applied sciences

MDPI

Article

Buckle Pose Estimation Using a Generative Adversarial Network

Hanfeng Feng [1,†], Xiyu Chen [2,†], Jiayan Zhuang [2], Kangkang Song [2], Jiangjian Xiao [2] and Sichao Ye [2,*]

[1] College of Electrical Engineering Computer Science, Ningbo University, Ningbo 315211, China
[2] Ningbo Institute of Materials Technology and Engineering, Chinese Academy of Sciences, Ningbo 315201, China
* Correspondence: yesichao@nimte.ac.cn
† These authors contributed equally to this work.

Abstract: The buckle before the lens coating is still typically disassembled manually. The difference between the buckle and the background is small, while that between the buckles is large. This mechanical disassembly can also damage the lens. Therefore, it is important to estimate pose with high accuracy. This paper proposes a buckle pose estimation method based on a generative adversarial network. An edge extraction model is designed based on a segmentation network as the generator. Spatial attention is added to the discriminator to help it better distinguish between generated and real graphs. The generator thus generates delicate external contours and center edge lines with help from the discriminator. The external rectangle and the least square methods are used to determine the center position and deflection angle of the buckle, respectively. The center point and angle accuracies of the test datasets are 99.5% and 99.3%, respectively. The pixel error of the center point distance and the absolute error of the angle to the horizontal line are within 7.36 pixels and 1.98°, respectively. This method achieves the highest center point and angle accuracies compared to Hed, RCF, DexiNed, and PidiNet. It can meet practical requirements and boost the production efficiency of lens coatings.

Keywords: buckle; generative adversarial network; high precision; pose estimation; real time

Citation: Feng, H.; Chen, X.; Zhuang, J.; Song, K.; Xiao, J.; Ye, S. Buckle Pose Estimation Using a Generative Adversarial Network. *Appl. Sci.* **2023**, *13*, 4220. https://doi.org/10.3390/app13074220

Academic Editors: João M. F. Rodrigues and Antonella Petrillo

Received: 22 December 2022
Revised: 21 March 2023
Accepted: 21 March 2023
Published: 27 March 2023

1. Introduction

Intelligent manufacturing has transformed the manufacturing industry. Although it improves production efficiency and product quality, it also faces new challenges in the automated production of mechanical parts. In industrial production, computer vision technology aims to process collected images into relevant information needed for industrial production, such as the pose of materials [1] and the presence of defects [2], through algorithm processing. The information is converted into instruction codes and transmitted to industrial robots that perform operations such as grasping and disassembly. Newly manufactured lenses must first be removed from their plastic plates, and their encapsulating buckles must be disassembled before being placed on an aluminum plate for coating. Manual buckle disassembly is laborious and characterized by temporal, efficiency, and quality constraints. Therefore, to automate the coating process, it is necessary to design an accurate real-time buckle pose estimation algorithm that assists these robots in automatically disassembling the buckle.

To address this need, this paper proposes a model that identifies the pose of a buckle on a plastic plate by obtaining its center position and deflection angle relative to the horizontal plane. To achieve this task, three key requirements must be met:

(1) The model must be generalizable enough to adaptively handle significant buckle shape differences. Figure 1 presents three common buckle types (buckles are inside the red box).
(2) The model must distinguish the buckle from its background. The differences between these two are small. Figure 1 illustrate this scenario.

(3) The accuracy of buckle pose estimation must be high enough to reduce the loss caused by the clumsy disassembly of the machine.

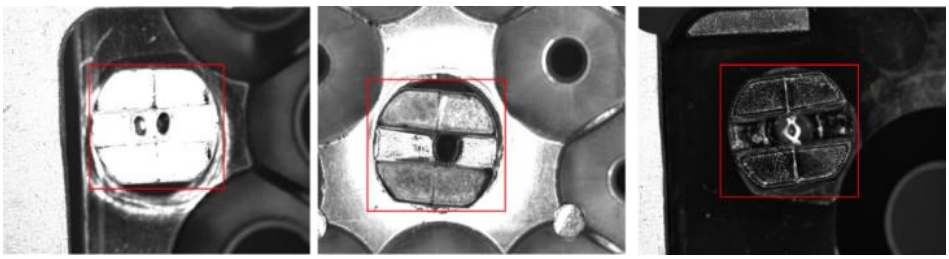

Figure 1. Different buckle types (buckles are inside the red box).

To meet these requirements, this paper proposes a generative adversarial network (GAN)-based [3] buckle pose estimation algorithm. A generator applies an edge extraction network to classify each pixel in the image, and the outer contours and center edge lines of buckles are regressed to estimate their center position and deflection angle. A discriminator-assisted generator is applied during training for edge refinement. First, its encoder is used for feature extraction and decoder feature reconstruction. A dilated convolution [4] is then performed so that the network can obtain information about the larger receptive field while paying attention to the overall target characteristics. Thus, the network can regress the overall outline information of the target to determine its pose to meet Requirements (1) and (2). Discriminators are introduced during training to assist the generator. Spatial attention is added to the discriminator, causing it to pay more attention to the difference between the generated image and its edge in the ground truth image. An auxiliary generator increases the attention paid to the detailed edge information and improves the accuracy to meet Requirement (3).

2. Related Work

In recent years, an increasing number of visual detection algorithms have been developed [5–7]; they are mainly divided into traditional and deep learning algorithms.

2.1. Traditional Detection Approaches

Traditional methods require manually labeled features designed for specific conditions and classifiers or template matching methods to detect targets using these features. The algorithms enumerate all possible targets in input images by combining several classical methods with parametric fine-tuning. Examples include cascade classification [8], sparse Fourier transformation [9], histogram of oriented gradient edge shape characteristic description [10], deformable part detection based on components [11], Haar wavelet characterization, and support vector machine (SVM) prediction [12]. Yu et al. [13] proposed a crack extraction method that used multiscale morphological operations for connector crack detection. However, it was susceptible to changes in background brightness. Zhi et al. [14] used a double-threshold method and gear involute geometric relationship to determine the tooth pitch in a local image of a gear by reversely mapping the filtered pixel information to the base circle and calculating the phase angle. However, this measurement method was susceptible to data diversity problems. Meanwhile, Zhang et al. [15] proposed a method to detect train center plate bolts. Gabor wavelets of different scales were used to act on the image, the weight obtained by each channel was optimized by a genetic algorithm, and the weighted features were classified using SVM. However, the requirements for manual design features relied too much on human subjectivity and, the method lacked generalizability.

