applied sciences

MDPI

Article

Real-Time Recognition Method for 0.8 cm Darning Needles and KR22 Bearings Based on Convolution Neural Networks and Data Increase

Jing Yang [1,2], Shaobo Li [1,3,*], Zong Gao [3], Zheng Wang [1] and Wei Liu [2,4]

1 School of Mechanical Engineering, Guizhou University, Guiyang 550025, China; yang_jing0903@163.com (J.Y.); zhengwang0216@123.com (Z.W.)
2 School of Electrical and Computer Engineering, Oklahoma State University, Stillwater, OK 74078, USA; wei.liu@okstate.edu
3 Guizhou Provincial Key Laboratory of Public Big Data (Guizhou University), Guiyang, Guizhou 550025, China; zong209@163.com
4 School of Electronics and Information, Xi'an Polytechnic University, Xi'an 710048, China
* Correspondence: lishaobo@gzu.edu.cn; Tel.: +86-139-8505-3753

Received: 11 September 2018; Accepted: 5 October 2018; Published: 9 October 2018

Abstract: The complexity of the background and the similarities between different types of precision parts, especially in the high-speed movement of conveyor belts in complex industrial scenes, pose immense challenges to the object recognition of precision parts due to diversity in illumination. This study presents a real-time object recognition method for 0.8 cm darning needles and KR22 bearing machine parts under a complex industrial background. First, we propose an image data increase algorithm based on directional flip, and we establish two types of dataset, namely, real data and increased data. We focus on increasing recognition accuracy and reducing computation time, and we design a multilayer feature fusion network to obtain feature information. Subsequently, we propose an accurate method for classifying precision parts on the basis of non-maximal suppression, and then form an improved You Only Look Once (YOLO) V3 network. We implement this method and compare it with models in our real-time industrial object detection experimental platform. Finally, experiments on real and increased datasets show that the proposed method outperforms the YOLO V3 algorithm in terms of recognition accuracy and robustness.

Keywords: data increase; object recognition; precision parts; 0.8cm darning needle; KR22 bearing

1. Introduction

Computer vision has been used extensively in the industrial field. The use of computer vision in the flexible manufacturing system (FMS) can effectively realize the simultaneous processing of different product parts, and produce flexible and intelligent FMS production management and scheduling [1]. We try to use computer vision technology to analyze the characteristics of machining parts to achieve the accurate identification of precision parts, and then control and manage the machining process.

Research on the identification of mechanical products in complex industrial backgrounds has made some progress in recent years. Liu et al. [2] proposed a back propagation neural network method based on fusion edge detection to identify precision parts. Jiang et al. [3] proposed a method to identify precision parts by feature classification and merging symmetry; however, the method of using face as a symmetric identification unit is inefficient. Inic et al. [4] proposed a vehicle part shape detection method based on milling simulation. However, this method contains image segmentation, edge extraction, edge refinement, shape description, and other operations. Therefore, the algorithm to extract shape features is complex, computationally intensive, and completely discards the color

information of the image. To further improve the recognition accuracy for precision parts, He et al. [5] designed an improved differential box-counting method to calculate the fractal dimension of part images and guide robots to grab parts. Wan et al. [6] analyzed the hierarchy theories of classification modeling based on the design structure matrix (DSM). Although the algorithm is simple and easy to implement, DSM only describes the overall statistical characteristics of the image of precision parts and does not effectively represent the spatial information of the image. Two images with different visual perceptions are prone to error recognition.

Deep convolutional neural networks (CNNs) [7,8] have recently shown remarkable advantages in object detection, which is realized with two main methods. The first method is based on region generation. In this method, a number of candidate regions that may contain objects are initially generated, and each candidate region is classified with a CNN [9–11]. The second method is a regression-based method that uses CNNs to process entire images and predict object classification to achieve object localization [12,13]. This method is generally faster than the pre-classification method. To compensate for the accuracy limitation of the post-type detection method, Li et al. [14] and Sang et al. [15] increased the structure of a deep CNN to improve network accuracy. However, a deep CNN frequently leads to high computational complexity, and the information tends to gradually disappear after multilayer transmission. Densely connected convolutional networks [16] allow each layer of a network to accept the feature maps of all layers before it, and perform data aggregation in a concatenate manner. These processes enhance feature propagation, effectively solve gradient disappearance, realize feature reuse, and improve network classification accuracy. CNNs have been widely investigated in many applications. Considering the imbalance distribution of machinery health and the uncertainty of neural network learning characteristics, Jia et al. [17] proposed a deep normalized CNN to aid the understanding of what CNNs learn in fault diagnosis for machinery. Guo et al. [18] combined eight classical time–frequency features and six related-similarity features to obtain original features. Then, they used the training features of a recurrent neural network to accurately predict bearings' remaining useful life. To solve many scientific problems in mining science, Tadeusiewicz [19] et al. indicated that neural networks can be valuable, and presented several studies on the usage of neural networks in the mining industry. Ganovska [20] et al. developed 150 different configurations of neurons in hidden layers, and combined the Bayesian regularization and Levenberg-Marquardt algorithm to predict the surface roughness of mechanical products. Zhang [21] et al. used the vibration signals of a deep groove ball bearing, extracted the relevant features, and utilized a neural network to model the degradation for identifying and classifying fault types. These studies focused on algorithmic improvements in CNNs and innovations in different application scenarios, whereas few studies reported on high-precision parts used in small quantities, especially in the aerospace industry. Meanwhile, many difficulties, such as high complexity, low recognition efficiency, and insufficient robustness, still exist in the detection of special mechanical parts with complex illumination and background. Therefore, we use a You Only Look Once (YOLO) V3 algorithm [22] based on the fast detection speed of a regression method to simplify the original YOLO V3 network, fuse the data increase [23] and multiscale training [24] strategies, and train the improved YOLO V3 algorithm to improve the speed, accuracy, and stability of 0.8 cm darning needles and KR22 bearings in industrial scenarios. The detection effect of this method is remarkably improved on a built real-time industrial platform. Figure 1 shows the technical flowchart of this study.

The contributions of this paper are summarized as follows:

- We implemented a mechanism for the real-time recognition of mechanical parts based on an industrial detection platform.
- We proposed an image increase algorithm based on direction reversal (IIA-DR) to expand the data set and verify the feasibility of the IIA-DR.
- We designed an improved neural network structure and feature extraction algorithms based on YOLO V3 for industrial detection platforms, and report refined recognition accuracy.

- We compared this study with the YOLO V3 model. This Study outperforms YOLO V3 in terms of it's predictive estimation.

