applied sciences

MDPI

Article

Color Structured Light Stripe Edge Detection Method Based on Generative Adversarial Networks

Dieuthuy Pham [1,2], Minhtuan Ha [1,2] and Changyan Xiao [1,*]

[1] College of Electrical and Information Engineering, Hunan University, Changsha 410208, China
[2] Department of Electrical, Saodo University, Hai Duong 170000, Vietnam
* Correspondence: c.xiao@hnu.edu.cn; Tel.: +86-153-6781-3281

Abstract: The one-shot structured light method using a color stripe pattern can provide a dense point cloud in a short time. However, the influence of noise and the complex characteristics of scenes still make the task of detecting the color stripe edges in deformed pattern images difficult. To overcome these challenges, a color structured light stripe edge detection method based on generative adversarial networks, which is named horizontal elastomeric attention residual Unet-based GAN (HEAR-GAN), is proposed in this paper. Additionally, a De Bruijn sequence-based color stripe pattern and a multi-slit binary pattern are designed. In our dataset, selecting the multi-slit pattern images as ground-truth images not only reduces the labor of manual annotation but also enhances the quality of the training set. With the proposed network, our method converts the task of detecting edges in color stripe pattern images into detecting centerlines in curved line images. The experimental results show that the proposed method can overcome the above challenges, and thus, most of the edges in the color stripe pattern images are detected. In addition, the comparison results demonstrate that our method can achieve a higher performance of color stripe segmentation with higher pixel location accuracy than other edge detection methods.

Keywords: GAN-based method; edge detection; structured light; pixel location accuracy

check for updates

Citation: Pham, D.; Ha, M.; Xiao, C. Color Structured Light Stripe Edge Detection Method Based on Generative Adversarial Networks. *Appl. Sci.* **2023**, *13*, 198. https://doi.org/10.3390/app13010198

Academic Editor: Byung-Gyu Kim

Received: 11 November 2022
Revised: 12 December 2022
Accepted: 20 December 2022
Published: 23 December 2022

Copyright: © 2022 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

Due to the high demand for scene-depth information in real-time applications, one-shot three dimensional (3D) imaging methods are attracting more and more researchers. Among them, the methods using a color stripe pattern have many outstanding advantages, such as robustness against occlusion and discontinuity problems, the ability to capture both dynamic and static scenes, and a short measurement time.

Most of the color stripe patterns in structured light systems are based on the De Bruijn sequence [1–4]. The De Bruijn sequence of n colors with a property window of size m is a sequence containing $n \times m$ subsequences of length m that occur exactly once in the entire pattern. The differences in the different coding techniques are the number of colors used and the size of the property window. Typically, the size of the property window in a color stripe or slit projection pattern is 1×5 or 1×7. For capturing multi-colored objects, in [5], Lee et al. used a filtering method that improved the color classification of the stripes and proposed a technique to combine the two- or three-axis coded stripe pattern to enhance the accurate color recognition of each stripe in the image of a deformed pattern. This way, the method was less sensitive to the characteristics of the scene. Instead of using a single pattern encoded with several colors, patterns encoded with only two colors in the RGB color space were introduced in [6,7]. Although the use of fewer colors to encode the pattern makes feature point detection easier, more complex pattern decoding methods are required.

In addition to the above advantages, the methods using a color stripe pattern are still challenged by detecting feature points, which are the points on the border between two adjacent color stripes in the deformed pattern images. To improve the density of

three-dimensional maps, high-resolution color stripe patterns, which are made up of a large number of color stripes with a small width, should be used. In addition, more colors are required to encode the patterns. These issues make the patterns more sensitive to noise and color crosstalk. Moreover, these methods also suffer from the object surface characteristics, such as color, inter-reflectivity, textures, etc., when capturing objects with a complex surface structure under room lighting conditions. These problems cannot be solved by traditional edge detection methods.

With an important role in practical image processing-based applications, so far, many studies on edge detection algorithms have been published. Most of the methods are applied to grayscale images because of their high accuracy [8–12]. In order to overcome the challenges of edge detection in color images, many methods of converting color images into grayscale images to take advantage of grayscale images in edge detection have been proposed. In [13], 10 methods for converting color images into grayscale images were performed to compare their influence on eight different edge detection algorithms. It was found that the resulting images depended not only on the edge detection method but also on the image conversion method.