Owing to the characteristics of large background interference and diverse buckle shape data, it is necessary to analyze the characteristics of each buckle using traditional methods. Moreover, the recognition accuracy depends largely on the feature generation process, which limits their applicability to industrial applications. Therefore, this study compares traditional binarization and segmentation methods. Binarization methods can be

divided into local and global categories. Global methods use only one threshold over the entire image, whereas local methods use multiple regional thresholds. We considered two contemporary methods, those developed by Otsu [16] and Wellner [17], for their global and local thresholds, respectively. Furthermore, we used the watershed segmentation algorithm [18] to segment images into disjointed regions that are related in terms of certain attributes. The results of these three methods are shown in Figure 2.

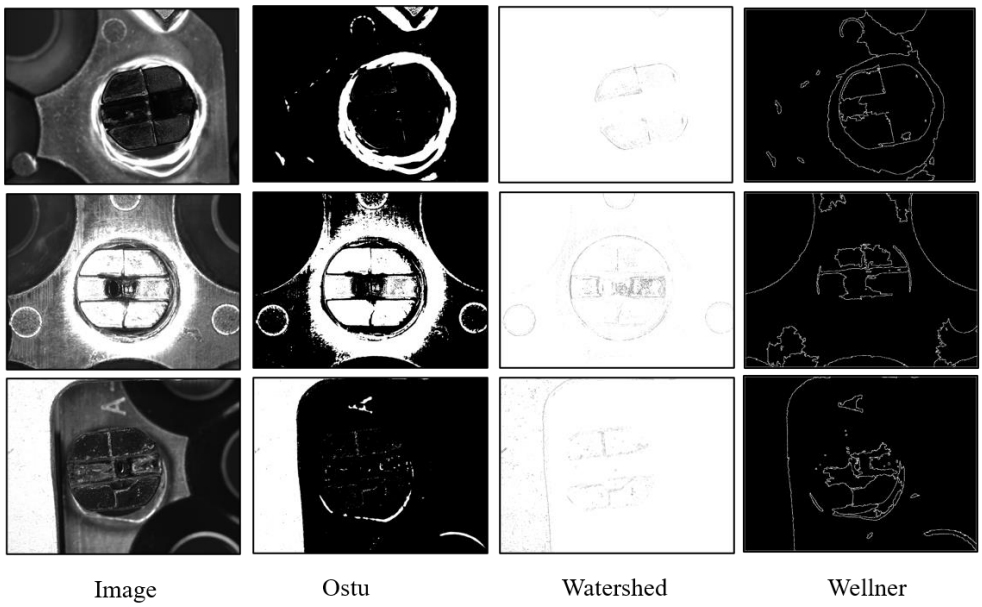

Figure 2. Experimental results of traditional lens buckle segmentation.

The results shown in Figure 2 indicate that traditional methods perform poorly in data extraction. In Otsu's method, although a rough outline of the target can be regressed, it is very difficult to find the target position and estimate the pose. The watershed method eliminates the influence of the background, but it is highly sensitive to noise; hence, much of the valuable target information is lost. Wellner's method treats parts of the background as foreground, resulting in a complete loss of the target information.

2.2. Deep Learning Detection Approaches

Deep learning [19] algorithms automatically learn the targets to be detected while avoiding the influence of human subjectivity and other noisy factors. Thus, they provide strong feature generalization. Nonlinear combinations are used to build convolutional neural network (CNN) architectures [20–22], and their capacities are controlled by varying the breadth and depth of features so that they can make strong and correct assumptions about the nature of an image [23]. Ge et al. [24] proposed a recognition method for two-dimensional (2D) instance segmentation and three-dimensional feature consistency pairing to assist in automatic workpiece painting. They used a mask region-based CNN (R-CNN) [25] to combine the fast segmentation and recognition of 2D workpieces with the strong discrimination of local details, using fast point feature histogram point cloud features to accurately distinguish dissimilar multi-view components and coarse-to-fine parts. Li et al. [26] proposed a method based on an improved "You Only Look Once" (YOLO) Version 3 (YOLOv3) real-time object detection algorithm [27] to identify workshop workpieces, wherein a deep separable convolution was used to improve the performance of the darknet backbone. However, its implementation was limited in its high-precision measurement capability, and it was considerably dependent on the accuracy of the regression detection framework. Li et al. [28] embedded an improved squeeze-and-excitation network (SENet) [29] module into a 50-layer convolutional residual network (ResNet50) [30] backbone and adopted a feature pyramid structure [31] to fuse dimensional features, which significantly improved the detection effect of vehicle-bottom parts. Because an improved

SENet module was added to each residual block, the complexity of the network increased, whereas the detection speed decreased. Faster R-CNN [32], YOLO [33], and SSD [34] are representative algorithms for deep learning object detection. These deep learning detection algorithms focus on the classification of the target; for the specific location of the target, they only need to give a rough candidate box. However, this study aims to identify the pose of the buckle, which needs to determine the deflection angle using a target detection algorithm that cannot estimate the rotation angle of the target.

3. Methods

This paper proposes a high-precision buckle pose estimation algorithm based on GAN that addresses the problems of large intra-class and small inter-class data differences. Figure 3 illustrates the proposed architecture, which consists of three components: a feature extraction module (encoder), a feature reconstruction module (decoder), and a refinement edge module (discriminator).