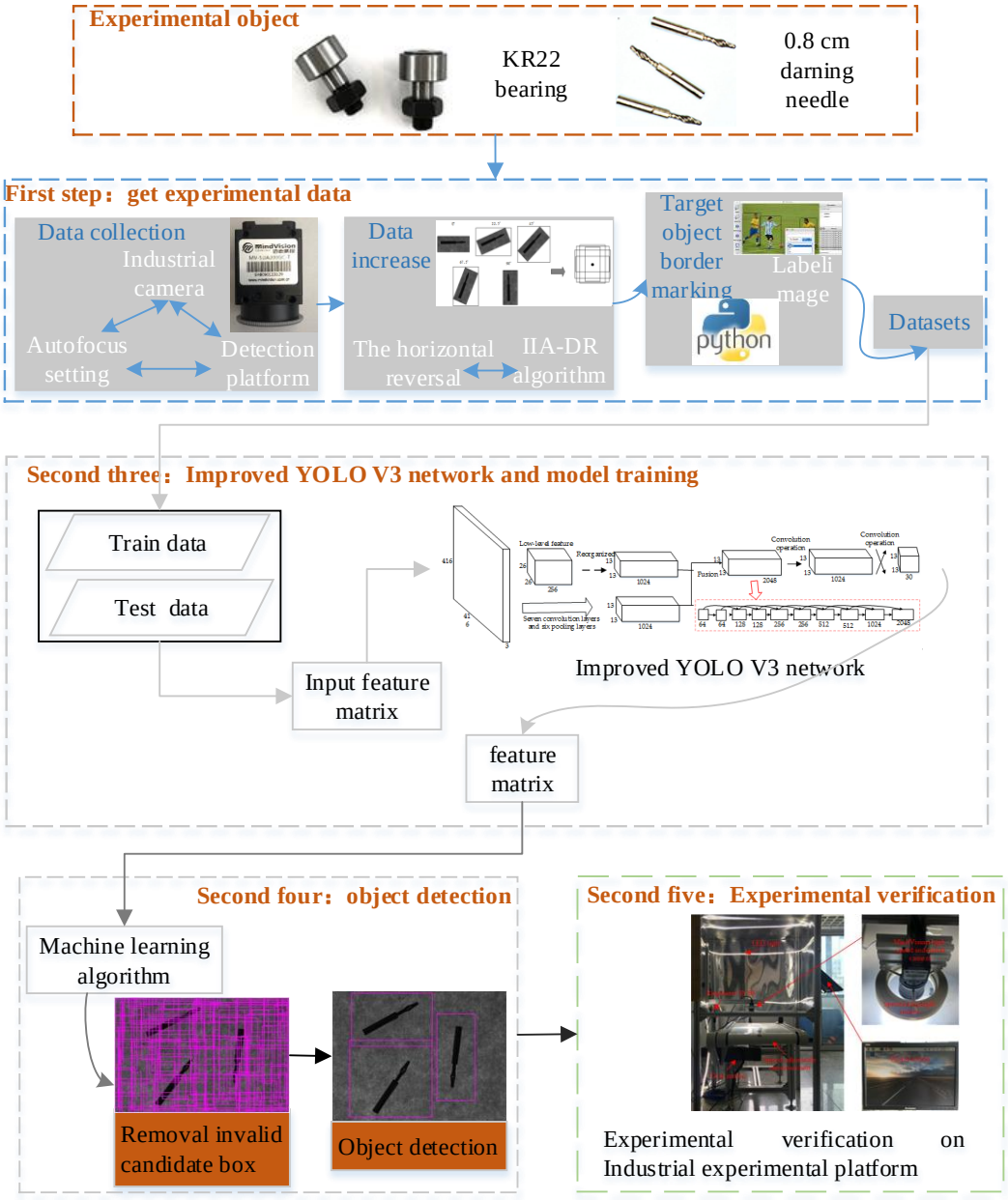

Figure 1. Technical flowchart of the study.

The rest of this paper is organized as follows. Section 2 presents an improved real-time object detection model and model training. Section 3 introduces an image increase algorithm that is based on direction reversal. Section 4 describes the experimental platform for real-time industrial object detection of mechanical parts. Section 5 presents the experimental data collection, data increase, border marking for samples of target objects, and dataset preparation. Section 6 provides the experimental results and analysis in detail. Section 7 concludes the paper and discusses future research issues.

2. Improved YOLO V3 Network Model and Model Training

2.1. Candidate Box Extraction and Object Detection Based on YOLO V3

The YOLO V3 network integrates candidate frame extraction, feature extraction, object classification, and object location into a neural network. The neural network directly extracts candidate regions from the target image, predicts the position and probability of object precision parts through the entire image feature, and transforms the positioning problem of object precision parts into a regression problem to achieve end-to-end detection. The detection of object precision parts involves the extraction of candidate frames from an input image to determine whether it contains the object precision parts when the position is given. The steps of the object detection algorithm based on the YOLO V3 network are divided into two parts, namely, candidate box extraction and object detection. The steps are described as follows.

2.1.1. Candidate Box Extraction

Candidate boxes represent the location of possible objects in an image. They can be extracted by using a region proposal network (RPN). The core concept is to use a CNN to directly generate candidate boxes, share the convolution features of the image with the entire detection network, and reduce the extraction time of the candidate box area. The RPN network extracts candidate regions by using a sliding window in the final convolutional layer, and generates multiscale candidate frames on the basis of the regression mechanism of the given initial specifications and positioning frames. Notably, the input image in the YOLO V3 detection method is divided into cells. Each cell is given B different specifications of the initial candidate box, and these candidate boxes serve as the initial position of the possibly existing object. Figure 2 shows that the predicted candidate frame is extracted through the volume layer network, and that the number of candidate frames per image is $M \times N \times B$.

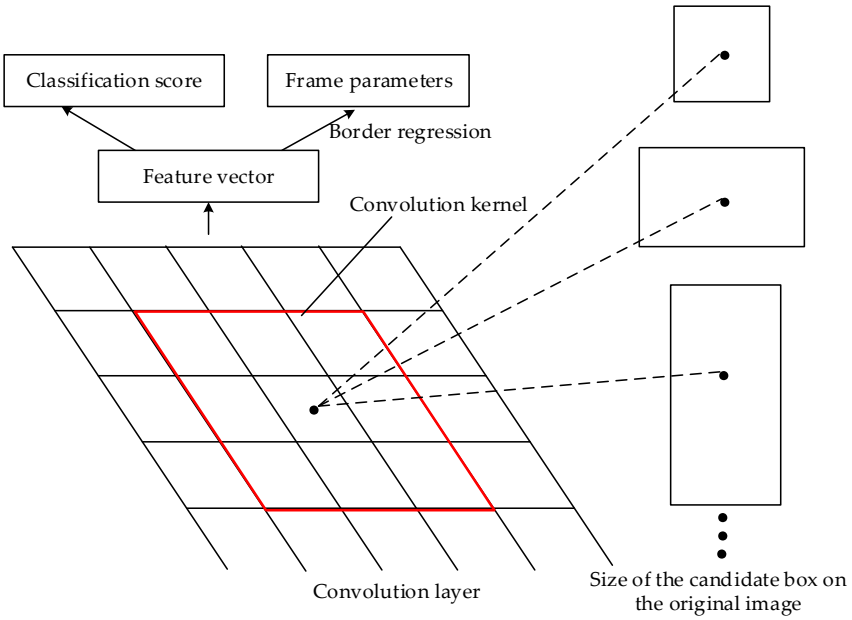

Figure 2. Extraction of candidate frames with different sizes.