On the other hand, the methods of directly detecting the edges in color images have also improved. In [14], Cheon et al. proposed a multi-dimensional edge detection algorithm for detecting an edge in color images. First, the noise in RGB images is suppressed by applying the Wiener filter. Then the Bhattacharyya distances of four block pairs are calculated. Finally, the maximum values are used to determine the edges. The performance of this method in detecting weak edges is better than the traditional Sobel and Canny operators. Taking advantage of the Canny operator, an improved algorithm based on the concept of vectors was proposed in [15]. Their experimental results proved that this method outperformed the edge detection methods of gray images in the field of color edge detection, especially in outline detection and fake edge suppression. Similarly, an edge detection method in color images was also proposed in [16]. The outstanding improvement of this method is the use of an adaptive median filter to eliminate noise and to apply type 2 fuzzy set-based thresholding to solve the problem of vagueness and uncertainties in gradient images. Therefore, this method achieves better edge detection performance than other traditional algorithms.

In recent years, deep learning techniques have been widely applied in the field of image processing. Among these, GAN-based approaches and their potential are increasingly being developed [17]. Furthermore, the application of GAN as an image-to-image translation network helps to solve many problems in computer vision, especially object semantic detection and segmentation [18–24]. Additionally, many variations of GAN have been introduced, such as CrackGAN [21], DCGAN [22], cycle-GAN [23], URCA-GAN [25], etc. To perform the above tasks, a series of other deep learning networks have also been proposed [26–33]. Although deep learning models have been used for edge detection in many studies [27–30], no study on color structured light stripe edge detection has been presented.

As mentioned above, the accurate edge detection for color stripe pattern images depends on the quality of the captured images, which is strongly influenced by noise. After studying many previous approaches, we found that GAN-based models can effectively solve the problem of noise reduction in images [34–36]. As a translator, after being trained with real images and those generated by the generator, the discriminator in GANs can generate images that look like the real ones. Therefore, for detecting edges in the color stripe pattern images with high pixel location accuracy, a color structure light stripe edge detection method based on generative adversarial networks (GANs) is proposed in this paper. In addition to the De Bruijn sequence-based color stripe pattern, a binary multi-slit pattern was also designed and used in the experiments. It should be noted that the slits in the binary multi-slit pattern are located in positions corresponding to the left edges of the color stripes in the stripe pattern. Firstly, the images of objects with the two patterns at each position are respectively captured by the camera. Among them, the images with multi-slit

patterns are thinned and then selected as ground-truth images. After that, the images of a deformed color stripe pattern with the corresponding ground-truth images are sequentially cut into small patches and then divided into the training, validation, and testing sets. After training the model, the test images are predicted by the trained model and then merged together into large images. Finally, the centerlines in the resulting images are detected as color structured light stripe edges. With our experimental results, the proposed method is shown to overcome the above-mentioned challenges and thus detect edges in color stripe images with higher pixel location accuracy than other methods.

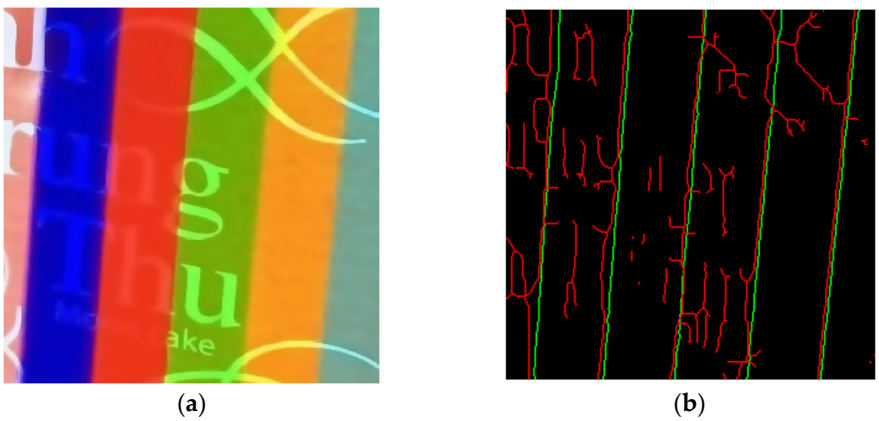
The rest of this paper is organized as follows. After the problem definition is introduced, the materials and methods, which include our network structure, dataset preparation, training process, and the color structured light stripe edge detection method, are presented in Section 2. After conducting the experiments, the evaluations of the segmentation performance of the proposed network and the pixel location accuracy of the detected edges are presented in Section 3. Finally, the conclusions about the proposed method are given in Section 4.