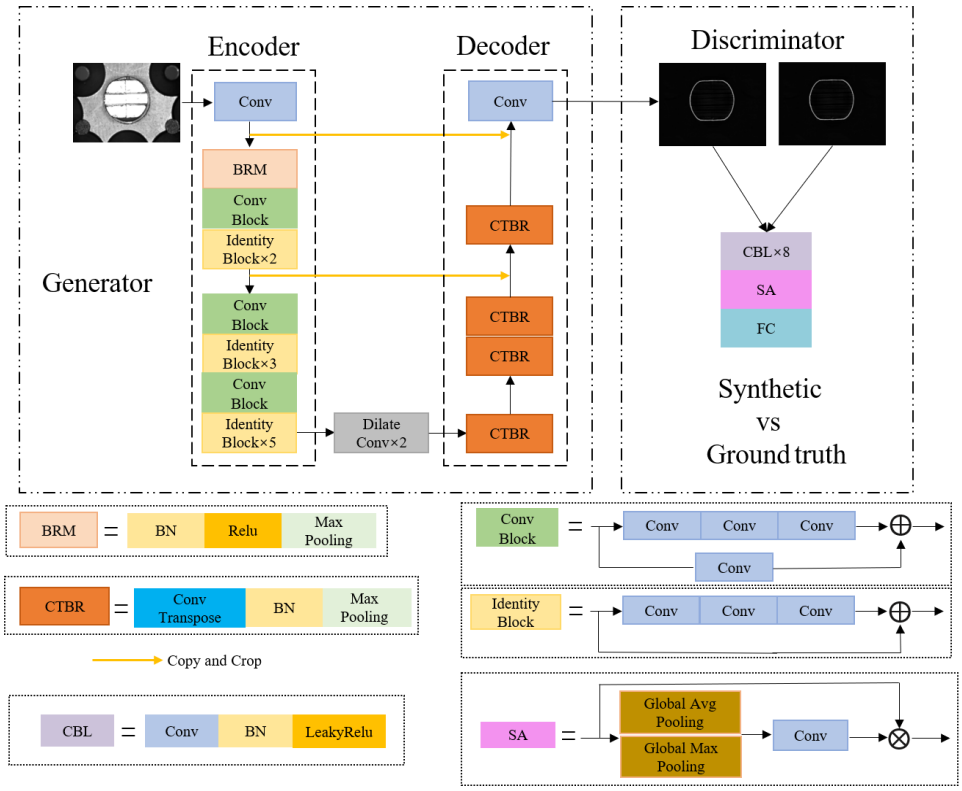

Figure 3. Generative adversarial network structure. The encoder is a feature extraction module, the decoder is a feature reconstruction module, and the discriminator serves as an auxiliary classifier to help the generator generate more realistic labels.

A batch of standard-sized original images and corresponding ground-truth labels are input into the network, and feature extraction is performed through convolution, pooling, and residual block operations in the encoder. The dilated convolution operation expands the receptive field of the network without increasing its depth, which is conducive to retaining spatial information. Thus, the network pays more attention to the overall characteristics of the target area. The decoder is used for feature reconstruction and skip connections are introduced in each layer of the encoder and decoder to combine different feature levels to compensate for the loss incurred by max-pooling. After obtaining the results of the feature reconstruction module, the channel is adjusted using a 1×1 convolution to obtain the final segmentation image. The model learns the effective features of each image in the training sample through forward propagation and calculates the loss by comparing the ground-truth label. A back-propagation algorithm is used to minimize generation

loss to optimize the network. During training, the generated prediction and ground-truth label images are input into the discriminator. Feature extraction is carried out through the convolution block, and feature screening is performed by the spatial attention module, which makes the discriminator pay more attention to the edge information, enhances the identification of the generated labels by the discriminator, and assists the generator in refining the edge effects.

3.1. Feature Extraction Module

CNNs [35] are widely used to extract the features of objects and are trained by stacking multiple convolution kernels and pooling layers. Our feature extraction network uses a ResNet50 backbone to extract the main features. As shown in Figure 4, the network consists of three blocks: a low layer that contains low-level features, a middle layer containing transition features, and a deep layer containing high-level features. First, a convolutional layer with a convolution block and two residual blocks is used to obtain the most original low-layer features. The middle-layer features are used to obtain contour and edge information. Finally, two convolution blocks and eight residual blocks are used to obtain deep semantic features.

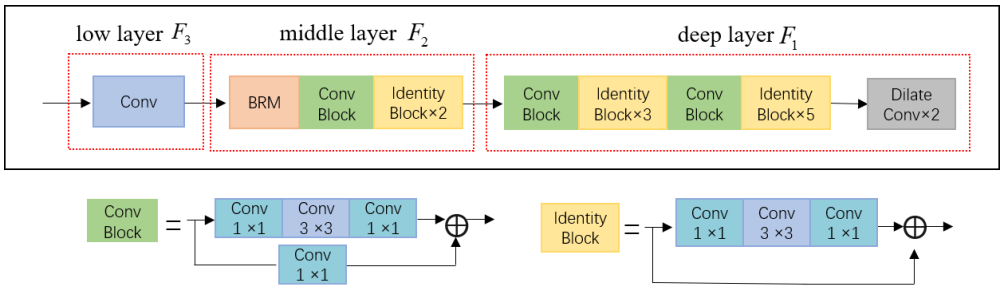

Figure 4. Feature extraction module. Through different convolution blocks, the features of different layers are obtained, which are low-level features F_3, mid-level features F_2, and deep features F_1.

An appropriate receptive field of deep features is key to the segmentation task. The receptive field size of each layer is calculated using Equation (1):

$$R(i, j-1) = (R(i, j) - 1) \times s + k. \quad (1)$$

This formula iterates from the uppermost to the lowest layer, where $R(i, j)$ represents the local receptive field of the ith to jth layer, s represents the step size, and k represents the size of the convolution kernel. The resolution of the training image is 1024 × 768, and the width of the target buckle accounts for ~40% of the entire image, which is approximately 400 pixels. Note that the ideal receptive field size cannot be lower than 400 pixels. However, the maximum receptive field of the feature extraction network is 267 pixels. The network cannot focus on the overall characteristics of the target after obtaining the deep features; however, using the pooling layer to improve the receptive field of the network loses part of the information. Therefore, dilated convolutions are added after the deep features to increase the receptive field of the network (see Table 1). The size of the receptive field in the final network is 427 pixels. To calculate the receptive field of the dilated convolution, the equivalent of the dilated convolution is first obtained, and the size of the receptive field is recalculated using the equivalent convolution. The conversion formula between the two is shown in Equation (2):

$$k' = k + (k-1) \times (d-1), \quad (2)$$

where k' represents the equivalent convolution, and d is the dilated ratios. In this study, dilated convolutions with void ratios of three and two and a convolution kernel size of three are used in correspondence to the convolution kernels with convolution kernel sizes of seven and five in the equivalent.

Table 1. Feature extraction network for each layer receptive field size.

Layer	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**
Name	Conv	Max-Pooling	Conv Block	Identity Block2	Conv Block	Identity Block3	Conv Block	Identity Block5	Dilate Conv1	Dilate Conv2
Dilation size	1	1	1	1	1	1	1	1	3	2
Kernel size	7×7	3×3	3×3	$(3 \times 3) \cdot 2$ *	3×3	$(3 \times 3) \cdot 3$ *	3×3	$(3 \times 3) \cdot 5$ *	3×3	3×3
Kernel receptive field	7×7	3×3	3×3	$(3 \times 3) \cdot 2$ *	3×3	$(3 \times 3) \cdot 3$ *	3×3	$(3 \times 3) \cdot 5$ *	7×7	5×5
Stride	2	2	1	1, 1	2	1, 1, 1	2	1, 1, 1, 1, 1	1	1
Receptive field	7	11	19	27, 35	43	59, 75, 91	107	139, 171, 203, 235, 267	363	427

$(k \times k) \cdot n$ * represents n convolution kernels of the same $k \times k$ size.