2.1.2. Object Detection for Candidate Frames

The process detects the candidate frames and predicts the confidence of each candidate box in the object $Conf(object)$. Most candidate boxes do not contain the object precision parts. If the probability of specifying precision parts is directly predicted for each candidate box, then the difficulty of network learning will increase. Therefore, during the object detection process of precision parts,

the confidence level of some prediction frames is set to zero, which reduces the difficulty of network learning, shown as follows:

$$Conf(Object) = P_r(Object) \times IOU_{Pred}^{Truth} \tag{1}$$

where $P_r(Object)$ indicates the probability that the object to be detected is in the corresponding cell of the candidate frame; the calculation formula is shown in Formula (2). IOU_{Pred}^{Truth} is the ratio of the intersection area of the prediction frame to the actual frame and the area of the union; the calculation formula is shown in Formula (3). The object confidence of the cell corresponding to the candidate frame is $Conf(Object) = IOU_{Pred}^{Truth}$ when the cell has a detection object; otherwise, $Conf(Object) = 0$.

$$P_r(Object) = \begin{cases} 0 \text{ No object to be detected in the cell} \\ 1 \text{ An object to be detected in the cell exist} \end{cases} \tag{2}$$

$$IOU_{Pred}^{Truth} = \frac{area(box(Truth) \cap box(Pred))}{area(box(Truth) \cup box(Pred))} \tag{3}$$

2.2. Accurate Discriminant Method for Precision Parts Based on Non-Maximum Suppression

Formula (1) is used to determine whether an object exists in the candidate box when object part A is to be detected. The predicted object denotes the conditional probability of specifying the precision; hence, part A is $Pr(A|Object)$, and the confidence of the candidate box containing A is $Conf(A)$.

$$Conf(\mathrm{A}) = Pr(\mathrm{A}\big|\mathrm{Object}) \times Pr(Object) \times IOU_{Pred}^{Truth} \tag{4}$$

Each candidate box is responsible for predicting its probability of containing object A and the location of the boundary box. By calculating Formula (4), each candidate box outputs a parameter of six tuples $[X, Y, W, H, Conf(Object), Conf]$, where X and Y are the offset of the prediction box center relative to the cell boundary, and W and H are the ratios of the prediction box width to the entire image. Each image is transformed into the following network output vector T:

$$T = M \times N \times B \times [X, Y, W, H, Conf(Object), Conf] \tag{5}$$

where B denotes the prediction candidate box. The confidence in predicting that multiple regions of the image contain different types of objects can be obtained by the preceding detection process. To accurately select the location of the object, we use a non-maximum suppression method [23] in eliminating invalid frames. The specific steps are expressed as follows.

Step 1: Assume that N candidate boxes are obtained in the detected image. Obtain set $B = \{B_1, B_2 \ldots, B_n\}$ of the predicted candidate boxes by sorting the confidence obtained by Formula (8) in decreasing order.

Step 2: Calculate the intersection-over-union (IOU) of the other candidate boxes with the highest confidence. The candidate boxes have the same objects as those in the prediction and should be discarded when IOU exceeds a certain threshold; only B_1 is retained.

$$IOU(B_1, B_2) = \frac{area(B_1 \cap B_2)}{area(B_1 \cup B_2)} \tag{6}$$

Step 3: Repeat Steps 1 and 2 for the remaining candidate boxes until all candidate boxes are traversed.

Figure 3 shows the flow of the removal of invalid candidate boxes using the non-maximum suppression algorithm.

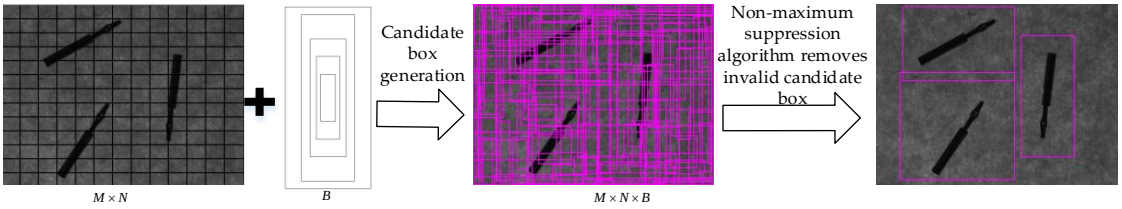

Figure 3. Removal of invalid candidate boxes using the non-maximum suppression algorithm.

2.3. Improved YOLO Network Structure

To realize real-time detection of precision parts, we use Tiny-YOLO with a simple structure and low computational complexity as the basic network of YOLO V3. Tiny-YOLO comprises 19 convolutional layers and 4 pooling layers, which alternately form a feedforward network, and easily lead to information loss layer by layer. This network fails to utilize multilayer feature information and reduces detection accuracy. To realize multilayer feature multiplexing and fusion, and to avoid the computational complexity caused by the new structure, we utilize the concept of a multilevel feature map fusion network proposed by Song et al. [25], although only in the low-resolution of the YOLO V3 network feature map. The Tiny-YOLO's 14th layer (resolution 1313) is replaced with dense modules (dotted line in Figure 4) by deeply embedding the dense modules to build a YOLO-dense network with dense connections (Table 1). A 14-volume layer can receive the multilayer convolution features of densely connected block outputs for feature multiplexing and fusion. The simplified network structure consists of seven convolutional layers and six pooling layers, as well as a feature reorganization layer and two convolutional layers for generating candidate frames and classifications.

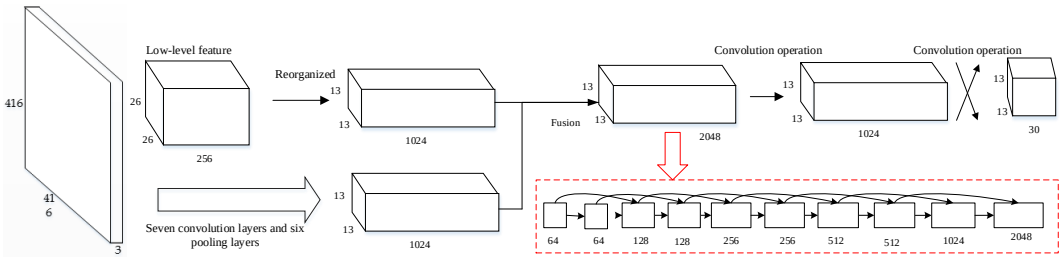

Figure 4. Structure of identification model network of precision parts.

Table 1. Parameter settings of the identification model network of precision parts.

Number of Layers	Operation Type	Kernel Size	Stride	Input	Output
1	Convolution layer	3 × 3	1	416 × 416 × 3	416 × 416 × 16
2	Pooling layer	2 × 2	2	416 × 416 × 16	208 × 208 × 16
3	Convolution layer	3 × 3	1	208 × 208 × 16	208 × 208 × 32
4	Pooling layer	2 × 2	2	208 × 208 × 32	104 × 104 × 32
5	Convolution layer	3 × 3	1	104 × 104 × 32	104 × 104 × 64
6	Pooling layer	2 × 2	2	104 × 104 × 64	52 × 52 × 64
7	Convolution layer	3 × 3	1	52 × 52 × 64	52 × 52 × 128
8	Pooling layer	2 × 2	2	52 × 52 × 128	26 × 26 × 128
9	Convolution layer	3 × 3	1	26 × 26 × 128	26 × 26 × 256
10	Pooling layer	2 × 2	2	26 × 26 × 256	13 × 13 × 256
11	Convolution layer	3 × 3	1	13 × 13 × 256	13 × 13 × 512
12	Pooling layer	2 × 2	1	13 × 13 × 512	13 × 13 × 512
13	Convolution layer	3 × 3	1	13 × 13 × 512	13 × 13 × 1024
14	Recombination layer	-	-	9 output layers + 13 output layers	13 × 13 × 2048
15	Convolution layer	3 × 3	1	13 × 13 × 2048	13 × 13 × 1024
16	Convolution layer	1 × 1	1	13 × 13 × 1024	13 × 13 × 30

2.4. Model Training of YOLO Network Algorithm

First, we need to select the appropriate initialization parameters for the aforementioned network training of the inspection network of precision parts. The selection of initial parameters considerably affects training efficiency. If the selection is not suitable, then cases may arise where convergence cannot be achieved. Second, we need to select the appropriate dataset. The tagged dataset is an important basis for network training. Finally, the appropriate loss function should be selected to ensure stable convergence of training.

