electronics

MDPI

Article

Improved Intelligent Image Segmentation Algorithm for Mechanical Sensors in Industrial IoT: A Joint Learning Approach

Xin Xie [1,*], Tiancheng Wan [1], Bin Wang [1], Tijian Cai [1], Ao Yu [2], Mohamed Cheriet [2] and Fengping Hu [3]

1 School of Information Engineering, East China Jiaotong University, Nanchang 330013, China; 2018068081203001@ecjtu.edu.cn (T.W.); 2018068081203002@ecjtu.edu.cn (B.W.); cai2017@ecjtu.edu.cn (T.C.)
2 École de Technologie supéRieure, Montréal, QC H3C1K3, Canada; yuao@bupt.edu.cn (A.Y.); mohamed.cheriet@etsmtl.ca (M.C.)
3 School of Civil Engineering and Architecture, East China Jiaotong University, Nanchang 330013, China; hufengping1968@126.com
* Correspondence: xiexin@ecjtu.edu.cn

Abstract: The industrial Internet of Things (IoT) can monitor production in real-time by collecting the status of parts on the production line with cameras. It is easy to have bright and dark areas in the same image because of the smooth surfaces of mechanical parts and the unstable light source, which affects semantic segmentation's performance. This paper proposes a joint learning method to eliminate the influence of illumination on semantic segmentation. Semantic image segmentation and image decomposition are jointly trained in the same model, and the reflectance image is used to guide the semantic segmentation task without the illumination component. Moreover, this paper adopts an enhanced convolution kernel to improve the pixel accuracy and BN fusion to enhance the inference speed, optimizing the model to meet real-time detection needs. In the experiments, a dataset of real gear parts was collected from industrial IoT cameras. The experimental results show that the proposed joint learning approach outperforms the state-of-the-art methods in the task of edge mechanical part detection, with about 4% pixel accuracy improvement.

Keywords: industrial IoT; joint learning; semantic segmentation; asymmetric convolution; BN fusion

check for updates

Citation: Xie, X.; Wan, T.; Wang, B.; Cai, Y.; Yu, A.; Cheriet, M.; Hu, F. Improved Intelligent Image Segmentation Algorithm for Mechanical Sensors in Industrial IoT: A Joint Learning Approach. *Electronics* **2021**, *10*, 446. https://doi.org/10.3390/electronics10040446

Academic Editor: Jaime Lloret Mauri
Received: 27 January 2021
Accepted: 8 February 2021
Published: 11 February 2021

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

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

With the development of Industry 4.0, promoting the combination of the Internet of Things (IoT) and modern manufacturing is of great significance to promoting industrial production modernization [1]. Common mechanical parts, such as gears and slender shafts, are widely used in the military, aerospace, automobile and manufacturing industries. The precision of parts directly affects the equipment's working performance and service life. Therefore, combined with the industrial IoT, a large number of intelligent cameras can be used to collect the status of parts on the production line in real-time. Each step of the production process can be identified, monitored and managed, which can significantly improve the yield rate of factory parts [2].

Using an intelligent camera, the size of mechanical parts can be measured with the non-contact method, in which the edge detection algorithm of the part image is crucial. Xin et al. [3] used the improved Roberts operator to extract the contour of the target, and then Zernike moments were used for sub-pixel positioning. At the same time, they were using the Otsu method to automatically select the segmentation threshold, which achieves good detection efficiency and detection accuracy. Ofir N et al. [4] regarded edge detection as a group of discrete curves to search for faint edges with noise interference, and effectively detect these faint edges. These traditional methods mostly extract the edge by analyzing the shape, texture, color and other features of the target image [5], then calculating the parts' size. Generally, to build a model for specific applications, we need to be familiar with the

process of edge detection and rely on the manually designed extractor for feature extraction. Besides, we also need to have professional knowledge and parameter adjustment process, which cannot be widely applied.

Industrial automatic production equipment is characterized by large scale and complicated process. Mostly distributed intelligent cameras collect different types of images, so classical algorithms are difficult to meet the actual needs. The development of deep learning theory and practice provides a useful reference for image edge detection and segmentation, and semantic segmentation is a challenging problem [6]. The sensor performance of the IoT camera is limited, and the image quality is low due to the movement of the parts and the specular reflection of the metal surface. The imaging conditions changes may hurt the segmentation process, including shadow, reflection, light source color and intensity.

For IoT devices with limited cost and performance, Sharma et al. [7] used Refined Graph Cut Segmenter to design an improved image segmentation technology for low-resolution images and restricted devices. They applied the algorithm to low-end IoT servers. Khan et al. [8] proposed a novel cascade method, which combines hand-made features with convolutional neural network (CNN) to process brain tumor image segmentation tasks generated by IoT devices. At present, there are few types of research on image segmentation algorithms in the field of industrial production, and the existing models are not ideal for image processing with large differences between light and shade. Thus, the state-of-the-art approaches are hard to meet the detection requirements of actual production in industrial IoT.

Industrial IoT camera is greatly affected by illumination factors when collecting images, so this paper's method improves the semantic segmentation effect by excluding illumination. Intrinsic image decomposition is the process of decomposing an image into reflectance components and illumination components. The reflectivity component is an inherent property of the object and will not change with light factors. Illumination components change continuously with light source factors, including specular reflection and shadows, that affect image semantic segmentation. Therefore, it is more effective to use the reflectance image for semantic segmentation because it does not contain the negative effects of illumination. On the contrary, semantic segmentation information has prior knowledge of object reflectivity, which can guide the intrinsic image decomposition.