2. Problem Definition and Our Approach Overview

2.1. Problem Definition

For single-shot structured light imaging systems using a color stripe pattern, the accurate detection of feature points, which are points on the boundary of two adjacent color stripes in a deformed pattern image, is extremely important because it directly affects the quality of the reconstructed point cloud. In practice, this faces many challenges from the complex surface structure of the captured object and ambient light noise. This problem becomes even more serious when high-frequency color stripe patterns are used to increase the density of the point cloud. The two popular methods for solving this problem are edge detection methods based on gradients and color classification [10]. The first method is false detection risk caused by the appearance of irregular edges due to the surface structure of the object [37]. Thus, it is difficult to apply to the images of a high-frequency color stripe pattern [38]. Meanwhile, the second method is easily affected by the colored textures and inter-reflectivity of the object's surface [39].

Figure 1 illustrates the influence of object surface characteristics on the accuracy of edge detection. Figure 1a shows an image of a plastic box whose surface has a lot of colored textures, such as printed shapes and text. In Figure 1b, the green lines indicate the edges in the ground-truth image, while the red lines indicate the edges detected by the edge-detection method based on color classification. It is worth noting that there are several false detections in the edge image. Traditional edge detection methods cannot distinguish the inherent colored textures of the object surface from the color structured light patterns projected on the object. To overcome this, a method based on deep learning needs to be developed.



(**a**) (**b**)

Figure 1. Illustration of the influence of object surface characteristics on the accuracy of the edge detection; (**a**) an image of a colored textured plastic box; (**b**) false detections in an edge image.

2.2. Approach Overview

The goal of this paper is to propose an edge detection method that can fully detect the edges in images of color stripe patterns with high pixel location accuracy. Our approach included the following steps:

(1) The characteristics of the deformed color stripe pattern images were analyzed;
(2) The previous methods of edge detection in color images were evaluated;
(3) A GAN-based method for detecting the color structured light stripe edge was proposed;
(4) A specific dataset was designed;
(5) The model was trained;
(6) The color structured light stripe edges in the test images were detected using the trained model;
(7) A comparison with other methods was conducted to evaluate the performance and accuracy of the proposed method.

The workflow of the proposed method is shown in Figure 2. First, the multi-slit binary and color stripe patterns are designed. Notice that the slits in the multi-slit projection are located at the positions of the left edges of the color stripes in the color stripe pattern. After the images of the objects with these two patterns are captured, they are divided into a training set and a test set. In the training set, the images of the multi-slit pattern, after being preprocessed by image processing algorithms (including noise reduction, skeletonization, and dilation morphology) with the corresponding color stripe pattern images, are cropped into image patches. Similarly, the test images are also cropped into small patches and predicted with the trained model. The resulting image patches are then merged together to form large images. Finally, the centerlines are detected in these images, which are located at the position of the left edges of the color stripes in the color stripe pattern images.