Parameter initialization trains the network effectively. If the initial weight is extremely small, then the training speed is too slow; otherwise, the gradient may disappear. The weight initialization is performed using the "Xavier" method, which was proposed by Xavier et al. in 2016 [26]. This method determines the distribution range of the random initialization of parameters according to the input and output dimensions of each layer, thereby ensuring that the variance of each layer is consistent in forward propagation and back propagation. Assume n is the input dimension of the layer in which the parameter is located, and m is the output dimension. Then, the initialization weight of the layer is selected in the range of $\left[-\sqrt{6/(m+n)}, \sqrt{6/(m+n)}\right]$ in a uniformly distributed manner.

In network training, the error between the predicted and real values is calculated by the loss function using the idea of error backpropagation in the neural network. In this method, the weight of each layer in the network is constantly adjusted, and the training of the model is completed. The loss function in the object detection algorithm consists of two parts, namely, the error on the object category and the error at the object position. To reduce training difficulty, we increase the link of assessing whether an object is in the cell. The error of backpropagation is needed in calculating the loss function to obtain accurate judgment in YOLO V3 object detection. The loss function consists of three parts, namely, the position error $coordError$, the cell object confidence error $objConfError$, and the class error $classError$.

$$\text{loss} = \sum_{i=0}^{M\times N} coordError + objConfError + classError \tag{7}$$

The position error formula is $coordError = \lambda_{coord}\sum_{i=0}^{M\times N}\sum_{j=0}^{B} I_{ij}^{obj}[(x_i-\hat{x}_i)^2+(y_i-\hat{y}_i)^2+(\sqrt{w_i}-\sqrt{\hat{w}_i})^2+\left(\sqrt{h_i}-\sqrt{\hat{h}_i}\right)^2]$. The object confidence error for cell detection samples is $objConfError = \sum_{i=0}^{M\times N}\sum_{j=0}^{B} I_{ij}^{obj}(c_i-\hat{c}_i)^2 + \lambda_{noobj}\sum_{i=0}^{M\times N}\sum_{j=0}^{B} I_{ij}^{noobj}(c_i-\hat{c}_i)^2$. The class error of detection is $classError = \sum_{i=0}^{M\times N} I_{ij}^{obj}\sum_{obj\in class}(p_i(obj)-\hat{p}_i(obj))^2$. The contribution of each part of the loss to the network is different; thus, parameters λ_{coord} and λ_{noobj} are introduced to adjust the weights of each part. I_{ij}^{obj} indicates whether the center of the test sample falls in prediction box j of cell i. I_{ij}^{noobj} indicates whether the center of the object falls in prediction box j of cell i. x_i, y_i, w_i, and h_i denote the center coordinate and width of the rectangular box of the position tag. $\hat{x}_i$, $\hat{y}_i$, $\hat{w}_i$, and $\hat{h}_i$ denote the position information of the object to be detected as predicted.

3. Image Increase Algorithm Based on Direction Reversal

A neural network with strong generalization capability is obtained by expanding the training sample data to improve the quality and diversity of samples and to reduce the overfitting phenomenon of the neural network; hence, the neural network suits the application scenario and shows improved detection accuracy [23,27–29]. Stern et al. [23] used brightness changes to increase the samples. Ding et al. [28] conducted white balance processing on an image of insect samples to improve the accuracy of insect detection. Xue et al. [29] used adaptive histogram equalization to enhance a mango image and reduce the effect of illumination on image quality. Adaptive histogram equalization is adopted to adjust image brightness and increase the diversity of sample image illumination. Image clarity is weakened due to the fast operation of precision parts on a conveyor belt. As a result, the surface

color of precision machinery parts considerably differs from that of these parts under normal light, and affects the quality of sample images of precision machinery parts. The quality of training samples affects the detection effect of the model. In the present work, the horizontal reversal of the training samples and the rotation from [0°, 22.5°, 45°, 67.5°, 90°] were used to improve the sample image quality and increase the number of samples. A centered screenshot of the increased image was captured. The label was discarded when the object near the edge of the image was missing or completely lost. Finally, we obtained the image increase algorithm on the basis of direction reversal (IIA-DR).

Algorithm 1: Image increase algorithm based on direction reversal (IIA-DR)

Input: Manually selected different poses, real image Ti of different backgrounds
Output: Increased datasets L, and the set P of the corresponding detection object position label
1) Set the number of increase images D_i, and input the real image *Ti* of different locations and backgrounds selected manually;
2) Read the real image *Ti*, and take the input image as the template image;
3) Manually label the image in Step 2. Obtain the position label of the precision parts and the four vertex coordinates of the rectangular box, that is, $\left(x_{top}, y_{left}\right)$, $\left(x_{top}, y_{right}\right)$, $\left(x_{bottom}, y_{left}\right)$, and $\left(x_{bottom}, y_{right}\right)$;
4) Flip template image *Ti* horizontally. Image datasets $L = \{l_1, l_2, l_3, \ldots, ln\}$ are generated by rotation transformation from $[0°, 22.5°, 45°, 67.5°, 90°]$ angles;
5) Transform the position label on the template image and obtain the corresponding four vertices, that is, (x_1, y_1), (x_2, y_2), (x_3, y_3), and (x_4, y_4);
6) Use Formulas (8–11) to correct the coordinates of the four points. Obtain the new label of the generated single image as $\left(x_{top'}, y_{left'}\right)$, $\left(x_{top'}, y_{right'}\right)$, $\left(x_{bottom'}, y_{left'}\right)$, and $\left(x_{bottom'}, y_{right'}\right)$;

$$x_{top'} = \min(x_1, x_2, x_3, x_4) \tag{8}$$

$$x_{bottom'} = \max(x_1, x_2, x_3, x_4) \tag{9}$$

$$y_{left'} = \min(y_1, y_2, y_3, y_4) \tag{10}$$

$$y_{right'} = \max(y_1, y_2, y_3, y_4) \tag{11}$$

7) If m = true, execute Steps 4–6;
8) Output increased dataset $L = \{l_1, l_2, l_3, \ldots, ln\}$ and the corresponding set of detection object location tags $P = \{P_1, P_2, P_3, \ldots, P_n\}$.

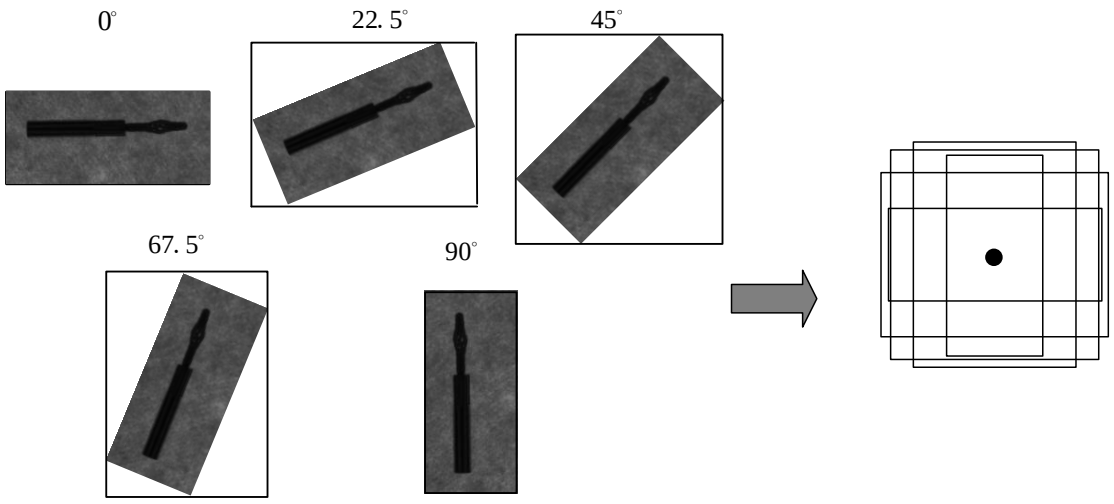

Figure 5. Initial candidate frame selection of precision parts.