3.2. Feature Reconstruction Module

After obtaining the overall features of the target using dilated convolutions, the features are reconstructed by the decoder. The feature extraction network obtains three scale features that are recorded as deep features, F_1; middle layer features, F_2; and low features, F_3. As shown in Figure 5, the deep features are upsampled to each scale using deconvolutions, and the original scale features are fused by skip connections to obtain the enhanced feature, f_i ($i = 1, 2$), for each scale. The final segmentation image, R, is obtained by adjusting the channel dimension of the feature map through a 1×1 convolution, as shown in Equations (3)–(5):

$$f_1 = \sigma(BN(upsample(dilate(F_1)))), \tag{3}$$

$$f_{i+1} = \sigma(BN(upsample(W_{1\times1} * CAT(f_i, F_{i+1})))(i = 1, 2), \tag{4}$$

$$R = W_{1\times1} * \sigma(BN(W_{3\times3} * f_3)), \tag{5}$$

where σ is the rectified linear unit activation function, ReL; $upsample$ represents the deconvolution operation, $dilate$ represents the dilate convolution, CAT represents the channel fusion (splicing); $W_{1\times1}$ represents a 1×1 convolution kernel, which changes the channel dimension of the concatenated feature, and $W_{3\times3}$ represents a 3×3 convolution kernel.

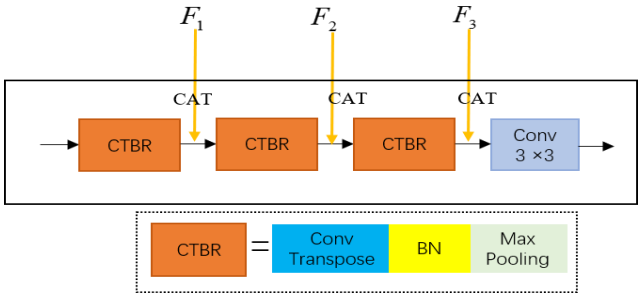

Figure 5. Feature reconstruction module. The features are upsampled through deconvolution blocks, and fused with F_1, F_2, and F_3 features extracted from features extraction through skip connections to obtain enhanced features.

Buckle pose estimation must regress the outer contour and the center edge line of the target. In each training sample, a positive sample pixel consists of the label position, whereas other positions are negative sample pixels. Therefore, the outer contour and the central edge line account for a small proportion of the overall pixels, which causes a serious imbalance between the positive and negative samples. We apply focal loss (FL) as the loss

function of the edge extraction network. *FL* balances the weights between positive and negative samples, including those of easy and difficult versions, as shown in Equation (6).

$$FL(p_t) = -\alpha_t(1-p_t)^{\gamma}\log(p_t)$$
$$\alpha_t = \begin{cases} \alpha & if\ y=1 \\ 1-\alpha & otherwise \end{cases}, \quad p_t = \begin{cases} p & if\ y=1 \\ 1-p & otherwise \end{cases}, \tag{6}$$

where p is the predicted value of each pixel, y is the true label of each pixel, α is the weight coefficient for controlling positive and negative samples, and γ is the weight coefficient for controlling difficult samples.

3.3. Refinement Edge Module

For high-precision buckle pose estimation, it is necessary to return the outer contour and center edge line closer to the ground-truth label. The feature reconstruction module is used to return the approximate pose of the target; however, some data samples obtain rough contour lines that reduce accuracy. Therefore, it is necessary to refine the regression results and add a discriminator to assist the generator in training. This discriminator is shown in Figure 6, which follows the concept of adversarial segmentation. The segmentation model, *G*, and the adversarial discriminator, *D*, are subjected to a minimax game. *G* aims to generate a label image to fool *D*, and *D* aims to distinguish the prediction image of *G* from the ground truth. Therefore, the discriminator is used to judge the quality of the edges generated by the edge extraction network, and the edge extraction network is used to generate better edges to deceive the discriminator. Therefore, edge refinement is accomplished. The mixture function of segmentation and discrimination losses is expressed as

$$L = FL(p_t) - \lambda\{-[zlnz' + (1-z)ln(1-z')]\}, \tag{7}$$

where the first term represents the original segmentation loss function of Equation (6), and the latter term is the loss of discriminator *D*, where z is a binary number indicating whether the input data contain the predicted (0) or ground-truth image (1). Thus, z' becomes the probability of the output data matching the predicted or ground-truth image.

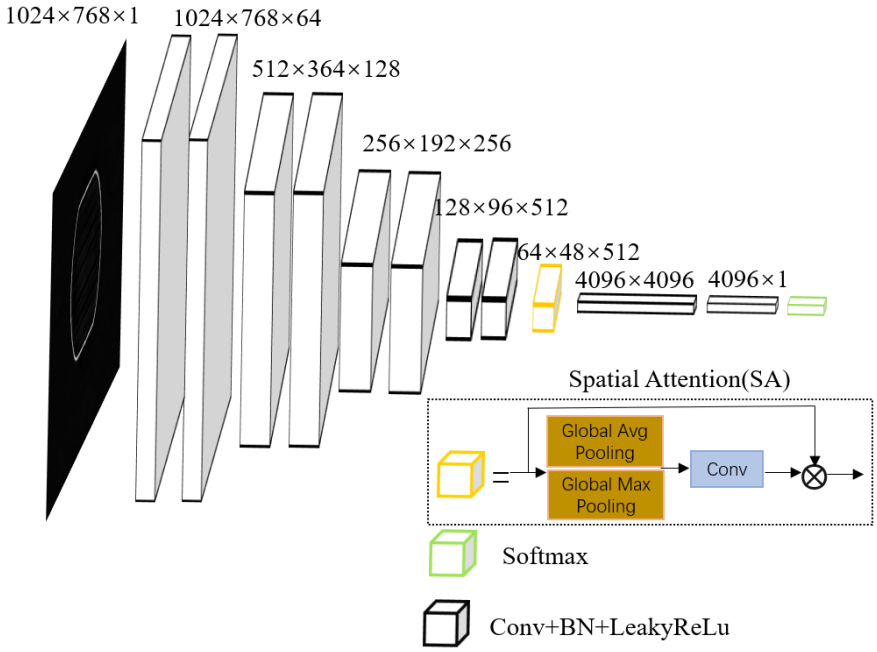

Figure 6. Refinement edge module. The generated result map is input into the discriminator, and the features are extracted through the convolution stack block. Spatial attention is used to pay more attention to the target area. Finally, the input probability is passed through the fully connected layer.