In this paper, a convolutional neural network with an encoder-decoder structure is proposed. After extracting image features through one encoder, two decoders are used to process image segmentation and image decomposition tasks, respectively. Regarding segmentation and decomposition as a mutually-promoting collaborative process, the model is trained with a joint learning strategy to eliminate the influence of light to improve mechanical parts segmentation accuracy. The contributions of this paper are as follows:

(1) Design a semantic segmentation model for parts image, use the joint learning method to improve the semantic segmentation performance;
(2) Create an image dataset of mechanical parts collected by IoT camera with labels;
(3) Use asymmetric convolution and BN fusion to optimize the model performance.

The remaining portions of this paper are organized as follows: the related work and principles are stated in Section 2; the method part is discussed in Section 3; and the experiments and results are discussed in Section 4. Finally, the conclusion of this paper is presented in Section 5.

2. Related Work

This section briefly introduces the progress of multi-task learning, and explains the principle of the two tasks to be jointly processed in this paper.

2.1. Multi-Task Learning

Multi-Task learning is a method that combines multiple tasks and learns simultaneously to enhance the ability of model representation and generalization. Joint learning can

be realized through the neural network approach. The main work can currently be divided into Parameter Sharing [9] and Tagging strategy [10].

As shown in Figure 1, there are three existing parameter sharing schemes: Hard sharing [11], Soft sharing [12], and hierarchical sharing [13]. Hard Sharing is the most widely used sharing mechanism today, embedding data representations of multiple tasks into the same semantic space, and then extracting feature representations for each task using a specific layer. Hard sharing is easy to implement and is suitable for tasks with strong correlation, but it often performs poorly when encountering weakly related tasks. For the soft sharing, each task uses a single network for learning, and the network of each task can access information in the network corresponding to other tasks, such as eigenvalues, gradients and so on. Although the soft sharing mechanism is very flexible and does not need any assumptions about task dependencies, additional parameters are required for assigning each task a network. Hierarchical sharing is to do the simple tasks in shallow layers of the network and complex tasks in deep layers. Hierarchical sharing is more flexible than hard sharing and requires fewer parameters than soft sharing. However, designing an efficient hierarchical structure for multiple tasks is mainly relies on expert experience.

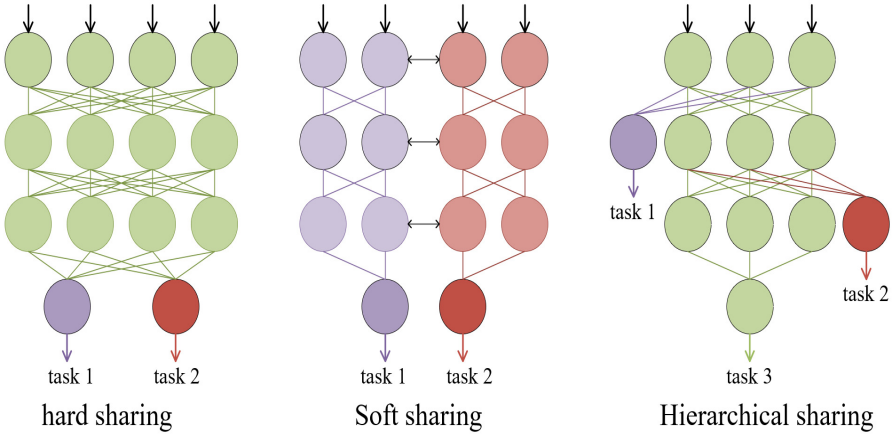

Figure 1. Parameter sharing.

Image segmentation is based on the object's category information, and the category information contains the prior knowledge of the object's reflectivity attributes, so segmentation and decomposition can be used as a vital correlation task. In this paper, a hard sharing mechanism is used to extract features with one encoder, and two decoders handle the corresponding tasks separately. Furthermore, the soft sharing is also combined between decoders to improve each task's performance by sharing parameters.

2.2. Image Semantic Segmentation

Image segmentation is a process of assigning a label to each pixel in the image such that pixels with the same label are connected for some visual or semantic property [6], then divide into a different region. Semantic segmentation technology based on deep learning has dramatically improved the performance of image edge detection. Long J et al. [14] proposed a full convolutional neural network (FCN), which replaces the full connection layer with the convolution layer in CNN, retains the target's spatial information in the output and realizes semantic segmentation by pixel-level classification. Benefit from rich spatial information and large perception domain [15], many classic models such as U-Net [16], Mask R-CNN [17], Deeplab [18], etc., were proposed based on FCN. Most of these models use public datasets such as ImageNet [19], to show the model's performance. According to the proposed model, researchers use some improvements and optimizations to satisfy the specific scenarios. To improve the segmentation performance, Stan T et al. [20] sampled a large number of small images in a fixed number of X-ray datasets to train the neural network. Smith A et al. [21] designed a U-Net-based convolutional neural network

and constructed an annotated chicory dataset, which successfully completes the root system segmentation task of plants. It also demonstrated the feasibility of using deep learning to create the own dataset. Vuola A et al. [22] compared the advantages and disadvantages between U-Net and Mask R-CNN and developed an integrated model, which achieved better results in the nuclear segmentation task.

In mechanical parts image segmentation in the industrial scene, the change of lighting conditions will cause the change of object appearance when the image is collected in the field, which hurts the semantic segmentation task. In this paper, the segmentation and the decomposition tasks are learned jointly to reduce the impact of illumination.