On the basis of the preceding method, the corresponding image datasets and labels (including classification and position labels) were generated by the template images of different precision parts. In the process of object detection and recognition of precision parts, the candidate frame was used to extract the precision parts in the object area and identify their categories, where accurate object position information was required to avoid the extraction of incomplete object precision parts. Selecting the appropriate initial candidate box specification could not only increase the training speed of the network, but also obtain further accurate location information. To adapt the position information of precision parts in different positions and postures, the specification of the position frame when the precision parts rotate at [0°, 22.5°, 45°, 67.5°, 90°] was taken as the specification of the initial candidate frame. Figure 5 shows the initial candidate box selection process for the precision parts.

4. Industrial Real-Time Object Detection Experimental Platform

Figure 6 shows the structural diagram of the experimental detection platform designed in this study. The platform included a conveyor belt, data processor, data acquisition sensor, light source, and other mechanical supports. The device used to input and display data was a 32-inch industrial touchscreen. The visual-sensing device used a MindVision high-speed industrial camera (http://www.mindvision.com.cn/cpzx/info_7.aspx?itemid=1758&lcid=21) with an electronic rolling shutter, which can collect high-speed motion samples in real time. The data processor was a Raspberry PI B3 (https://www.raspberrypi.org/products/raspberry-pi-3-model-b/). An adjustable ambient light-emitting diode light strip (https://store.waveformlighting.com/products/real-uv-led-strip-lights-16-ft-5-m-reel) was installed to ensure sufficient light in the system box. A special ring light source for the alarm was installed outside the industrial camera to fill the light on the test sample and obtain a clear sample image. For data analysis, the workstation used to reduce the computing load of the data processor was configured with Intel Xeon CPU E5-1620 V3 3.5 GHz, 16 G memory, NVIDIA GeForce GTX1080, and Ubuntu 16.04. Meanwhile, the computer processor of the detection device and the workstation were installed with OpenCV 3.1, TensorFlow 1.3, and YOLO V3. The Raspberry PI B3 contains a wireless communication module that can realize end-to-end communication between the experimental detection platform and the workstation. Specifically, traditional visual calibration methods cannot easily obtain high-definition experimental images for small mechanical parts. Thus, we used manual experience to calibrate our visual inspection platform.

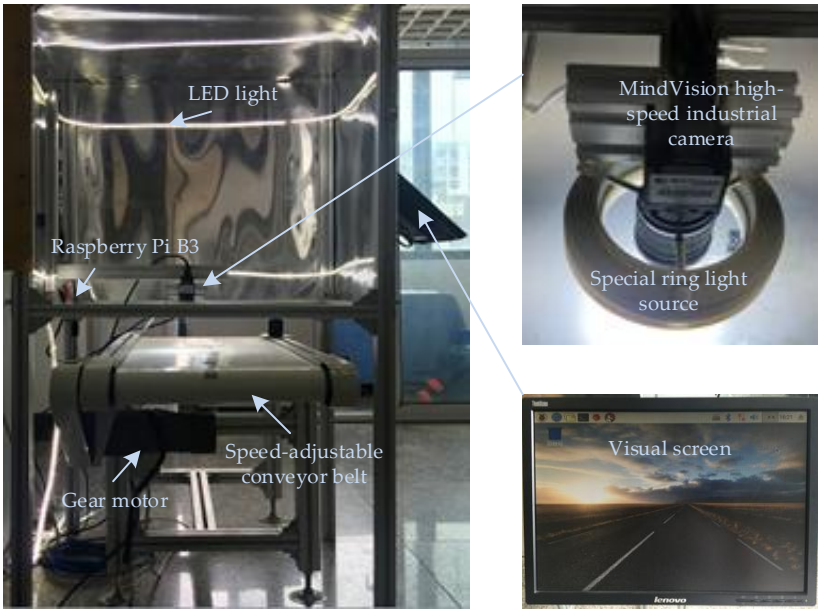

Figure 6. Experimental platform for real-time industrial object detection.

5. Experimental Data

5.1. Experimental Data Collection

The high-speed MindVision industrial camera was used to collect images of 0.8 cm darning needles and KR22 bearings in motion on a self-built, real-time industrial object detection platform. The real-time industrial testing platform was introduced in Section 4. The distance between the camera and the conveyor belt was 10 cm. The datasets for the 0.8 cm darning needles and KR22 bearings were established by constantly changing object positions. Of the 3000 images obtained, 2000 images were randomly selected as training images. The training images contained 5602 0.8 cm darning needles and KR22 bearings. The remaining 1000 images were used as test images. These test images contained 2361 0.8 cm darning needles and KR22 bearings. Figure 7 shows a sample image.

Figure 7. Example of autonomous collection of experimental datasets, including the 0.8 cm darning needle dataset and the KR22 bearing dataset.

5.2. Target Object Border Marking for Samples

To verify the effectiveness of the proposed method, we used manual methods for labeling real data, and Python programs to process the increased dataset automatically. Labeling the object frame manually on the 7963 0.8 cm darning needles and KR22 bearings in the original training set took approximately 10.56 h. The object border was automatically marked by a Python program when the increased dataset was generated. Meanwhile, the increased dataset containing 6000 target objects took 0.45 h.

5.3. Dataset Preparation

The training dataset was composed of image data in a complex industrial situation. During feature model extraction, the recognition performance of the feature model under different parameters was evaluated to refine the model. Two types of datasets were obtained by collecting and increasing the training samples on the real-time testing platform in Section 4. As shown in Table 2, dataset A of the original image comprised the 0.8 cm darning needle and KR22 bearing dataset. This dataset is called "real datasets" and has 6963 0.8 cm darning needles and KR22 bearings. Dataset B comprised the 0.8 cm darning needle and KR22 bearing dataset. This dataset is called "increased datasets" and has 6000 0.8 cm darning needles and KR22 bearings.

Table 2. Datasets and size.

Datasets	Processing Methods	Number of Images	Number of B-Box Labeling
A	Manually label the border	3000	6963
B	The program automatically labels	3000	6000

6. Experimental Results and Analysis

6.1. Evaluation Index

To evaluate the stability and robustness of the algorithm in this study, recall rate, accuracy rate, average IOU value, and *F1* value are defined to evaluate the model. Recall is the proportion of precision parts detected in images. Precision is the correct proportion of all tested candidate frames. Average IOU is the coincidence degree between the prediction and actual boxes, thereby reflecting the accuracy of the detected location of the object precision parts. The performance of the proposed test model is evaluated by using the statistics of *F1* measure, where P and R denote the precision and recall rates, respectively.

$$F1 = \frac{2PR}{P + R} \tag{12}$$

6.2. Data Increase Experimental Results

KR22 bearings and 0.8 cm darning needles with different backgrounds and postures were selected to increase the image data. The sample image quality and the number of increased samples are improved by horizontal mirroring of the training samples and angular rotation from [0°, 22.5°, 45°, 67.5°, 90°], and by intercepting the central location of object detection in the increased image. Figure 8 shows an example of an image after partial data increase. The label is discarded when the object near the edge of the image is missing or completely lost. The same amplification data and corresponding tags as input data objects are generated when data is supplied to the data increase algorithm.