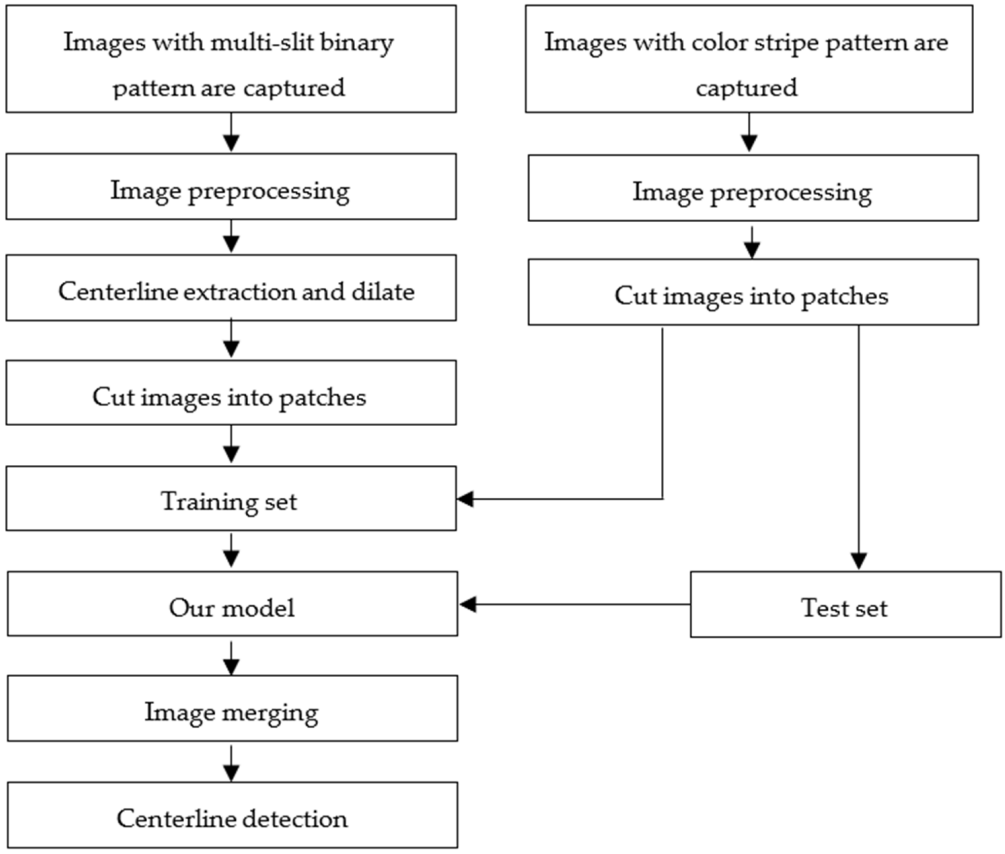


Figure 2. The workflow of the proposed method.

3. Proposed Method

3.1. Network Architecture

3.1.1. Attention Gate

In the field of image segmentation, especially medical images, Unet has achieved many outstanding achievements that are reflected in its variants that have been proposed in many papers [26,31–40]. However, many redundant low-level features are brought across from the down sampling path to the up sampling path when skip connections are used. Therefore, the skip connection needs to be improved with an attention gate to actively suppress activations in irrelevant regions and reduce the number of redundant features obtained. After referring to the content on attention gates in [41], we designed an attention gate in Keras, as shown in Figure 3.

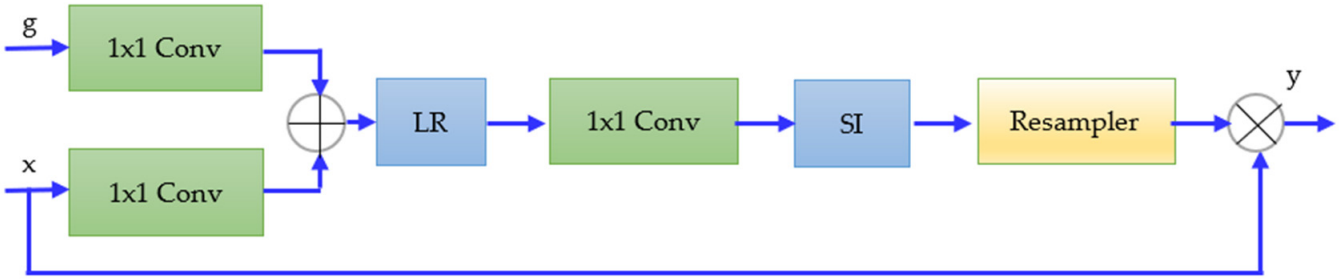


Figure 3. Illustration of the attention gate in the proposed network.