2.3. Intrinsic Image Decomposition

An image can be decomposed to generate countless combinations of reflectance and illumination, so image decomposition is a long-standing ill-posed problem [23]. Li et al. [24] added non-local texture constraints to traditional techniques to optimize intrinsic image decomposition, significantly improving previous algorithms. With the development of technology, the latest research on intrinsic image decomposition has turned to deep learning technology. Shi et al. [25] used a neural network decoder to jointly optimize each component by learning the correlation between intrinsic attributes, and achieved robust and real decomposition results. Based on this study, we use image segmentation attributes as an assistant to improve the performance of other tasks by joint learning.

3. Method

To reduce the influence of illumination factors on parts image segmentation, we use the joint learning method in the encoder-decoder model. A shared encoder is used to extract features, and two decoders are used to learn image segmentation and image decomposition, respectively. Through intrinsic image decomposition, the reflectance image without illumination component is used to guide the semantic segmentation task; simultaneously, the class attribute provided by semantic segmentation contains the prior knowledge of reflectance of the target object, which guides the image decomposition task. Besides, asymmetric convolution and BN fusion enhance the feature learning ability and accelerate the operation speed, respectively.

3.1. Joint Learning Method

Image intrinsic decomposition is based on Retinex theory [26], which decomposes an image into the product of a reflectance image and illumination image. The results obtained by image decomposition are not unique, and most of the current work is devoted to solving the ill-posed problem of intrinsic image decomposition. Suppose that I is the original image, R is the reflectance image, S is the illumination image and (x, y) is the image's pixel coordinates. The classical intrinsic image decomposition can be formulated as:

$$I(x, y) = R(x, y)S(x, y) \tag{1}$$

Since the reflectance image is an object's own property, it is not affected by any light. According to the ShapeNet [25] model, given an image I, the process of obtaining reflectance component R and illumination component S by intrinsic image decomposition can be understood as follows:

$$(R, S) = F(I, \theta) \tag{2}$$

θ contains all the parameters learned by the image intrinsic decomposition decoder. MSE is used to optimize each component of θ. Let R^* to be the ground truth parameter in the dataset, R is the parameter learned by the decoder and $r = R^* - R$ is the learning difference. To obtain the most realistic reflectance image, minimize the following formula:

$$L_1(R^*, R) = \frac{1}{n}\sum_{i,j,c} r_{i,j,c}^2 - \frac{1}{2n^2}\left(\sum_{i,j,c} r_{i,j,c}^2\right)^2 \tag{3}$$

where *i*,*j* are pixel coordinates; n is the total number of pixels; C is the RGB channel index of the color image.

Image semantic segmentation task assigns a label based on what the pixel represents. Through a series of convolution and pooling operations, we can obtain a low-resolution multichannel feature map containing the characteristics of the relationship between the object and its environment. The feature map can provide the contextual semantic information of the segmented target in the entire image.

For a dataset with n classes and labels, predictions are made of the probabilities that the pixels belong to each class, and the sum of these probabilities is 1:

$$\sum_{i=1}^{n} P((x,y)|i) = 1 \tag{4}$$

P represents the probability that a pixel with coordinates (x,y) belongs to class i. After sampling from a deep feature map, P can be predicted, and the most significant P can be selected as the label of the pixel. Gather the pixels with the same label to generate a Mask, which divides the same class of the region.

For our image semantic segmentation task, the dataset has only two classes: conveyor background and part foreground to reduce the calculation. We use the loss function L_2 to calculate the number of wrong pixels in foreground and background prediction. The smaller of L_2 the better outputs, so minimize the following formula:

$$L_2 = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i| \tag{5}$$

where y_i is the class of predicted pixels (0 as background, 1 as foreground); $\hat{y}_i$ is ground truth; n is the total number of pixels in the image.

For the whole model, to achieve joint learning, we combine the loss function of segmentation decoder and decomposition decoder with training the parameters, as shown below jointly:

$$L = \gamma_1 L_1 + \gamma_2 L_2 \tag{6}$$

γ is a manually set coefficient, which represents the weight of the corresponding loss function. By optimizing L to get the best output of image segmentation, and impact of γ will be shown in the evaluation part.

3.2. Asymmetric Convolution

To achieve better segmentation performance, most ideas for model improvement are mainly focused on: (1) how to connect the layers [27]; (2) combining different layers to improve the learning quality [28]. The asymmetric convolution [29] uses an improved scheme independent of the network structure. It does not increase computation and can fulfill the real-time requirements of image detection in the industrial field. The structure of asymmetric convolution is shown in Figure 2.

Suppose several 2D kernels with compatible sizes operate simultaneously on the same input to produce the output with the exact resolution, and their outputs can be summed. In that case, these kernels can be added at the corresponding positions to obtain an equivalent kernel producing the same output:

$$I \times K^1 + I \times K^2 + I \times K^3 = I \times \left(K^1 \oplus K^2 \oplus K^3\right) \tag{7}$$

Let the input $I \in R^{X \times Y \times C}$, the convolution kernel $K \in R^{A \times B \times C}$ and $\oplus$ is the addition of kernel parameters at corresponding positions. Taking the convolution kernel of 3×1 as an example, there are M convolution kernels and the output of the j convolution kernel K^j is at the Jth channel, the value of a point P in the output can be expressed as follows:

$$P_{:,:,j} = \sum_{c=1}^{C} \sum_{a=1}^{3} \sum_{b=1}^{1} K_{a,b,c}^{j} W_{a,b,c} \tag{8}$$

where W is the corresponding sliding window. If the points P output by the three convolution kernels corresponds to the same sliding window, then the additivity of formula 7 holds (dark color in Figure 2).