Figure 8 shows that the 0.8 cm darning needle and KR22 bearing datasets on the increased dataset can increase the image data with various poses based on the input image. A comparison of the increased dataset and real data reveals that the posture of the detected object in the increased data image is abundant, and that the specifications are uniform. Subsequent observations of manual and automatic labeling samples can be found on the 0.8 cm darning needle datasets. The manual labeling of samples and the automatic labeling of the sample area proposed by the program show no subjective difference. For the KR22 bearing dataset, the object position of the automatically labeled sample area is at the center of the labeled box, although the manual labeling of the sample area is more compact than the automatic labeling of the sample area is. Despite no manual labeling of the sample compact, the automatic labeling of the sample area is in a relatively compact range. The comparison shows that the experimental data increase in this study is diverse and feasible.

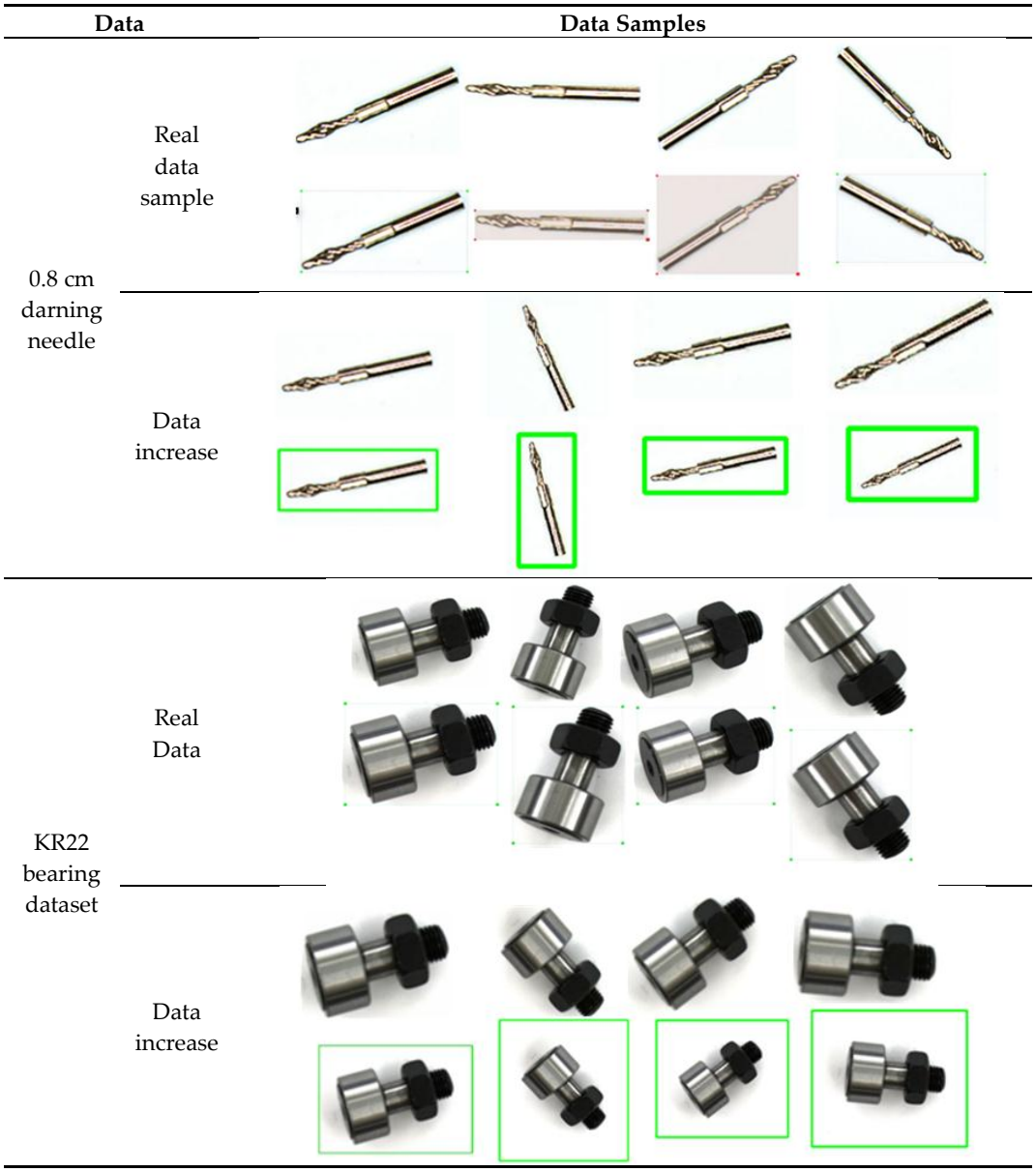

Figure 8. Comparison of experimental results between increased and real data in automatic and manual annotation.

6.3. Model Training Strategy and Model Validation Parameter Analysis

To speed up the training and prevent overfitting, we set the weight attenuation coefficient to the empirical value of 0.0005 and used a multi-stepping strategy to set the learning rate. Table 3 presents the learning strategy. In the experiment, the f value was set to 0.60, and the IOU threshold was 0.65. The initial network learning rate was set to 0.003, and the activation function was set to leaky.

Table 3. Multi-stepping strategy for learning rate.

Number of Training Steps	Learning Rate
0–1000	0.0001
1000–3000	0.001
3000–10000	0.0001
10000–30000	0.00001

Different thresholds that affect performance must be considered in evaluating the suitability of the trained model for practical applications. This section discusses the effects of different thresholds on recall rate and accuracy to determine the suitable training thresholds and scale. The trained model was evaluated on the verification set, and different thresholds were used for the verification. Table 4 shows the test results when the candidate frame with prediction confidence higher than the threshold is used as the object detection result.

Table 4. Results of the verification set.

Threshold	Number of Images	Number of Correctly Predicted Precision Parts	Total Precision Parts	Average IOU	Recall Rate	Number of Prediction Frames	Accuracy Rate
0.00	500	500	500	89.25%	100.00%	99,885	0.87%
0.10	500	500	500	90.65%	100.00%	920	97.38%
0.20	500	500	500	90.85%	100.00%	910	98.88%
0.30	500	500	500	91.15%	100.00%	900	98.96%
0.40	500	500	500	91.20%	100.00%	880	99.18%
0.50	500	500	500	91.20%	100.00%	880	99.18%
0.55	500	500	500	91.25%	100.00%	830	99.36%
0.60	500	500	500	93.25%	100.00%	800	100.00%
0.65	500	500	500	92.65%	100.00%	800	100.00%
0.70	500	500	500	92.35%	100.00%	800	100.00%
0.75	500	500	500	92.26%	99.87%	798	100.00%
0.80	500	500	500	92.24%	99.37%	763	100.00%
0.85	500	500	500	91.24%	98.87%	736	100.00%
0.90	500	500	500	90.25%	97.87%	690	100.00%
0.00	500	500	500	89.25%	100.00%	99,885	87.00%

Table 4 shows that the average IOU value and accuracy of the model increases for the 500-image verification set when the threshold ranges between 0.00 and 0.55. At the same time, the number of prediction frames gradually decreases. The recall and accuracy rates are 100%, the average IOU is 93.25%, and the number of prediction frames is stable at 800 when the threshold is 0.60. The number of detection frames decreases when the threshold ranges between 0.75 and 0.95, although the detection accuracy value is 100% in the region. This result indicates a missed object. In terms of detection speed, the average detection time per image is approximately 0.0034 s. Real-time requirements for the identification and location of object detection are satisfied.