In the attention gate, x is the skip connection, g represents the gating signal, and y is the output of the gate. The x and g signals, after being added together element-wise, will pass through the convolution and resampled layers with the leaky ReLU (LK) and sigmoid (SI) activation functions. After that, the resulting signal is multiplied element-wise to the original x. This is then passed along in the skip connection as normal.

3.1.2. Horizontal Elastomeric Attention Residual Unet-Based GAN (HEAR-GAN)

Accurately detecting the edges in the images of a deformed color stripe pattern is difficult because of the effects of the complex structure of the object surface, as mentioned above. To improve this, the proposed method consists of two stages: (1) segmenting lines with a width of 3 pixels from the original images; (2) detecting the centerlines in the resulting images. With the idea of solving the task of line segmentation in an image based on image-to-image translation, our network consists of two parts: the generator and discriminator (Figure 4).

The generator is a CNN consisting of encoding and decoding paths arranged in a W-shaped model. Among these, most of the plain neural blocks are replaced by residual blocks. The size input and output images of the network are $128 \times 128 \times 3$. In the entire network, the batch normalization (BN) and leaky ReLU (LK) activation functions are applied to all blocks except the input and output layers. Filters with a kernel size of 4×4 are used. The amount of filters is doubled after each block, which starts with $f_d = 32$ filters in encoding paths and is reduced by half in decoding paths. In encoding paths, max-pooling is used to reduce the sample size, while up sampling layers are employed in decoding paths. In addition, dropout is applied to the layers that have a large number of parameters in each path to reduce overfitting. Lastly, a convolutional layer with a kernel size of 1×1 with the sigmoid (SI) activation function is used for the last layer of the network. It is worth noting that the skip connections used in traditional Unet networks are all improved with attention gates.