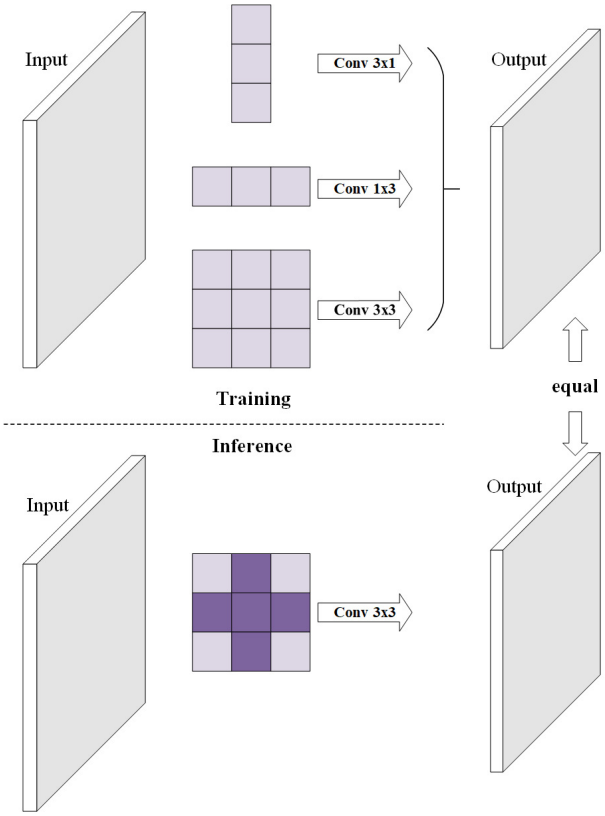

Figure 2. Asymmetric convolution.

3.3. BN Fusion

Batch normalization [30] can accelerate the convergence speed of model training, make the model training process more stable and avoid gradient explosion or gradient disappearance. Usually, neural networks are batch normalized after convolution, which requires two calculations. BN fusion combines these two steps into one.

For a convolution layer, the output is determined by the weights ω and the bias b: $X_l = \omega \times X_{l-1} + b$. The batch normalization is shown in formula (9):

$$X_{norm} = \gamma \times \frac{X - \mu}{\sqrt{\sigma^2 + \varepsilon}} + \beta \tag{9}$$

γ and β are trainable hyperparameters, which are iterated by backpropagation. As a restore parameter, it retains the distribution of the original data to a certain extent. μ is the mean value of input X, σ^2 is the variance and ε is a constant to avoid errors of dividing by zero.

The homogeneity of convolution allows subsequent BN operations and linear scaling to be integrated into the convolution layer with additional bias. Expand X in formula (8) and take the following deformation:

$$X_{norm} = \frac{\gamma \times \omega}{\sqrt{\sigma^2 + \varepsilon}} \times X_{l-1} + \beta + \gamma \times \frac{b - \mu}{\sqrt{\sigma^2 + \varepsilon}} \tag{10}$$

Construct a new convolution kernel that new weights $\tilde{\omega} = \frac{\gamma \times \omega}{\sqrt{\sigma^2+\varepsilon}}$, new bias $\tilde{b} = \beta + \gamma \times \frac{b-\mu}{\sqrt{\sigma^2+\varepsilon}}$ .Then, the new kernel's output is the same as the result of the original Conv+BN, but only one calculation. After the model is trained, using BN fusion to speed the inference time.

3.4. Model Structure

The model used in this paper has a 5-layer encoder and a 5-layer decoder, respectively. The encoder uses a convolutional kernel of 3×3 and a stride of 2 to extract each layer's features and then uses the batch normalization to reduce the correlation between the layers. After BN operation, the rectified linear unit (ReLU) [31] is used as the activation function. The decoder uses the feature size symmetrical to the encoder for up sampling. The model is shown in Figure 3.

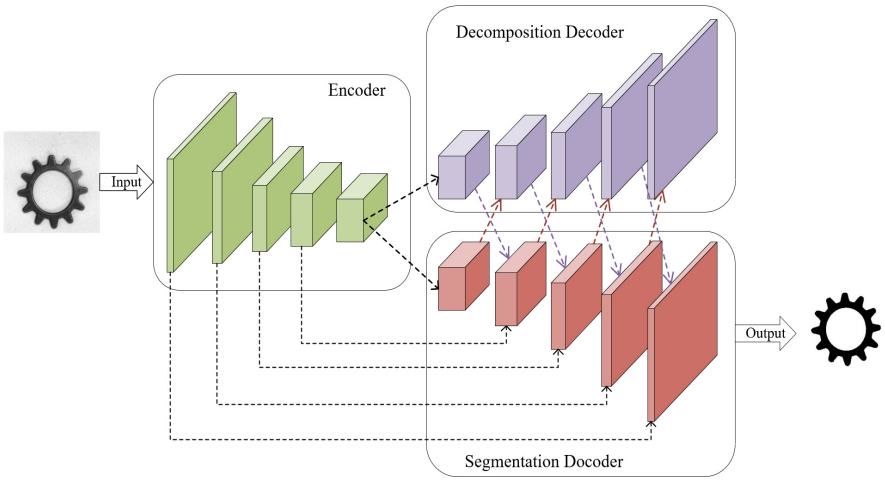

Figure 3. Model structure for joint learning.