6.4. Experimental Results of YOLO V3 and the Algorithm on the Dataset

To verify the effectiveness and superiority of the proposed algorithm, we utilized the original YOLO V3 algorithm on the established industrial object detection platform. Moreover, the original YOLO V3 algorithm is compared and analyzed by using the designed object datasets of precision parts A and B. The relevant settings of the original YOLO V3 algorithm are the same as the original experimental settings [30]. According to the test and comparative analysis of the designed datasets A and B, the statistical results of the context identification accuracy, the predicted probability estimation value of the algorithm designed by the proposed algorithm, and the original YOLO V3 algorithm are shown in Table 5. We also analyzed the experiment from subjective and objective perspectives.

6.4.1. Analysis of Subjective Test Results

Two types of dataset, A and B, were evaluated using the proposed algorithm, and subjective detection was performed on 20 images. Some test results are shown in Figure 9. The detection object is matched. The method can effectively surround the object in the real dataset A and increased dataset B.

The detection method can realize multi-object detection in one image, and the detection results are expressed as follows. For the preceding experimental verification, the YOLO network-based object detection method for precision parts and the deep network model constructed by the data can be used to detect the position of precision parts and to identify them. The accuracy can reach 100%, and the detection speed can reach 290 fps, both of which satisfy real-time requirements.

(**a**)

(**b**)

Figure 9. Example of resulting object recognition for subjective precision parts. (**a**) Example of resulting object recognition of mechanical parts for dataset A. (**b**) Example of resulting object recognition of mechanical parts for dataset B.

6.4.2. Analysis of Objective Test Results

Table 5 shows that in the real dataset A, the estimated probability of the 0.8 cm darning needle part evaluated in the YOLO V3 algorithm is 0.927, the variance is 0.011, and the exact value is 0.923. For the KR22 bearing data, the predicted probability of the test in the YOLO V3 algorithm is 0.948, the variance is 0.008, and the exact value is 0.934. The improved algorithm in this study has a prediction probability of 0.961 in the 0.8 cm darning needle data, a variance of 0.012, and an exact value of 0.944. For the KR22 bearing data, the predicted probability of this algorithm is 0.974, the variance is 0.008, and the exact value is 0.956. Subsequent observations of increased dataset B show that the estimated probability and accuracy of the proposed algorithm are 0.005 and 0.030 higher, respectively, than that of YOLO V3 on 0.8 cm darning needle data. For the KR22 bearing data, the estimated probability and accuracy of the

proposed algorithm are 0.090 and 0.031 higher, respectively than that of YOLO V3. Thus, the algorithm and YOLO V3 have a good mean square error.

The comparison of objective analyses shows that the detection effect of the algorithm is clearly better than that of the traditional detection model. This result is mainly due to the structural improvement of the original YOLO V3 neural network, the reduction of the number of network layers, and the improvement of network accuracy and training time.

Table 5. Test results of the system for different datasets.

		YOLO V3			Proposed Algorithm		
		Predicted Probability Estimate		Accurate Value	Predicted Probability Estimate		Accurate Value
		Mean	Variance		Mean	Variance	
A	0.8 cm darning needle	0.927	0.011	0.923	0.961	0.012	0.944
	KR22 bearing	0.948	0.008	0.934	0.974	0.008	0.956
B	0.8 cm darning needle	0.937	0.009	0.935	0.942	0.010	0.965
	KR22 bearing	0.955	0.012	0.954	0.964	0.008	0.985

6.4.3. Analysis of Multi-Category Test Results

We collected multi-category images on the real-time industrial object detection experimental platform to evaluate the experimental results of this study in multiple categories. The experimental setting is the same as that in Section 6.3. The test results of the multi-category data are shown in Figure 10.

Figure 10 shows that the proposed algorithm can accurately detect multiple categories. This observation is the main reason for our use of Formula (1) to remove invalid candidate boxes, and the non-maximum suppression algorithm to obtain an accurate recognition of detected objects. The average recognition rate of the test results from the multi-category target images is 81.19%. As shown in Table 5, the recognition rate of multi-category targets is lower than the average recognition rate of single-category targets. To improve the recognition accuracy of multi-class targets, we can expand the sample diversity of the training dataset and place the erroneous samples in the corresponding training datasets to obtain a robust feature model.

Figure 10. Multi-category test result.

7. Summary

We have presented a detection and recognition algorithm for precision parts based on YOLO V3 network structure to satisfy the identification requirements of industrial machinery parts. The proposed algorithm adopts a multilevel-feature fusion method to improve the detection accuracy of YOLO V3 and the robustness of the model, and to increase the network structure. The feature fusion layer enhances the network's capability to express subtle features. We implemented IIA-DR and obtained the method of scaling, rotating, and translating detected objects. Then, we generated the image and label annotations suitable for model training, which solved the few training data and time-consuming manual labeling problems. We constructed an experimental platform for industrial testing to verify the proposed algorithm.

We will increase the experimental object, improve the network's capability to detect multiple objects, and combine the real-life production environment of a factory in our next research work. We will develop a real-time-based detection method for defective mechanical parts on the basis of this study to improve robustness. We will focus on the unsuitable recognition effect problem on multi-category parts on conveyer belts under complex illumination and backgrounds (Figure 11).

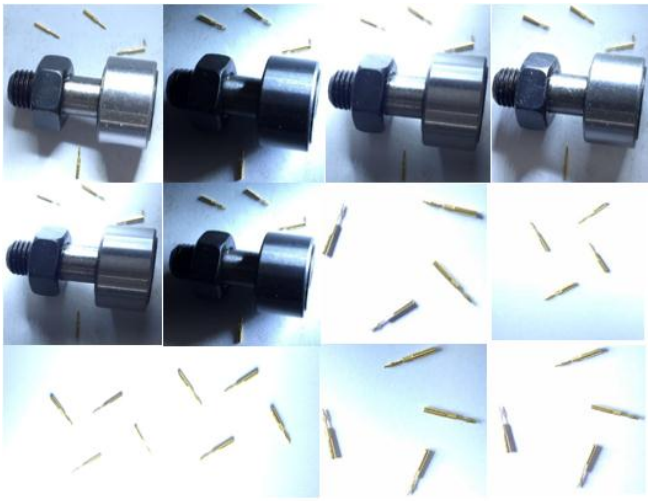

Figure 11. Sample images of multiple categories of mechanical parts with complex illumination and background.

Author Contributions: S.L. conceived of and designed the study. J.Y. and Z.W. worked on the algorithm design. J.Y. and Z.G. implemented the baseline methods. J.Y. and Z.W. collected and processed the data. W.L. and J.Y. revised and polished the manuscript. All authors have read and approved the final manuscript.

Acknowledgments: J.Y. received financial support from the China Scholarship Council (CSC) and Guizhou Province Postgraduate Research Fund Project No. 18057. This work is supported by the National Natural Science Foundation of China under Grant No. 51475097, 91746116, and 51741101; the Science and Technology Foundation of Guizhou Province under Grant No. [2015] 4011, [2016] 5013; and the college students' science and technology innovation project of Xi'an Polytechnic University No. 1720. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research, and the support of the National Institute of Measurement and Testing Technology which provided the Semi-anechoic laboratory.