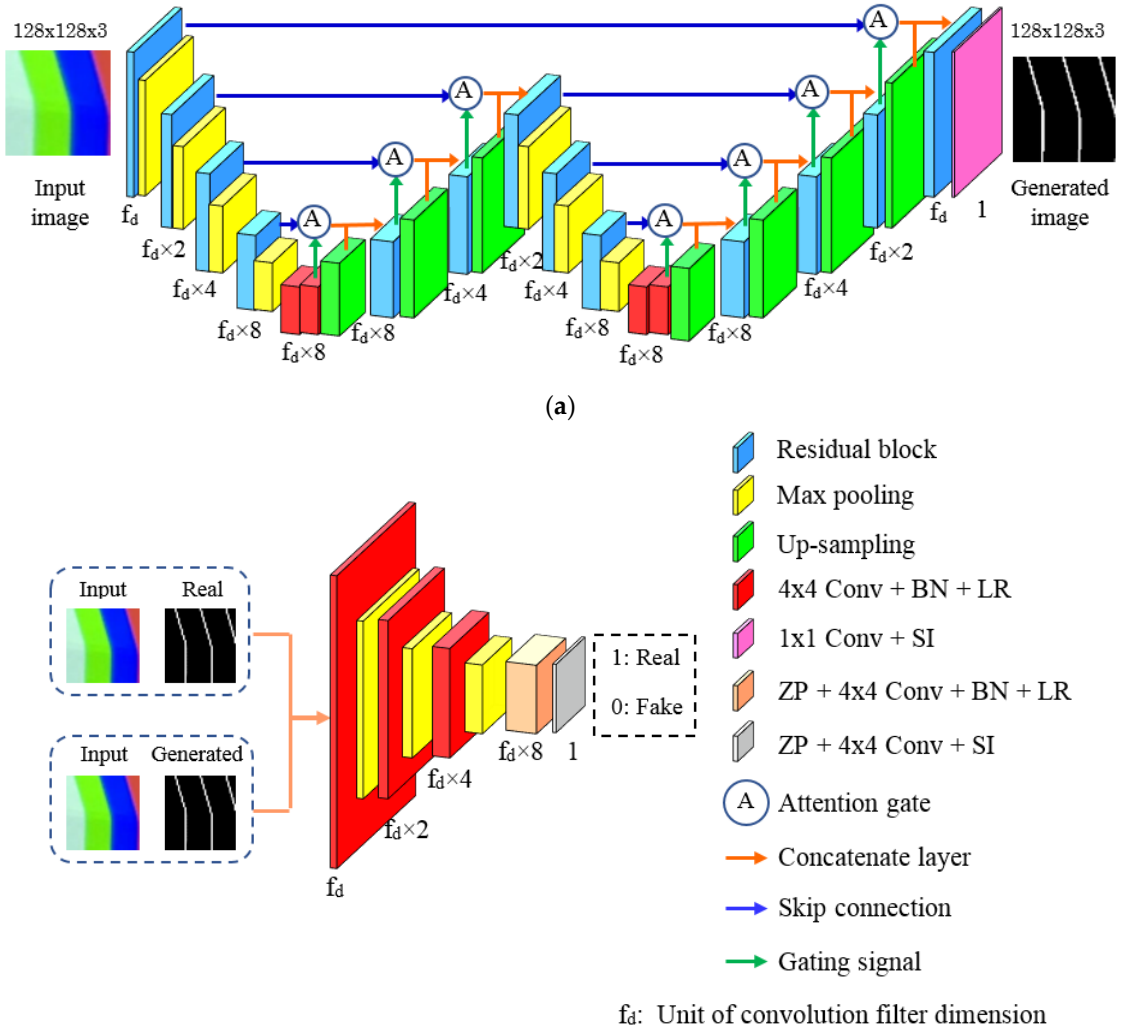


Figure 4. The architecture of our network. (**a**) Generator. (**b**) Discriminator.

In the discriminator, the input data are the image pairs from two sets. The first set includes the input images themselves and their real label. Therefore, the data from this set are called the real pair data. Meanwhile, the second set includes the input images and the label generated from the generator. The data from this set are called fake pair data. Similar to the generator, we use convolutional layers with leaky ReLU and batch normalization in each layer to encode the input data into a feature vector. Lastly, the sigmoid activation function is applied to the feature vector to obtain the binary output {0: fake pairs, 1: real pairs}.

Compared to those in previous studies, the proposed network has the following characteristics: (1) All plain neural blocks, except for convolutional layers at bridges, are replaced by residual blocks; (2) the batch normalization and leaky ReLU activation functions are applied to all blocks; (3) skip connections are improved with attention gates; (4) the generator is a W-shaped network whose number of convolution layers is much larger than in previous GAN-based networks; (5) applying dropout on layers with a large number of parameters and max-pooling on encoding paths can help the model avoid overfitting.

3.2. Dataset Design

The dataset is designed in three main steps: (1) image collection, (2) data labeling, and (3) data augmentation.

First, the color stripe and multi-slit binary patterns are projected onto the objects, respectively. After the images are captured by the camera, they are classified into two groups: stripe images and multi-slit images.

The process of data labeling is shown in Figure 5. The multi-slit images are first preprocessed with a Gaussian filter to reduce noise in the image. After the centerline images are obtained (Figure 5b), a morphological dilation operation is applied to obtain 3-pixel line images (Figure 5c). Finally, the resulting images are sequentially cut into patches with a size of 128 × 128 pixels and selected as the labeled images.

