using Resnet-50, and the detection time had a great advantage compared with Resnet-50 and VGG-19.

It can be seen from Figure 11 below that the detection results using MobileNet deviated from the ground-truth label the most. Using our CNN model, the Resnet-50, or the VGG-19 as the feature extractor was more accurate.

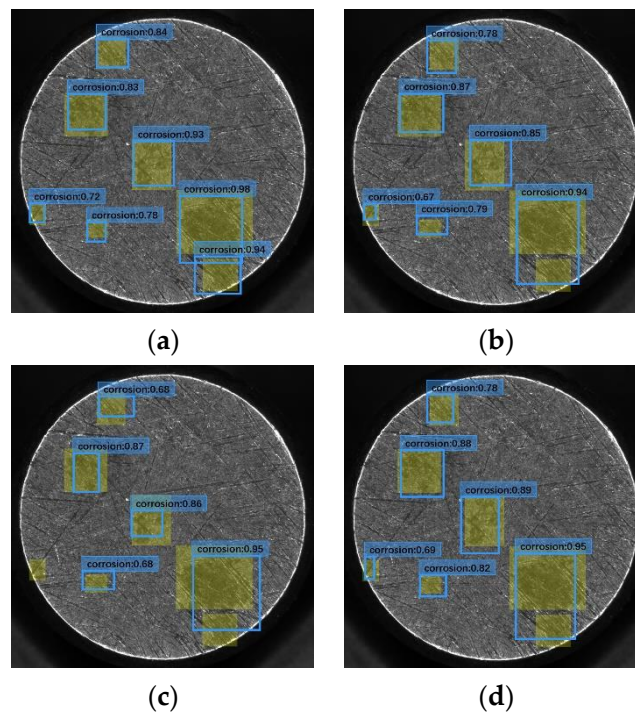


Figure 11. Detection results under different base networks: (a) Resnet-50; (b) VGG-19; (c) MobileNet; and (d) our CNN model.

4.2.6. Influence of Different Factors on Performance

We summarized all the influencing factors, as shown in Table 10 below. We got the best results when using more image augmentation, higher resolution, and the pre-trained model.

Table 10. Summary of influencing factors.

	Defect Detection Network						
More augmentation	✓			✓		✓	✓
High resolution		✓		✓	✓		✓
Pre-trained network			✓		✓	✓	✓
mAP(%)	67.18	69.45	65.84	76.17	74.08	73.69	84.49

4.3. Comparison between Pattern Recognition Methods and Our Method

To evaluate the performance of the classification module of our method, we compared the accuracy of the defect classification between our method and traditional methods whose codes are publicly available. (1) GLCM (Grey-Level Co-Occurrence Matrices) [37]: The GLCM feature refers to a common method of describing texture features by studying the spatial correlation properties of grayscale, and the texture features are a combination of energy, contrast, entropy, and correlation. (2) HOG (Histogram of Oriented Gradients) [38]: The HOG feature is a feature descriptor used for object detection in image processing. The algorithm first divided the image into small connected regions, which we call cell units. Then we collected the gradient or edge direction of each pixel in the cell unit to get a histogram. Finally, these histograms were combined to form a feature descriptor.

After obtaining the features described above, we used the SVM (Support Vector Machine) and the MLP (Multi-layer Perceptron) to classify the features. The MLP consisted of a two-layer neural network, a hidden layer, and an output layer. The hidden layer had 15 hidden units and the output

layer had 4 output units. We evaluated the performance of the defect classifier quantitatively using the micro F1 score. The micro F1 score was introduced in Section 4.1.2.

The results are shown in the Table 11. It can be seen from the Table that the traditional method could only achieve a micro F1 score of about 70, whereas our method achieved a score of over 90 in the classification task. That was because we used deep convolutional neural networks to learn the internal features of the defects, which had a positive impact on the classification task.

Table 11. Performance of classification using different methods.