There is a mirror link between the encoder and the decoder. We use the copy and crop method in U-Net [16] to make the upsampling process more accurate. There is parameter sharing between the two decoders, the feature values after ReLU activation are shared with each other. There are two reasons for this: (1) The reflectance image obtained by the intrinsic decomposition does not contain illumination components, so as to guide the segmentation process to reduce the incorrect segmentation caused by the difference of highlight and shadow. (2) The class information obtained by semantic segmentation contains prior knowledge about an object's reflectance, which can guide the generation of more accurate reflectance images.

In the training phase, we use three convolution kernels of 3×1, 1×3 and 3×3, all of which use a 3×3 sliding window to match the existing square convolution kernel. The branches of the three convolution kernels are all Conv + BN operation. After the training is completed, they are fused into a standard square convolution kernel of size 3×3. This process does not require any additional hyperparameters. The output composition is as follows:

$$O = \begin{cases} (I \times K_1 - \mu_1)\frac{\gamma_1}{\sigma_1} + \beta_1 \\ (I \times K_2 - \mu_2)\frac{\gamma_2}{\sigma_2} + \beta_2 \\ (I \times K_3 - \mu_3)\frac{\gamma_3}{\sigma_3} + \beta_3 \end{cases} \tag{11}$$

I is the input image, and the output O is the sum of three branches after convolution and batch normalization. μ, γ, σ and β are the parameters in BN operation. The enhanced square convolution kernel contains BN fusion:

$$O = I \times \left(\frac{\tilde{\gamma}_1}{\tilde{\sigma}_1} K_1 \oplus \frac{\tilde{\gamma}_2}{\tilde{\sigma}_2} K_2 \oplus \frac{\tilde{\gamma}_3}{\tilde{\sigma}_3} K_3 \right) + \beta \tag{12}$$

$\tilde{\gamma}$, $\tilde{\sigma}$ and $\tilde{\beta}$ are the parameters after BN fusion. Branches can be converted into standard convolution kernel by adding kernel parameters at the corresponding position.

4. Results and Discussion

This section first introduces the dataset and evaluation index. Then proves the effectiveness of the proposed method through the contrast experiment. Finally, we compare the experimental results with the state-of-the-art model.

4.1. Dataset

In this study, we needed to do effective semantic segmentation for the images collected on the conveyor belt in industrial production, so we constructed a new dataset containing lots of images of metal parts. Considering the different illumination conditions in different conveyor belt areas, different sizes of gears occupy different pixels. A total of 600 original images were collected in a manual setting in an area without specular reflection. After collecting the original images, we added 5% Gaussian noise through Photoshop to obtain noisy images. There are 1200 images in the dataset, and we extracted the target area and cropped the images to 320 × 320 resolution. We randomly selected 1000 images as the training set, and the remaining 200 images were used as the test set.

Intrinsic image decomposition training requires labels that cannot be labeled by hand. Therefore, we used the part image with ideal shooting results as the benchmark, and rendered these images with the specified intensity of high light. The original low-light images were taken as the ground truth reflectance attributes. As for the semantic segmentation label, we manually defined the black area as the foreground and the white area as the background. An example of the dataset is shown in Figure 4.

Figure 4. Examples from the dataset. The first image was taken in the ideal environment, which achieved the best imaging result; the second image has the foreground and background manually marked, which can be used as the label of image semantic segmentation; last is the image with interference factors collected in the simulated industrial environment as the input of our model.

4.2. Evaluation Criteria

The proposed model performs two tasks simultaneously, but the main target is image segmentation. Therefore, for the intrinsic image decomposition decoder, we quote parameters from the existing model ShapeNet [25], and then fine-tune with the industrial parts dataset, and use MSE to measure the decomposition effect.

For the image semantic segmentation task, we use the pixel accuracy to evaluate the prediction effect:

$$PA = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

True positive (TP) is the number of pixels correctly predicted as the target; true negative (TN) is the number of pixels accurately indicated as the background. On the contrary, false positive (FP) and false negative (FN) are the pixels with the wrong predictions.

To comprehensively analyze the segmentation performance and compare with the mainstream segmentation models, we also used IoU to evaluate the segmentation effect:

$$IoU = \frac{target \cap prediction}{target \cup prediction} \tag{14}$$

4.3. Experimental Results and Analysis

The experimental configuration was as follows: CPU, AMD R5 2600; GPU, NVIDIA GTX 1660Ti; RAM, 16 GB. Experiments were programmed in the TensorFlow framework [32], the code running environment was Python3.8 and the deep learning environment was CUDA10.1 and cudnn7.6. We used RMSProp [33] optimizer to train the model.

4.3.1. Experiment One: Effectiveness of Parameter Sharing in Joint Learning

This experiment mainly verifies the effectiveness of parameter sharing. We use the control variable method to test the following cases. Case 1: without joint learning; only the semantic segmentation decoder works. Case 2: the intrinsic image decomposition decoder shares one-way parameters with the semantic segmentation decoder to help with the segmentation task. Case 3: two decoders pass parameters to each other to achieve joint learning. The experimental results are shown in Table 1.

Table 1. Parameter sharing in joint learning.

/	Segmentation		Decomposition
	PA	**IoU**	**Reflectance MSE**
case 1	0.903	0.766	-
case 2	0.927	0.787	0.0113
case 3	0.941	0.813	0.0089

The results show that the joint learning method dramatically improves the performance of semantic segmentation task. Although a single semantic segmentation task can learn some illumination changes, it cannot eliminate the adverse factors caused by lighting differences. The intrinsic image decomposition can extract some commonalities. For example, shading is usually smooth and gray, and the specular reflection is sparse and has high contrast. Therefore, these commonalities can be used to guide the semantic segmentation task. At the same time, the object category provided by semantic segmentation also contains some commonalities of reflectivity, which can in turn, guide the intrinsic image decomposition task. Therefore, in case 3 of joint learning, the segmentation result is better, but the decomposition is improved. This is because the two tasks promote each other and enhance the performance.