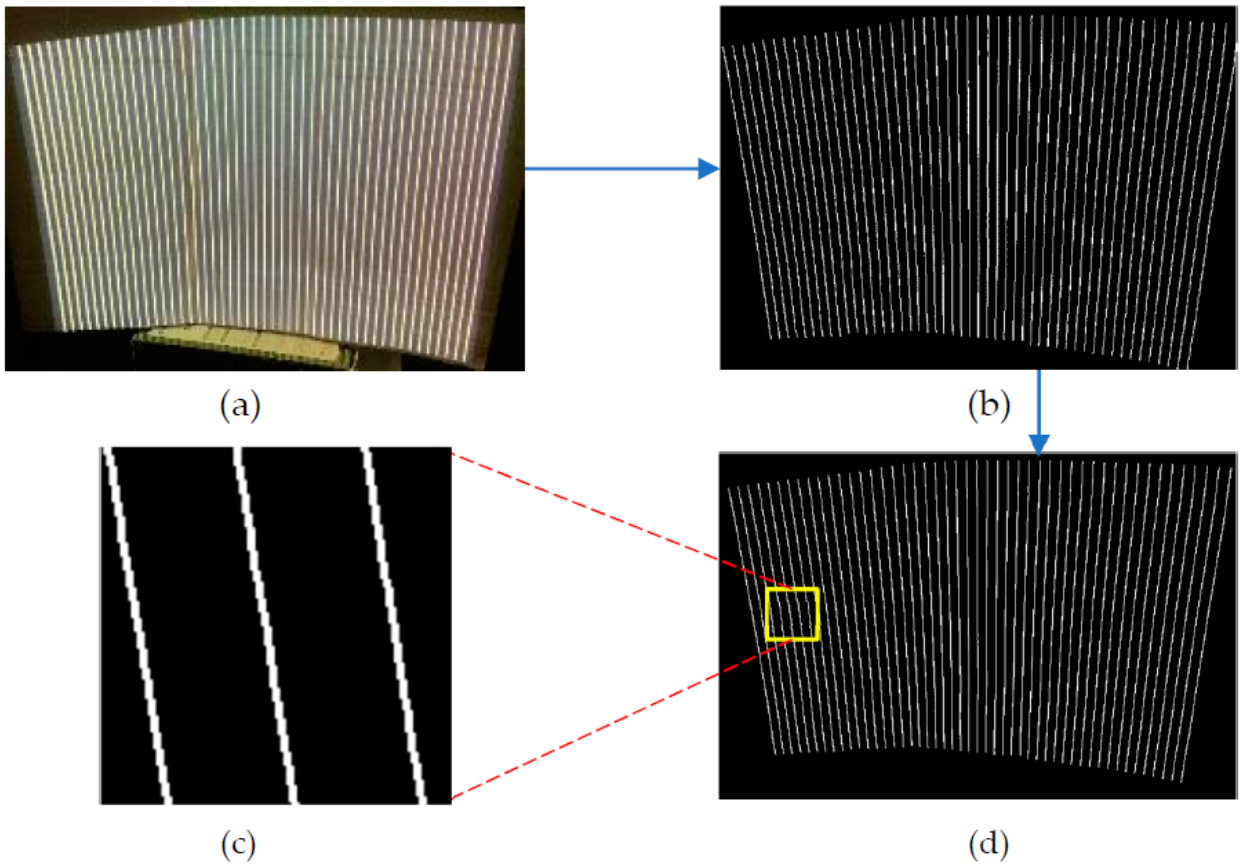


Figure 5. Illustration of the data-labeling process. (**a**) Multi-slit image. (**b**) Centerline image. (**c**) Labeled image. (**d**) Three-pixel line image.

Our original dataset consisted of 40 pairs of original large images captured with the two patterns. Five of them were selected as test sets. After a bilateral filter was applied to reduce noise in the images of a color stripe pattern, these images with the corresponding 3-pixel line images were sequentially cut into patches with a size of 128 × 128 pixels. As seen in Figure 6, each sample included an image patch and its label, and 20% of them were selected as the validation set. After applying some data augmentation techniques to all remaining samples, the training set was obtained with more than 1 million samples.

In the color stripe edge detection method proposed in this paper, the design of the dataset plays a very important role. A color stripe pattern and a multi-slit binary pattern need to be designed accurately first, where the slits in the multi-slit binary pattern must be located at the positions corresponding to the left edges of the color stripes in the color stripe pattern. The multi-slit binary pattern images are processed to obtain the ground-truth images corresponding to the color stripe pattern images in the training and test sets. In many previous studies, data labeling has been done manually by experts. However, this requires a lot of time and effort, especially when the number of images is large. Moreover, the manual labeling is not objective, so the pixel location accuracy of the ground-truth

images is not high. With our dataset, the ground-truth images are obtained directly from processing the multi-slit binary pattern images, so the pixel location accuracy is guaranteed. Thus, the proposed method can not only achieve ground-truth images with higher quality, but it also saves more time and effort than traditional manual labeling.