Method	Micro F1 Score
GLCM + MLP	75.53
GLCM + SVM	70.83
HOG + MLP	72.29
HOG + SVM	69.44
Our method	90.97

4.4. Performance of the Surface Inspection System

As the detection of a defect does not mean that a bearing roller fails, it is necessary to determine whether the roller is qualified according to the category, position, and area of the defect. In the following experiments, we inspected three different sized bearing rollers. We used the F1 score to evaluate the performance of the entire bearing roller surface inspection system. We obtained 1800 bearing rollers from the production line by manual screening, 600 for each size, including 300 qualified products (positive) and 300 unqualified products (negative). Then we used our surface inspection system to test these bearing rollers, and checked the precision and recall rate to calculate the F1 score. The F1 score was introduced in Section 4.1.2.

To evaluate the actual performance of our surface inspection system, we compared our approach to the traditional method currently being used in the production line. The traditional method captured the images and adjusted the resolution to 500×500 , then it performed median filtering and divided the ROIs on the image, and then it performed threshold segmentation [39] and morphological processing in the ROIs to segment the defects. After the segmentation, defects were separated from the background. Finally, the traditional method determined whether the bearing roller was qualified by calculating whether the area of the defect exceeded the set threshold. The results of the comparison experiment are shown in Table 12 below.

Table 12. Comparison between the traditional method and our method.

Method	Traditional Method			Our Method		
	10	12	15	10	12	15
Diameter (mm)	10	12	15	10	12	15
Precision (%)	86.70	86.94	86.40	92.81	92.67	92.59
Recall (%)	81.38	81.56	80.85	90.64	90.21	90.08
F1 score	83.96	84.16	83.53	91.71	91.42	91.32
Detection time	1.98s	2.07s	2.16s	2.00s	2.08s	2.16s

It can be seen from the results that the accuracy and recall rate of the traditional method, which was currently being used in the manufacturing, were lower than our method; the recall rate especially was very low. The main reason for this is that traditional methods can easily misjudge some non-defects (e.g., textures, oil stains, marks, etc.) as defects, so that some qualified products will be misjudged as unqualified, resulting in a low precision and a low recall rate. The recall rate and accuracy of our method were relatively higher because our method classifies defects well.

5. Conclusions

In this paper, we proposed a machine vision system for bearing roller surface inspection. In order to control the quality of the product, a multi-task convolutional neural network was designed to detect the defects. The features of the defects were extracted through the shared convolutional neural network, and then the defects were classified and the position of the defects were calculated simultaneously. Finally, we determined if the bearing roller was qualified base on the position, category, and area of the defects. We conducted a large number of experiments, and compared our method with the traditional surface inspection methods used in manufacturing. The quantitative experimental results showed that our method was superior in accuracy and robustness, and meet the requirements of industrial manufacturing.

The limitation of our proposed approach is that deep learning requires a large amount of labeled data and depends on the performance of the hardware. In the future, we will continue to optimize the algorithm and network structure to reduce the computational cost and, thus, allow them to be truly widely used in industrial manufacturing. And we will try to use semi-supervised learning or GAN (Generative Adversarial Networks) to generate new data to solve the problem of insufficient data.

Author Contributions: Project administration, S.W.; validation, Z.C.; investigation, Z.C. and C.L.; and resources, C.L.

Funding: This research received no external funding.

Acknowledgments: This work was supported by the National Key Research and Development Program of China (Grant No. 2016YFB0302300), the Key Program of National Natural Science Foundation of China (Grant No. 51435005), the National Natural Science Foundation of China (Grant No.51505153), and the Science and Technology Program of Guangzhou, China (Grant No.201607010240).

Conflicts of Interest: The authors declare no conflicts of interest.