For the discriminator, D, convolutional block stacking is used to extract features, and the spatial attention mechanism is added after obtaining deep features so that the network can focus more on generating the contour label values. Average pooling encodes global statistics, while Max pooling encodes the salient parts. It is the same as the attention module proposed by [36], the feature map, S, is first obtained by stacking convolutional blocks. It is then averaged, and the maximum values of the channels in feature map S are calculated to obtain the average and maximum feature maps. The two maps are then channel-fused, and the channel is changed by a 1×1 convolution to obtain the attention map, M, which is multiplied by the feature map S pixel-by-pixel. The calculated final attention feature map, SA, is expressed as

$$M = W_{1\times1} * CAT(Ga(S), Gm(S)), \tag{8}$$

$$SA = \varphi(S) \otimes M, \tag{9}$$

where φ represents the sigmoid activation function, $\otimes$ represents pixel-by-pixel multiplication, Ga and Gm are the global average and maximum values of all channels in the feature map, respectively.

4. Experiments and Results

4.1. Datasets

The datasets used in this study were taken from real industrial production scenarios, and their resolutions were normalized to 1024×768. A total of 1572 real data items were used: 500 for training, 200 for validation, and 872 for testing.

4.2. Training Label

To meet the high-precision buckle pose estimation requirement, the training label was determined by analyzing the data of the outer contour and center edge line of the target buckle, as shown in Figure 7. The data were labeled by the labelme annotation tool, which mainly marks the outer contour and the center of the target. After model inferencing, the center point of the minimum circumscribed rectangle was considered the center position. The least square fitting slope was used as the relative horizontal deflection angle for the center edge line. The outer contour labels are edge and mask labels, respectively. Based on the following experimental comparison, the edge label was superior to the mask label.

(**a**) (**b**) (**c**) (**d**)

Figure 7. Different labels during training: (**a**) image; (**b**) edge label; (**c**) mask label; (**d**) line label.

4.3. Experimental Environment

In this study, we used an NVIDIA 2080Ti graphics card with 11-GB memory for the PyTorch framework in the Ubuntu system environment for this experiment. The Adam optimizer was used for generator training, with a learning rate of 0.001 for 40 training epochs. The RMSprop optimizer was used as the discriminator, and to make its learning process slower than that of the generator, its learning rate was set to half that of the generator (0.0005). The first five epochs only trained the generator, and when a certain level of generation ability was found, the discriminator was trained.

4.4. Evaluation Methodology

To evaluate the performance of our algorithm, the evaluation metric for edge detection was used to measure the quality of the results, and the absolute distance of the error

was used as the evaluation metric for the final result. For the edge extraction network, the optimal dataset scale (ODS), optimal image scale (OIS), average precision (*AP*), and R50 measures were used, and the *F*-measure was the reconciled average of precision (*P*) and recall (*R*). ODS represents the global optimal threshold, which is fixed in all images to maximize the overall *F*-measure. OIS represents the optimal threshold of each image and was used to maximize the *F*-measure of the image. AP is the integral of the *P*/*R* curve, and R50 is the recall rate at 50% precision. For edge detection, a distance tolerance parameter was used to determine whether the predicted boundary pixels were correctly predicted, allowing for small positioning errors between the predicted and ground-truth boundaries. The distance tolerance was obtained by multiplying the width and height of the image by the maxDist (0.0075). The formulae for *P* and *R* are shown in Equations (10) and (11); the formulae for the *F*-measure and AP are shown in Equations (12) and (13).

$$P = nnz(matchE)/nnz(E1), \tag{10}$$

$$R = sum(matchG)/sum(allG), \tag{11}$$

$$F = \left(1+\beta^2\right) \times \frac{P \times R}{\beta^2 \times P + R}, \tag{12}$$

$$AP = \int_0^1 P(R)dR, \tag{13}$$

where $E1$ represents the binarization result of all edge images, *matchE* is the predicted edge point that matches the ground-truth point, *allG* is the ground-truth edge point, and *matchG* is the number of predicted edge points that were ground true. The number of non-zeros (*nnz*) reflects the number of non-zero elements and *sum* represents the sum of the points. The F-measure was tuned by adjusting the value of β to obtain the optimized proportion of *P* and *R*, where $\beta = 1$.

The evaluation indices of the center point and angle prediction were evaluated using the Euclidean distance and absolute value error, respectively. The calculation formulae are as follows:

$$\rho = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}, \tag{14}$$

$$\theta = |\alpha - \alpha_0|, \tag{15}$$

where (x_1, y_1) and (x_2, y_2) represent the center point coordinates of the true value and those of the predicted value, respectively. α and α_0 represent the deflection angles of the true and predicted values, respectively.

4.5. Experimental Results and Analysis

The two given labels were evaluated, and the compared state-of-the-art contour regression algorithms included Hed [37], richer convolutional features for edge detection [38], DexiNed [39], and PidiNet [40]. First, the label results, whose parameters were consistent during training, were evaluated; only the training label style was changed. From Figure 8, it can be seen that the maximum error of the center distance of the edge label is in the range of seven-to-eight pixels, whereas the maximum error of the center distance of the mask label is more than eight pixels. Because the machine is clumsy, it is possible to damage the lens when the error exceeds eight pixels.

We selected data from the mask and edge labels of the resulting image, as shown in Figure 9. Then, the segmentation image generated by the mask label forced the network to classify the surrounding background pixels into positive values because all pixels are classified during training. Owing to the small differences between the background and target classes, the mask label caused the network to learn useless pixel information, which resulted in errors. When the edge label was used, the overall target shape was not affected and its center position could be estimated using the circumscribed rectangle, although

there was a local missing part of the regression outer contour. Based on the following experimental comparison, the edge label is superior to the mask label.