4.3.2. Experiment Two: Weights of Loss Function

This experiment verified the influence of loss function weights. Let the sum of γ_1 and γ_2 be 1; we analyzed the weights of two tasks of different proportions, and how that affect the model's segmentation performance. The experimental results are shown in Table 2.

Table 2. Influences of loss function weights.

γ_1	γ_2	**PA**	**IoU**
0.1	0.9	0.927	0.792
0.3	0.7	0.941	0.813
0.5	0.5	0.892	0.766
0.6	0.4	0.825	0.694
0.7	0.3	0.735	0.586
0.9	0.1	0.516	0.367

In Table 2, as the decomposition loss function γ_1 increased, the degree to which the intrinsic attributes guide the semantic segmentation gradually increases, and the segmentation performance is improved. However, with the weight γ_1 becoming bigger and bigger, the target of the model focuses on image decomposition; then the evaluation index of segmentation decays rapidly. The goal of this study was to achieve the best semantic segmentation performance, so according to the experimental results, $\gamma_1 = 0.3$ and $\gamma_2 = 0.7$ were selected as relatively optimal combinations.

4.3.3. Experiment Three: Influence of Asymmetric Convolution

This experiment compared the effects of the standard convolution kernel and asymmetric convolution blocks on the model. This experiment was carried out on the parameter configuration of the first two experiments, and three groups of test results were selected, as shown in Table 3.

Table 3. Performances of different convolution kernels.

/	Test 1		Test 2		Test 3	
	PA	**IoU**	**PA**	**IoU**	**PA**	**IoU**
convention	0.941	0.813	0.938	0.809	0.940	0.811
new kernel	0.945	0.818	0.943	0.813	0.945	0.816

For the comparability, all the models were trained until complete convergence, and all used the same configurations, such as learning rate and batch size. In the data comparisons, conventional kernel enhanced by the asymmetric convolution blocks can improve the performance of segmentation, increasing the pixel accuracy of about 0.5%; this phenomenon shows that the different weights inside the enhanced convolution kernel are more important to the model's representation ability. Enhanced convolution kernel does not require additional hyperparameters and inference calculations. This method can be used to improve the accuracy when the model is constrained by computational budgets or model size.

4.3.4. Experiment Four: Influence of BN Fusion

Based on the previous investigation, the convolution layer and BN layer were fused in this experiment. To verify BN fusion's ability to accelerate the processing, we divided the test set images into 10 batches, and each batch had 100 images. We calculated the time required to process each batch under different conditions, as shown in Figure 5.

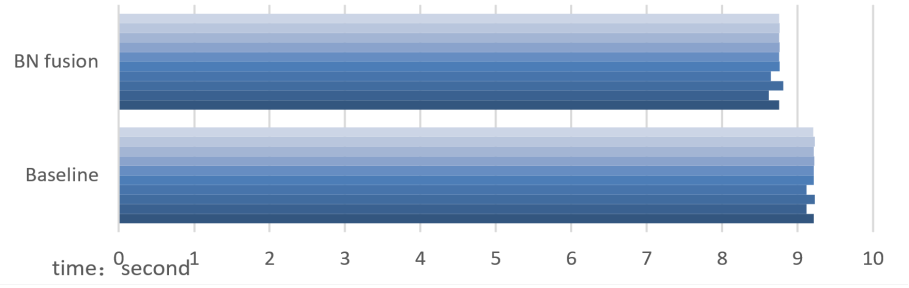

Figure 5. Processing time of each batch.

According to the figure, each batch of images enhanced by BN fusion was generally faster than baseline, and the average time was increased by about 4.5%. It is worth noting that the model proposed in this paper is a Conv + BN + ReLu structure, so the fusion of the convolutional layer and the BN layer is a linear transformation, which will not bring about calculation errors. BN fusion cannot be used for any nonlinear operation in the middle layer, such as a Conv + ReLu + BN structure.

4.3.5. Experiment Five: Comparison with Classical Semantic Segmentation Model

We verified the effectiveness of our model through the segmentation of the industrial part dataset. We chose two state-of-the-art classical models: FCN [14] and U-Net [16]. The experimental results are shown in Table 4.

Table 4. Results of different models.

Model	PA	IoU
SegNet	0.908	0.759
U-Net	0.911	0.769
Ours	0.945	0.818

In Table 4, the results show that, compared with the traditional single-task segmentation model, the proposed joint learning approach can significantly improve the not's segmentation performance after eliminating adverse interference. The comparison results are shown in Figure 6.