References

- Oyeleye, O.; Lehtihet, E.A. A classification algorithm and optimal feature selection methodology for automated solder joint defect inspection. *J. Manuf. Syst.* **1998**, *17*, 251–262. [[CrossRef](#)]
- Chen, S.; Zhou, T.; Zhang, X.D.; Sun, C.K. Monocular vision measurement system of the position and attitude of the object. *Chin. J. Sens. Actuators* **2007**, *20*, 2011–2015.
- Raafat, H.; Taboun, S. An integrated robotic and machine vision system for surface flaw detection and classification. *Comput. Ind. Eng.* **1996**, *30*, 27–40. [[CrossRef](#)]
- Liou, F.; Barua, S.; Newkirk, J.; Sparks, T. Vision-based defect detection in laser metal deposition process. *Rapid Prototyp. J. Vol.* **2013**, *20*, 77–85.
- Do, Y.; Lee, S.; Kim, Y. Vision-based surface defect inspection of metal balls. *Meas. Sci. Technol.* **2011**, *22*. [[CrossRef](#)]
- Zhang, X. Vision inspection of metal surface defects based on infrared imaging. *Acta Opt. Sin.* **2011**, *31*, 0312004. [[CrossRef](#)]
- Hou, T.H.; Pern, M.D. A computer vision-based shape-classification system using image projection and a neural network. *Int. J. Adv. Manuf. Technol.* **1999**, *15*, 843–850. [[CrossRef](#)]
- Shiau, Y.R.; Jiang, B.C. Study of a measurement algorithm and the measurement loss in machine vision metrology. *J. Manuf. Syst.* **1999**, *18*, 22–34. [[CrossRef](#)]
- Shen, H.; Li, S.; Gu, D.; Chang, H. Bearing defect inspection based on machine vision. *Measurement* **2012**, *45*, 719–733. [[CrossRef](#)]
- Impedovo, S.; Ottaviano, L.; Occhinegro, S. Optical character recognition—A survey. *Int. J. Pattern Recognit. Artif. Intell.* **1991**, *5*, 1–24. [[CrossRef](#)]
- Wei, X.; Jia, K.; Lan, J.; Li, Y.; Zeng, Y.; Wang, C. Automatic method of fruit object extraction under complex agricultural background for vision system of fruit picking robot. *Opt. Int. J. Light Electron. Opt.* **2014**, *125*, 5684–5689. [[CrossRef](#)]
- Canny, J. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **1986**, *PAMI-8*, 679–698. [[CrossRef](#)]