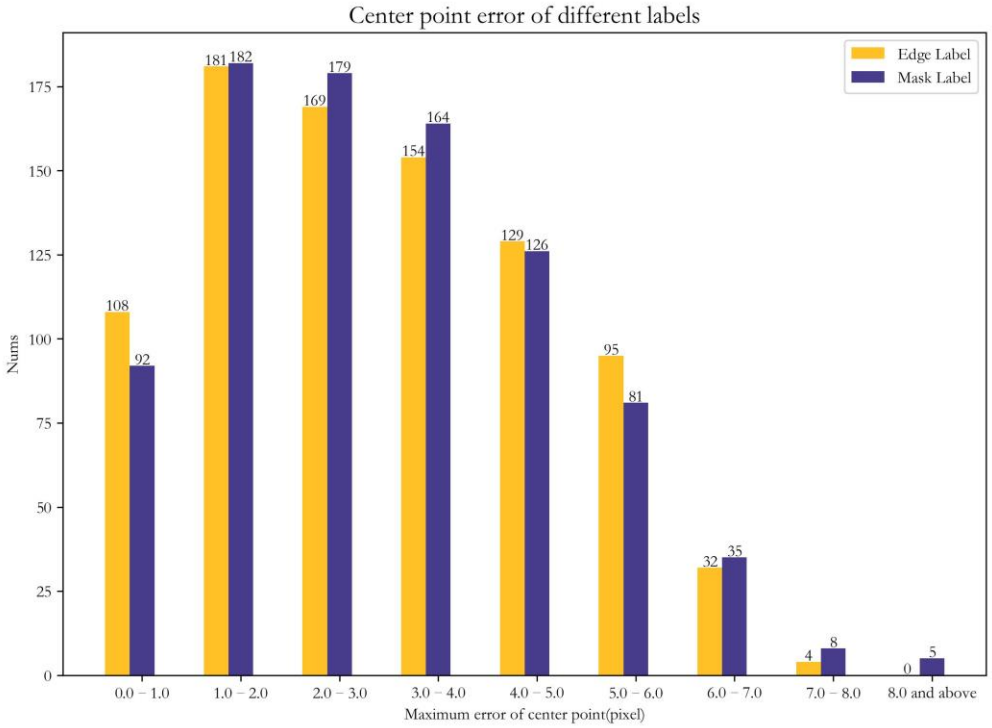

Figure 8. Statistical chart of center distance error for different labels.

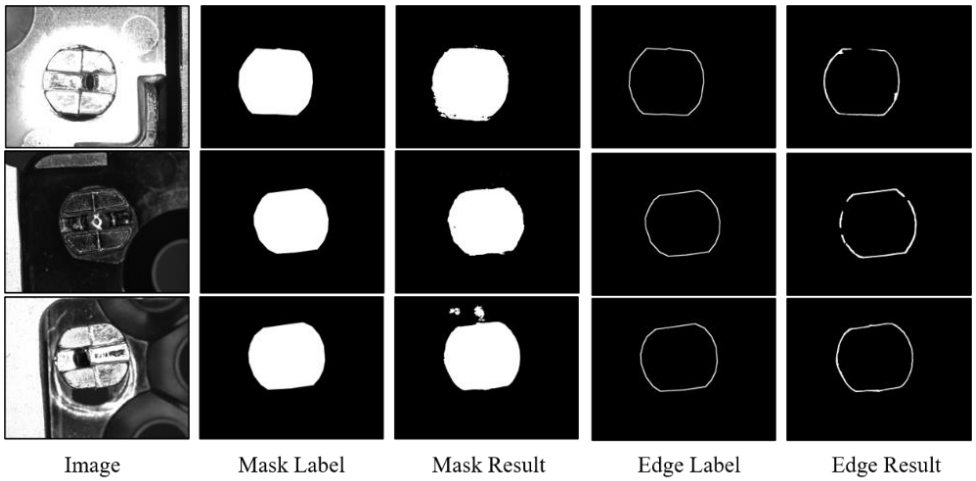

Figure 9. Part of the results of different labels.

We verified the advantages of our contour regression method and compared it to the other methods. As shown in Table 2, the ODS, AP, and R50 of this method are higher than other methods, and only 0.1% lower than the best method under OIS.

Although the results of all methods were not significantly different from the overall indicators, they did not perform well for specific difficult samples. As shown in Figure 10, the background interferences of some samples were quite significant, and the characteristics of the target areas were insignificant. The proposed method was superior to the other methods in terms of the contour and line regression results. Notably, other methods produced incomplete or inaccurate contours and line regressions, resulting in inaccurate results. The method proposed in this study can effectively regress contours and lines better than the other methods.

Table 2. Evaluation indexes of different methods.

Method	ODS	OIS	AP	R50
Hed	0.9720	0.9835	0.9680	0.9775
RCF	0.9725	0.9860	0.9750	0.9885
DexiNed	0.9660	0.9735	0.9400	0.9905
PidiNet	0.9250	0.9540	0.9580	0.9960
ours	0.9740	0.9850	0.9775	0.9970

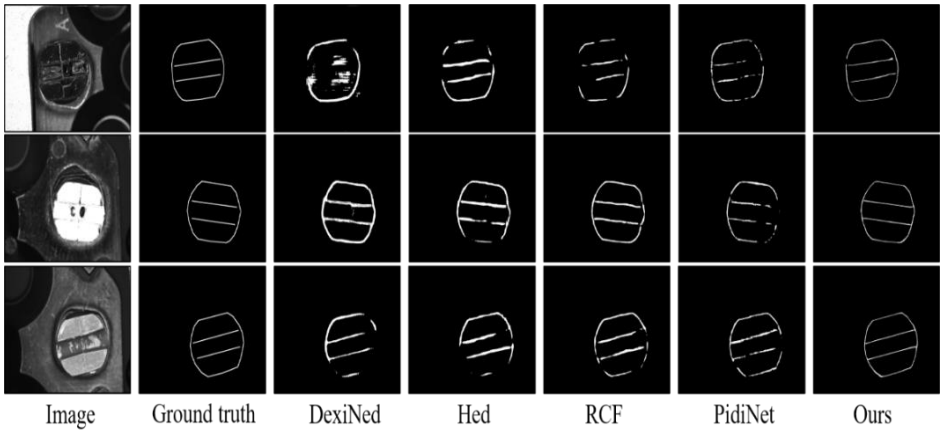

Figure 10. Results of different methods.

To show the advantages of our method more clearly, we used the center point pixel distance differences of under seven pixels and angle absolute errors below 1.5 as correct sample estimations to recalculate accuracy. As shown in Table 3, the center point accuracy of the proposed method reached 99.5%, and the angle estimation accuracy reached 99.3%, which was higher than the other methods. In all samples, the maximum error of the center point pixel distance deviation was 7.36 pixels, and the maximum error of the angle absolute error was 1.98°. Hence, the results met the three requirements pointed out in the introduction.

Table 3. Accuracy and error values of different methods.

Method	Center Point Accuracy	Maximum Error of Center Point (Pixel)	Angle Accuracy	Maximum Angle Error (Degree)
Hed	94.7%	18.27	97.2%	2.86
RCF	96.5%	12.68	98.4%	2.93
DexiNed	93.6%	13.45	92.7%	4.99
PidiNet	95.2%	14.31	96.4%	3.56
ours	99.5%	7.36	99.3%	1.98

4.6. Ablative Study

To evaluate the impact of each module on overall performance, ablation experiments were conducted using the same training parameters and datasets for the cavity convolution and refinement edge modules. The refinement edge module was used to assist in segmentation network training and the inference phase was not used. As shown in Table 4, the accuracy of the center point estimation increased from 93.2 to 98.7% after adding the dilated convolutions to expand the network receptive field, and the accuracy of angle estimation increased from 93.0 to 98.6%. When using the refinement edge module of GAN to assist generator training, the network better refined the contours of the generated images. The accuracy of central point estimation increased by 0.8% and that of angle estimation increased by 0.7%.

Table 4. Comparison of ablation experiments of different modules.