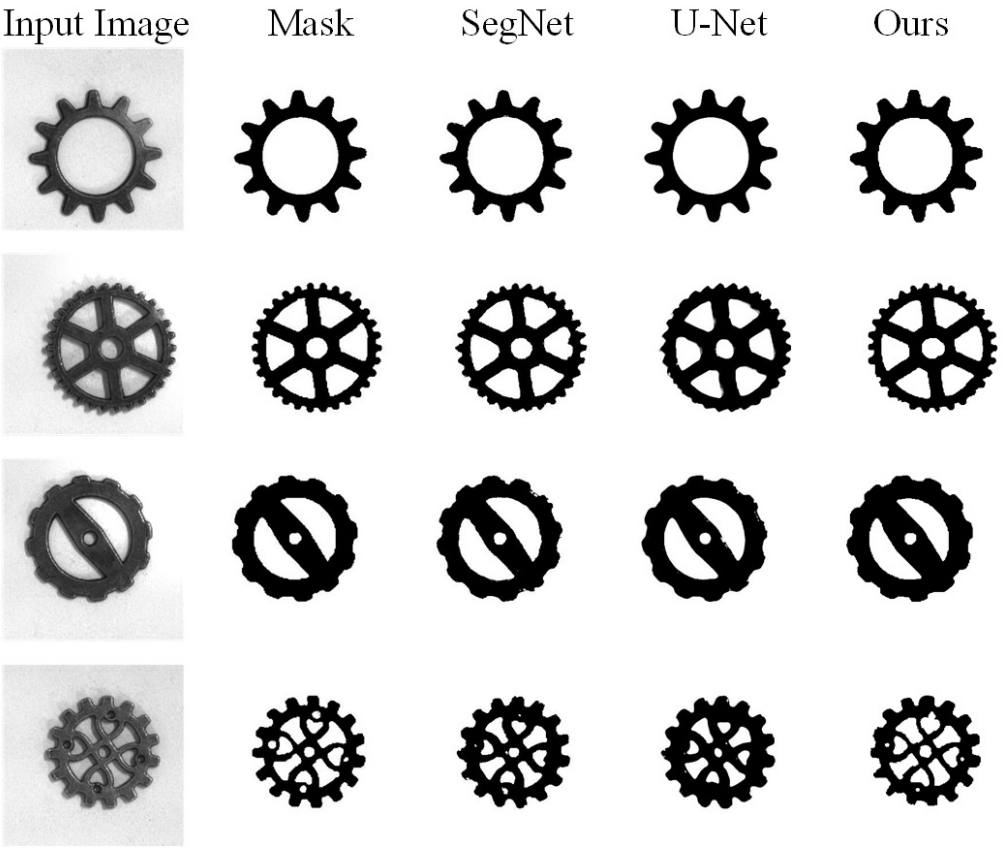

Figure 6. Comparison of different models.

In the visualized semantic segmentation results, this paper provides four groups of different types of gear images to compare the performances of different models. The first group's image has a simple object, and the segmentation results of our method and comparative method are both excellent. However, for the second group, the metal part image has specular reflection and shadow at the same time. Our approach can eliminate these two disadvantageous interferences. In the contrast experiment, SegNet eliminated shadow interference, but the external contour has many gaps; U-Net has continuous contour, but the shadow is also judged as the target. To demonstrate the ability to resist the highlight influence, the third group of images has obvious light details on the upper right of the target. Our method correctly identified it as the target. Still, the two comparison algorithms have

some misjudgments to some extent. That is, there are some separate background points in the gear area. The last group of images is complicated; the proposed model combines the reflection features for semantic segmentation, and achieved a relatively complete contour. However, because the color inside the target was close to the shadow, the classification of details was not ideal. U-Net tends to distinguish the shadow as the part target, so the segmentation result is bulky. SegNet, on the contrary, got many missing edges.

5. Conclusions

The images of mechanical parts collected by industrial IoT cameras are affected by the light source. Traditional methods cannot eliminate the effects of illumination. To address this problem, this paper investigated the use of deep learning-based semantic segmentation methods for mechanical image detection in industrial IoT. Then, a joint learning approach was proposed to improve the detection performance. The proposed method uses reflection feature maps to guide semantic segmentation without the influence of illumination. Although the proposed model is trained on the rendering dataset, it can simulate the effects of specular reflection and shadow well, so it can produce better results on real images compared to other algorithms, with a pixel accuracy improvement of about 4%. The simulation study showed that our proposed approach can effectively eliminate illumination and achieve satisfying image detection performance in industrial IoT. In the future, the follow-up work will be carried out in multi-object instance segmentation to make the algorithm more suitable for the images collected on the industrial scene.

Author Contributions: All authors took part in the discussion of the work described in this paper. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China, under grant numbers 61762037 and 61872141, and the Science and Technology Key Research and Development Program of Jiangxi Province, under grant numbers 20202BBEL53004 and 20203BBE53029.

Data Availability Statement: Data available on request due to restrictions by funding organization.

Acknowledgments: The authors acknowledge the anonymous reviewers and editors for their valuable comments and suggestions.

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

Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional neural networks
MSE	Mean square error
BN	Batch normalizationm
ReLU	Rectified linear unit
IoU	Intersection over union

References

1. Songlin, S.; Michel, K.; Liang, G.; Bo, R. Integrating Network Function Virtualization with SDR and SDN for 4G/5G Networks. *IEEE Netw.* **2015**, *29*, 54–59.
2. Fumi, T.; Masaharu, K.; Aizoh, K. High-precision measurement of an involute artefact: By a rolling method and comparison between measuring instruments. *Meas. Sci. Technol.* **2009**, *20*, 1–12.
3. Xie, X.; Ge, S.; Xie, M. An improved industrial sub-pixel edge detection algorithm based on coarse and precise location. *J. Ambient Intell. Humaniz. Comput.* **2019**, *10*, 1–10. [CrossRef]
4. Ofir, N.; Galun, M.; Alpert, S.; Brandt, A.; Nadler, B.; Basri, R. On Detection of Faint Edges in Noisy Images. *Pattern Anal. Mach. Intell.* **2019**, *42*, 894–908. [CrossRef] [PubMed]
5. Duan, R.; Li, X.; Li, Y. Summary of image edge detection. *Opt. Tech.* **2005**, *31*, 415–419.
6. Ghosh, S.; Das, N.; Das, I.; Maulik, U. Understanding Deep Learning Techniques for Image Segmentation. *ACM Comput. Surv.* **2019**, *52*, 1–35. [CrossRef]
7. Nishant, S.; Parveen, S.; Rahul, S.; Shriniwas, P. Image Segmentation in Constrained IoT Servers *Procedia Comput. Sci.* **2019**, *165*, 336–342.