13. Perona, P.; Malik, J. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *12*, 629–639. [[CrossRef](#)]
14. Felzenszwalb, P.F.; Huttenlocher, D.P. Efficient graph-based image segmentation. *Int. J. Comput. Vis.* **2004**, *59*, 167–181. [[CrossRef](#)]
15. Von Gioi, R.G.; Jakubowicz, J.; Morel, J.-M.; Randall, G. Lsd: A fast line segment detector with a false detection control. *IEEE Trans. Pattern Anal. Mach. Intell.* **2010**, *32*, 722–732. [[CrossRef](#)] [[PubMed](#)]
16. Topal, C.; Akinlar, C. Edge drawing: a combined real-time edge and segment detector. *J. Vis. Commun. Image Represent.* **2012**, *23*, 862–872. [[CrossRef](#)]
17. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [[CrossRef](#)] [[PubMed](#)]
18. Weimer, D.; Scholz-Reiter, B.; Shpitalni, M. Design of deep convolutional neural network architectures for automated feature extraction in industrial inspection. *CIRP Ann.* **2016**, *65*, 417–420. [[CrossRef](#)]
19. Li, Y.; Huang, H.; Xie, Q.; Yao, L.; Chen, Q. Research on a Surface Defect Detection Algorithm Based on MobileNet-SSD. *Appl. Sci.* **2018**, *8*, 1678. [[CrossRef](#)]
20. Tao, X.; Zhang, D.; Ma, W.; Liu, X.; Xu, D. Automatic Metallic Surface Defect Detection and Recognition with Convolutional Neural Networks. *Appl. Sci.* **2018**, *8*, 1575. [[CrossRef](#)]
21. Lin, J.; Yao, Y.; Ma, L.; Wang, Y. Detection of a casting defect tracked by deep convolution neural network. *Int. J. Adv. Manuf. Technol.* **2018**, *97*, 573–581. [[CrossRef](#)]
22. Zheng, H.; Kong, L.X.; Nahavandi, S. Automatic inspection of metallic surface defects using genetic algorithms. *J. Mater. Process. Technol.* **2002**, *125*, 427–433. [[CrossRef](#)]
23. Tao, X.; Wang, Z.; Zhang, Z.; Zhang, D.; Xu, D.; Gong, X.; Zhang, L. Wire defect recognition of spring-wire socket using multitask convolutional neural networks. *IEEE Trans. Compon. Packag. Manuf. Technol.* **2018**, *8*, 689–698. [[CrossRef](#)]
24. Yuan, Z.C.; Zhang, Z.T.; Su, H.; Zhang, L.; Shen, F.; Zhang, F. Vision-based defect detection for mobile phone cover glass using deep neural networks. *Int. J. Precis. Eng. Manuf.* **2018**, *19*, 801–810. [[CrossRef](#)]
25. Hough, B.P. Methods and Means for Recognizing Complex Pattern. U.S. Patent No. 3,069,654, 18 December 1962.
26. Karen, S.; Andrew, Z. Very deep convolutional networks for large-scale image recognition. In Proceedings of the International Conference on Representation Learning (ICRL 2015), San Diego, CA, USA, 7–9 May 2015.
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016.
28. Deng, J.; Dong, W.; Socher, R.; Li, L.J. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Miami, FL, USA, 20–25 June 2009; pp. 248–255.
29. 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, 8–16 October 2016; Springer International Publishing: Cham, Switzerland, 2016; pp. 21–37.
30. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. In Proceedings of the International Conference on Neural Information Processing Systems, Montreal, Canada, 7–12 December 2015; MIT Press: Cambridge, MA, USA, 2015; Volume 39, pp. 91–99.
31. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [[CrossRef](#)]
32. Lim, J.J.; Salakhutdinov, R.; Torralba, A. Transfer learning by borrowing examples for multiclass object detection. In Proceedings of the International Conference on Neural Information Processing Systems, Granada, Spain, 12–17 December 2011; Curran Associates Inc.: Vancouver, BC, Canada, 2011; pp. 118–126.
33. Huh, M.; Agrawal, P.; Efros, A.A. What makes ImageNet good for transfer learning? *arXiv*, **2016**, arXiv:1608.08614.
34. Zhuang, F.Z.; Ping, L.; Qing, H.E.; Shi, Z.Z. Survey on transfer learning research. *J. Softw.* **2015**, *26*, 26–39.
35. George, D.; Shen, H.; Huerta, E.A. Deep Transfer Learning: A new deep learning glitch classification method for advanced LIGO. *arXiv*, **2017**, arXiv:1706.07446.
36. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. *arXiv*, **2017**, arXiv:1704.04861.
37. Chondronasios, A.; Popov, I.; Jordanov, I. Feature selection for surface defect classification of extruded aluminum profiles. *Int J. Adv. Manuf. Technol.* **2016**, *83*, 33–41. [[CrossRef](#)]

38. Shumin, D.; Zhoufeng, L.; Chunlei, L. Adaboost learning for fabric defect detection based on hog and SVM. In Proceedings of the 2011 International Conference on Multimedia Technology (ICMT), Hangzhou, China, 26–28 July 2011; pp. 2903–2906.
39. Ng, H.-F. Automatic thresholding for defect detection. *Pattern Recognit. Lett.* **2004**, *27*, 1644–1649. [[CrossRef](#)]



© 2018 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 (<http://creativecommons.org/licenses/by/4.0/>).