Method	Center Point Accuracy	Angle Accuracy
Edge Network	93.2%	93.0%
Dilate	98.7%	98.6%
Refinement Edge	99.5%	99.3%

The improvements highlighted samples with poor regression effects regarding the edge extraction network. The regression of contours and lines was more refined to be closer to the actual values, and the effect image and difficult sample result image were refined. As shown in Figure 11, when only the edge extraction network is used to regress the target, a double contour phenomenon may occur, which affects the regression results. When dilated, the convolution is added to increase the receptive field, whereas the double contour and shape line regression phenomena are reduced to a certain extent. However, the regression contour was rougher. Again, leveraging the GAN method, the discriminator and generator confrontation training was added to distinguish the authenticity of the generated image; thus, the double contour phenomenon disappeared.

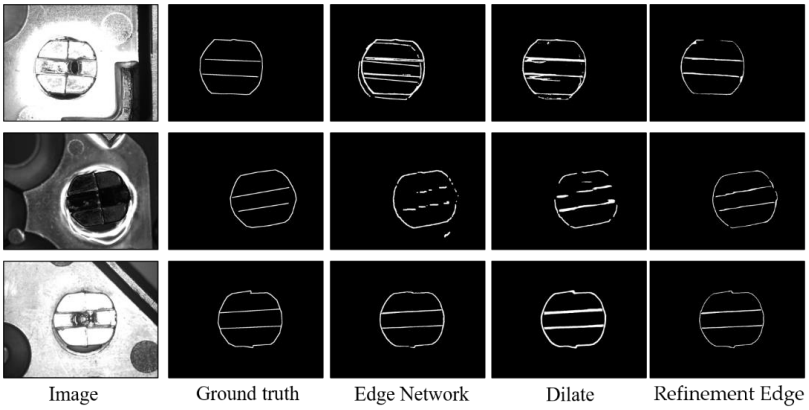

Figure 11. Comparison of the results of ablation experiments.

After obtaining the outer contour and the center edge line of the target using the proposed method, the minimum circumscribed rectangle of the outer contour is estimated, and the center point of the circumscribed rectangle is taken as the center point of the target location. The center edge line is fitted using the least square method, and the slope obtained is used as the rotation angle of the target relative to the horizontal position. Some of the final results are shown in Figure 12.

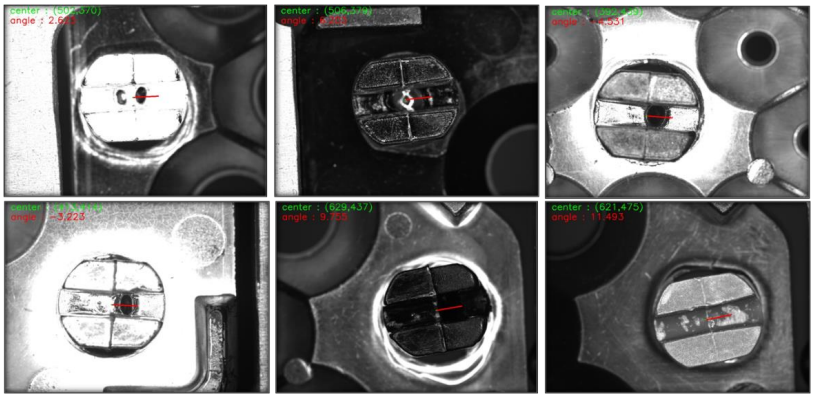

Figure 12. Final result image (the center point and the rotation angle are obtained by the least circumscribed rectangle and the least square method for the outer contour and the center edge line, respectively).

5. Conclusions

In this study, aiming at the high-precision pose estimation requirements of buckles under large background interference and complex data, a GAN-based buckle pose estimation algorithm was proposed. The algorithm uses dilated convolutions to enable the network to focus on the overall target characteristics. A discriminator and spatial attention module were added for edge refinement. Problems of incomplete and low-accuracy regressions of snap contours and lines were solved, thereby improving the accuracy of the overall pose estimation. Finally, the external matrix and least squares methods were used to estimate the center position and deflection angle of the target, respectively. The maximum error of the center point distance of the test set was 7.36 pixels, and the absolute maximum error of the angle was 1.98°. The inferencing speed of the code deployed on an industrial computer equipped with NVIDIA 2080Ti was approximately 30 ms per frame, which meets real-time requirements and accelerates the production efficiency of the lens coating. For some samples (the first row in Figure 11), the regression results of our method were missing. In the future, we plan to use a more advanced network to optimize our pose estimation technique while also extending its generalizability to other industrial parts.

Author Contributions: Conceptualization, H.F.; methodology, X.C.; software, X.C.; validation, K.S. and J.X.; investigation, H.F.; resources, X.C.; writing—original draft preparation, H.F. and S.Y.; writing—review and editing, X.C.; supervision, S.Y.; project administration, K.S. and J.X.; funding acquisition, J.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research is supported by Zhejiang Provincial Natural Science Foundation (LQ23F050006), Technology Innovation 2025 Major Project (2020Z019, 2021Z063, 2021Z126), Ningbo Medical Science and Technology Plan Project (2020Y34) and Ningbo Science and Technology Program for the Public Interest (2022S078).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data presented in this paper are available on request from the author.

Acknowledgments: The authors would like to thank all the participants who were involved in the experiments. The authors would also like to thank Ningbo Cstar Precision Machine Co., Ltd., Ningbo, China for their support with the data.

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

Abbreviations

GAN	Generative adversarial network
SVM	Support vector machine
CNN	Convolutional neural network
2D	Two-dimensional
R-CNN	Region-based CNN
SENet	Squeeze and excitation network
FL	Focal loss
SA	Spatial attention
ODS	Optimal dataset scale
OIS	Optimal image scale
AP	Average precision

References

1. Hu, J.; Liu, S.; Liu, J.; Wang, Z.; Huang, H. Pipe pose estimation based on machine vision. *Measurement* **2021**, *182*, 109585. [CrossRef]
2. Bai, X.; Fang, Y.; Lin, W.; Wang, L.; Ju, B.-F. Saliency-Based Defect Detection in Industrial Images by Using Phase Spectrum. *IEEE Trans. Ind. Inform.* **2014**, *10*, 2135–2145. [CrossRef]