8. Khan, H.; Shah, P.M.; Shah, M.A.; ul Islam, S.; Rodrigues, J.J. Cascading handcrafted features and Convolutional Neural Network for IoT-enabled brain tumor segmentation. *Comput. Commun.* **2020**, *153*, 196–207. [CrossRef]
9. Zheng, S.; Hao, Y.; Lu, D.; Bao, H.; Xu, J.; Hao, H.; Xu, B. Joint Entity and Relation Extraction Based on A Hybrid Neural Network. *Neurocomputing* **2017**, *257*, 59–66. [CrossRef]
10. Zheng, S.; Wang, F.; Bao, H.; Hao, Y.; Zhou, P.; Xu, B. Joint Extraction of Entities and Relations Based on a Novel Tagging Scheme. *Assoc. Comput. Linguist.* **2017**, *1*, 1227–1236.
11. Subramanian, S.; Trischler, A.; Bengio, Y.; Pal, C.J. Learning General Purpose Distributed Sentence Representations via Large Scale Multi-task Learning. *arXiv* **2018**, arXiv:1804.00079.
12. Ruder, S.; Bingel, J.; Augenstein, I.; Søgaard, A. Latent Multi-task Architecture Learning. *AAAI* **2019**, *30*, 4822–4829. [CrossRef]
13. Hashimoto, K.; Xiong, C.; Tsuruoka, Y.; Socher, R. A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing 2017, Copenhagen, Denmark, 7–11 September 2017; pp. 1923–1933.
14. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
15. Krizhevsky, A.; Sutskever, I.; Hinton, G. ImageNet Classification with Deep Convolutional Neural Networks. *Adv. Neural Inf. Process. Syst.* **2012**, *25*, 1097–1105. [CrossRef]
16. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Cham, Switzerland, 2015; pp. 234–241.
17. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask R-CNN. In Proceedings of the IEEE International Conference on Computer Vision, Piscataway, NJ, USA, 22–29 October 2017; pp. 2961–2969.
18. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. DeepLab: Semantic Image Segmentation with Deep Convolutional Nets, Atrous Convolution, and Fully Connected CRFs. *IEEE Trans. Pattern Anal. Mach. Intell.* **2018**, *40*, 834–848. [CrossRef] [PubMed]
19. Deng, J.; Dong, W.; Socher, R.; Li, L.J.; Li, K.; Li, F.-F. ImageNet: A large-scale hierarchical image database. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 248–255.
20. Stan, T.; Thompson, Z.T.; Voorhees, P.W. Optimizing convolutional neural networks to perform semantic segmentation on large materials imaging datasets: X-ray tomography and serial sectioning. *Mater. Charact.* **2020**, *160*, 110–119. [CrossRef]
21. Smith, A.G.; Petersen, J.; Selvan, R.; Rasmussen, C.R. Segmentation of roots in soil with U-Net. *Plant Methods* **2020**, *16*, 13–27. [CrossRef] [PubMed]
22. Vuola, A.O.; Akram, S.U.; Kannala, J. Mask-RCNN and U-net Ensembled for Nuclei Segmentation. In Proceedings of the International Symposium on Biomedical Imaging, Venice, Italy, 8–11 April 2019; pp. 208–212.
23. Baslamisli, A.S.; Groenestege, T.T.; Das, P.; Le, H.A.; Karaoglu, S.; Gevers, T. Joint Learning of Intrinsic Images and Semantic Segmentation. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 289–305.
24. Li, S.; Ping, T.; Stephen, L. Intrinsic image decomposition with non-local texture cues. In Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, AK, USA, 23–28 June 2008.
25. Shi, J.; Dong, Y.; Su, H.; Yu, S.X. Learning non-lambertian object intrinsics across shapenet categories. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 5844–5853.
26. Land, E.H.E. Lightness and Retinex theory. *J. Opt. Soc. Am.* **1971**, *61*, 1–11. [CrossRef] [PubMed]
27. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
28. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
29. Ding, X.; Guo, Y.; Ding, G.; Han, J. ACNet: Strengthening the Kernel Skeletons for Powerful CNN via Asymmetric Convolution Blocks. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Seoul, Korea, 27–28 October 2019; pp. 1911–1920.
30. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In Proceedings of the International Conference on Machine Learning, Lille, France, 6–11 July 2015; pp. 448–456.
31. Xavier, G.; Antoine, B.; Bengio, Y. Deep Sparse Rectifier Neural Networks. *J. Mach. Learn. Res.* **2011**, *15*, 315–323.
32. Google. TensorFlow, an Open-Source Machine Learning Framework for Everyone. 2016. Available online: https://www.tensorflow.org (accessed on 11 August 2020).
33. Zou, F.; Shen, L.; Jie, Z.; Zhang, W.; Liu, W. A Sufficient Condition for Convergences of Adam and RMSProp. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 11127–11135.