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

References

1. Kamalakumari, J.; Vanitha, D.M. IMAGE SEQUENCES BASED FACIAL EXPRESSION RECOGNITION USING SUPPORT VECTOR MACHINE. *Int. J. Eng. Technol.* **2017**, *9*, 3605–3609.
2. Liu, B.; Chen, Y.; Qi, X. Research on the Application of Mechanical Parts Using Neural Network in Image Recognition. In Proceedings of the 2015 Fifth International Conference on Instrumentation and Measurement, Computer, Communication and Control (IMCCC), Qinhuangdao, China, 18–20 September 2015; pp. 659–662.
3. Jiang, J.; Chen, Z.; He, K. A feature-based method of rapidly detecting global exact symmetries in CAD models. *Comput. Aided Des.* **2013**, *45*, 1081–1094. [CrossRef]
4. Inui, M.; Umezu, N. Fast Detection of Head Colliding Shapes on Automobile Parts. *J. Adv. Mech. Des. Syst. Manuf.* **2013**, *7*, 818–826. [CrossRef]
5. He, T.; Yu, K.; Chen, L.; Lai, K.; Yang, L.; Wang, X.; Zhai, Z. Image classification and recognition method to mechanical parts based on fractal dimension. In Proceedings of the 2017 International Conference on Mechanical, System and Control Engineering (ICMSC), St. Petersburg, Russia Mechanical, 19–21 May 2017; pp. 63–66.
6. Wan, W.H.; Xu, J.; Chen, P.L.; Liu, M.H. Classification Modeling of Parts for Complex Machinery Product Based on Design Structure Matrix. *Appl. Mech. Mater.* **2014**, *680*, 539–542. [CrossRef]
7. Yang, J.; Yang, G. Modified Convolutional Neural Network Based on Dropout and the Stochastic Gradient Descent Optimizer. *Algorithms* **2018**, *11*, 28–43. [CrossRef]
8. Yang, G.; Yang, J.; Sheng, W.; Fernandes Junior, F.E.; Li, S. Convolutional Neural Network-Based Embarrassing Situation Detection under Camera for Social Robot in Smart Homes. *Sensors* **2018**, *18*, 1530. [CrossRef] [PubMed]
9. Kumar, A.; Kim, J.; Lyndon, D.; Fulham, M.; Feng, D. An Ensemble of Fine-Tuned Convolutional Neural Networks for Medical Image Classification. *IEEE J. Biomed. Health Inf.* **2016**, *21*, 31–40. [CrossRef] [PubMed]

10. Salamon, J.; Bello, J.P. Deep Convolutional Neural Networks and Data Augmentation for Environmental Sound Classification. *IEEE Signal Process. Lett.* **2017**, *24*, 279–283. [CrossRef]
11. Yang, Y.; Luo, H.; Xu, H.; Wu, F. Towards Real-Time Traffic Sign Detection and Classification. *IEEE Trans. Intell. Transp. Syst.* **2016**, *17*, 2022–2031. [CrossRef]
12. Akcay, S.; Kundegorski, M.E.; Willcocks, C.G.; Breckon, T.P. Using Deep Convolutional Neural Network Architectures for Object Classification and Detection Within X-Ray Baggage Security Imagery. *IEEE Trans. Inf. Forensics Secur.* **2018**, *13*, 2203–2215. [CrossRef]
13. Kheradpisheh, S.R.; Ganjtabesh, M.; Thorpe, S.J.; Masquelier, T. STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks* **2018**, *99*, 56–67. [CrossRef] [PubMed]
14. Li, J.; Liang, X.; Shen, S.; Xu, T.; Feng, J.; Yan, S. Scale-aware fast R-CNN for pedestrian detection. *IEEE Trans. Multimedia* **2018**, *20*, 985–996. [CrossRef]
15. Sang, J.; Guo, P.; Xiang, Z.; Luo, H.; Chen, X. Vehicle detection based on faster-RCNN. *J. Chongqing Univ.* **2017**, *40*, 32–36.
16. Chen, Y.; Kang, X.; Wang, Z.J.; Zhan, Q. Densely Connected Convolutional Neural Network for Multi-purpose Image Forensics under Anti-forensic Attacks. In Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security, Innsbruck, Austria, 20–22 June 2018; pp. 91–96.
17. Jia, F.; Lei, Y.; Lu, N.; Xing, S. Deep normalized convolutional neural network for imbalanced fault classification of machinery and its understanding via visualization. *Mech. Syst. Sig. Process.* **2018**, *110*, 349–367. [CrossRef]
18. Guo, L.; Li, N.; Jia, F.; Lei, Y.; Lin, J. A recurrent neural network based health indicator for remaining useful life prediction of bearings. *Neurocomputing* **2017**, *240*, 98–109. [CrossRef]
19. Tadeusiewicz, R. Neural networks in mining sciences–general overview and some representative examples. *Arch. Min. Sci.* **2015**, *60*, 971–984. [CrossRef]
20. Ganovska, B.; Molitoris, M.; Hosovsky, A.; Pitel, J.; Krolczyk, J.B.; Ruggierio, A.; Krolczyk, G.M.; Hloch, S. Design of the model for the on-line control of the AWJ technology based on neural networks. *IJEMS* **2016**, *23*, 279–286.
21. Zhang, L.; Tao, J. Research on Degeneration Model of Neural Network for Deep Groove Ball Bearing Based on Feature Fusion. *Algorithms* **2018**, *11*, 21–40. [CrossRef]
22. Redmon, J. YOLO: Real-Time Object Detection. Available online: https://pjreddie.com/darknet/yolo/ (accessed on 20 October 2017).
23. Stern, U.; He, R.; Yang, C.-H. Analyzing animal behavior via classifying each video frame using convolutional neural networks. *Sci. Rep.* **2015**, *5*, 1–13. [CrossRef] [PubMed]
24. Fazan, F.S.; Brognara, F.; Fazan Junior, R.; Murta Junior, L.O.; Virgilio Silva, L.E. Changes in the Complexity of Heart Rate Variability with Exercise Training Measured by Multiscale Entropy-Based Measurements. *Entropy* **2018**, *20*, 47–57. [CrossRef]
25. Song, W.; Han, J.; Hua, T. A method for medical image retrieval using multi-level feature fusion. *J. Inf. Comput. Sci.* **2009**, *6*, 967–974.
26. Junior, J.R.B.; Do Carmo Nicoletti, M. Enhancing Constructive Neural Network Performance Using Functionally Expanded Input Data. *J. Artif. Intell. Soft Comput. Res.* **2016**, *6*, 119–131. [CrossRef]
27. Zhong, Z.; Zheng, L.; Kang, G.; Li, S.; Yang, Y. Random erasing data augmentation. Available online: https://arxiv.org/abs/1708.04896 (accessed on 16 November 2017).
28. Ding, W.; Taylor, G. Automatic moth detection from trap images for pest management. *Comput. Electron. Agric.* **2016**, *123*, 17–28. [CrossRef]
29. Xue, Y.; Huang, N.; Tu, S.; Mao, L.; Yang, A.; Zhu, X.; Yang, X.; Chen, P. Immature mango detection based on improved YOLOv2. *Trans. Chin. Soc. Agric. Eng.* **2018**, *34*, 173–179.
30. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.