3. Ian, G.; Jean, P.-A.; Mirza, M.; Bing, X.; David, W.-F.; Sherjil, O.; Aaron, C.; Yoshua, B. Generative Adversarial Nets. *Adv. Neural Inf. Process. Syst.* **2014**, *2*, 2672–2680.
4. Yu, F.; Koltun, V. Multi-Scale Context Aggregation by Dilated Convolutions. In Proceedings of the ICLR, San Juan, Puerto Rico, 2–4 May 2016.
5. Huang, G.; Chen, J.; Liu, L. One-Class SVM Model-Based Tunnel Personnel Safety Detection Technology. *Appl. Sci.* **2023**, *13*, 1734. [CrossRef]
6. Zhao, F.; Xu, L.; Lv, L.; Zhang, Y. Wheat Ear Detection Algorithm Based on Improved YOLOv4. *Appl. Sci.* **2022**, *12*, 12195. [CrossRef]
7. Hwang, B.; Lee, S.; Han, H. DLMFCOS: Efficient Dual-Path Lightweight Module for Fully Convolutional Object Detection. *Appl. Sci.* **2023**, *13*, 1841. [CrossRef]
8. Viola, P.; Jones, M. Rapid Object Detection Using a Boosted Cascade of Simple Features. In Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Kauai, HI, USA, 8–14 December 2001; Volume 1, pp. I–I.
9. Gonzalez, R.C.; Woods, R.E. Wavelets and Multiresolution Processing. In *Digital Image Processing*, 3rd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2007; Volume 7, pp. 461–521.
10. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the Computer Vision and Pattern Recognition, San Diego, CA, USA, 20–26 June 2005; pp. 886–893. [CrossRef]
11. Felzenszwalb, P.F.; Girshick, R.B.; McAllester, D.; Ramanan, D. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. Pattern Anal. Mach. Intell.* **2009**, *32*, 1627–1645. [CrossRef]
12. Hearst, M.A.; Dumais, S.T.; Osuna, E.; Platt, J.; Scholkopf, B. Support Vector Machines. *IEEE Intell. Syst. Their Appl.* **1998**, *13*, 18–28. [CrossRef]
13. Yu, W.X.; Xu, G.L. Connector Surface Crack Detection Method. *Laser Optoelectron. Prog.* **2022**, *59*, 1415015.
14. Shan, Z.; Xin, M.; Di, W. Machine Vision Measurement Method of Tooth Pitch Based on Gear Local Image. *J. Sci. Instrum.* **2018**, *39*, 7.
15. Hongjian, Z.; Ping, H.; Xudong, Y. Fault Detection of Train Center Plate Bolts Loss Using Modified LBP and Optimization Algorithm. *Open Autom. Control Syst. J.* **2015**, *7*, 1916–1921. [CrossRef]
16. Otsu, N. A threshold selection method from gray-level histograms. *IEEE Trans. Syst. Man Cybern.* **1979**, *9*, 62–66. [CrossRef]
17. Wellner, P. Interacting with paper on the DigitalDesk. *Commun. ACM* **1993**, *36*, 87–96. [CrossRef]
18. Beucher, S.; Lantuejoul, C. Use of Watersheds in Contour Detection. In Proceedings of the International Workshop on Image Processing: Real-Time Edge and Motion Detection/Estimation, Rennes, France, 17–21 September 1979.
19. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
20. Jarrett, K.; Kavukcuoglu, K.; Ranzato, M.A.; LeCun, Y. What is the best multi-stage architecture for object recognition? In Proceedings of the 12th International Conference on Computer Vision Workshops, Kyoto, Japan, 27 September–4 October 2009; pp. 2146–2153. [CrossRef]
21. Turaga, S.C.; Murray, J.F.; Jain, V.; Roth, F.; Helmstaedter, M.; Briggman, K.; Denk, W.; Seung, H.S. Convolutional Networks Can Learn to Generate Affinity Graphs for Image Segmentation. *Neural Comput.* **2010**, *22*, 511–538. [CrossRef]
22. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [CrossRef] [PubMed]
23. Dong, H.; Song, K.; He, Y.; Xu, J.; Yan, Y.; Meng, Q. PGA-Net: Pyramid Feature Fusion and Global Context Attention Network for Automated Surface Defect Detection. *IEEE Trans. Ind. Inform.* **2019**, *16*, 7448–7458. [CrossRef]
24. Ge, J.H.; Wang, J.; Peng, Y.P.; Li, J.; Xiao, C.; Liu, Y. Recognition Method for Spray-Painted Workpieces Based on Mask R-CNN and Fast Point Feature Histogram Feature Pairing. *Laser Optoelectron. Prog.* **2022**, *59*, 1415016.
25. He, K.; Gkioxari, G.; Dollar, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017.
26. Li, J.X.; Qiu, D.; Yang, H.T.; Liu, K. Research on Workpiece Recognition Method Based on Improved YOLOv3. *Modular Mach. Tool Autom. Manuf. Tech.* **2020**, *8*, 92–96+100.
27. Redmon, J.; Yolov, F.A. An Incremental Improvement. *arXiv* **2018**, arXiv:1804.02767.
28. Li, L.R.; Wang, Z.Y.; Zhang, K.; Yang, D.C.; Xiong, W.; Gong, P.C. Detection Algorithm of Train Bottom Parts Based on OSE-dResnet Network. *Comput. Eng. Sci.* **2022**, *44*, 692–698.
29. Hu, J.; Shen, L.; Albanie, S. Squeeze-and-Excitation Networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 7132–7141.
30. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the of the 2016 Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 770–778.
31. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Hariharan, B.; Belongie, S. Feature Pyramid Networks for Object Detection. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 936–944.
32. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster r-cnn: Towards Real-Time Object Detection with Region Proposal Networks. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [CrossRef]
33. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 26 June–1 July 2016; pp. 779–788.

34. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot Multibox Detector. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; Springer: Cham, Switzerland, 2016; pp. 21–37.
35. Khan, A.; Chefranov, A.; Demirel, H. Image Scene Geometry Recognition Using Low-Level Features Fusion at Multi-layer Deep CNN. *Neurocomputing* **2021**, *440*, 111–126. [CrossRef]
36. Woo, S.; Park, J.; Lee, J.Y.; Kweon, I.S. Cbam: Convolutional block attention module. In Proceedings of thes European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 3–19.
37. Xie, S.; Tu, Z. Holistically Nested Edge Detection. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 11–18 December 2015; pp. 1395–1403.
38. Liu, Y.; Cheng, M.M.; Hu, X.; Wang, K.; Bai, X. Richer Convolutional Features for Edge Detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3000–3009.
39. Soria, X.; Sappa, A.; Humanante, P.; Akbarinia, A. Dense Extreme Inception Network for Edge Detection. *Pattern Recognit.* **2023**, *139*, 109461. [CrossRef]
40. Su, Z.; Liu, W.; Yu, Z.; Hu, D.; Liao, Q.; Tian, Q.; Liu, L. Pixel Difference Networks for Efficient Edge Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 11–17 October 2021; pp. 5117–5127.