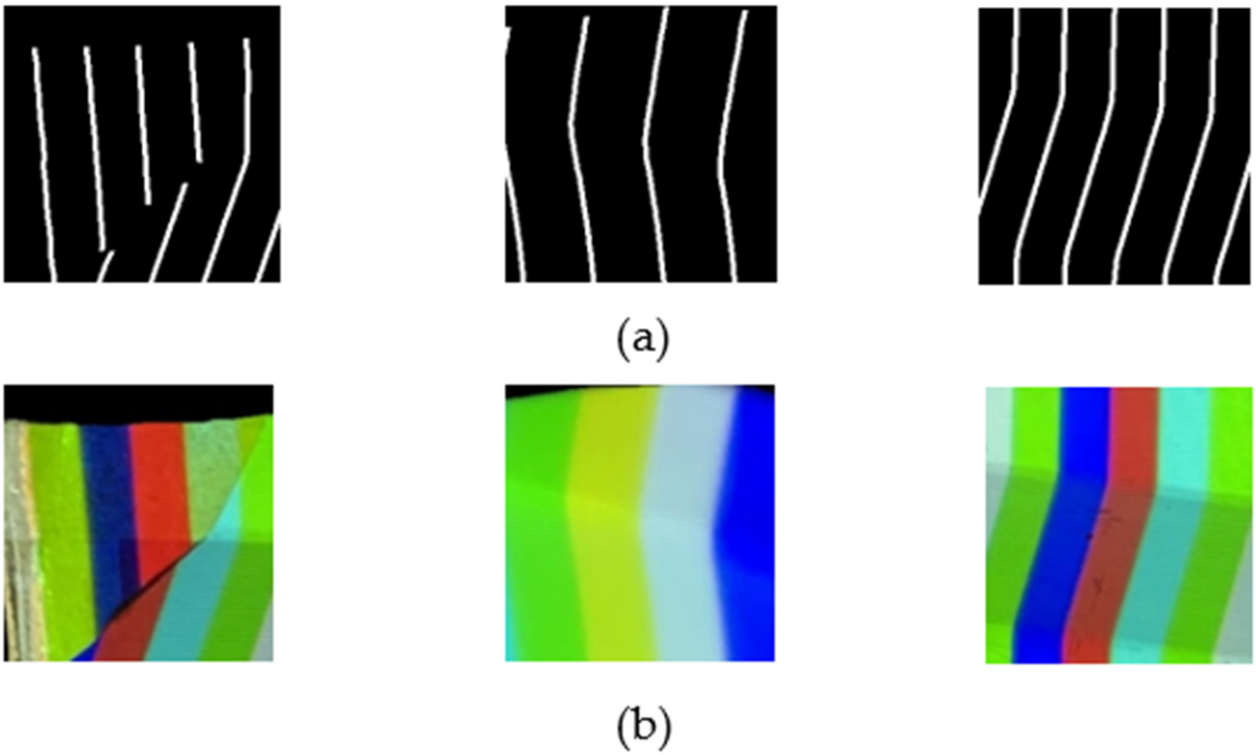


Figure 6. Illustration of input samples in the training set. (**a**) Labeled images; (**b**) Image patches.

3.3. Color Structured Light Stripe Edge Detection Method

As analyzed above, 3D imaging systems using color stripe patterns have many outstanding advantages. However, the influence of noise and the characteristic of the object surface make accurate color stripe edge detection difficult in many practical applications. In some situations, the edges are detected with a very large distance between themselves and the ones in the ground-truth images. Moreover, many wrong detections may occur because of the colored textures on the object's surface. These issues surely reduce the quality of 3D reconstruction. In this paper, the GAN-based method for color structured light stripe edge detection (Figure 7) includes the following steps:

(1) The GAN-based model for color structured light stripe edge detection is trained with the prepared training set and pre-set parameters. With the properties of the GANs, this model can significantly reduce the effect of noise and characteristics, even colored textures, of the object surface;
(2) Test images are preprocessed with a bilateral filter to reduce noise and preserve edges in the images. Similar to the images in the training set, these images are sequentially cut into patches of 128 × 128 pixels. These patches are then fed into the trained model for prediction;
(3) The predicted image patches consist mainly of two components: stripes (bright regions) and background (dark regions). They are sequentially merged together to obtain large images of the same size as the original test image;
(4) With such clean resulting images, a morphological skeletonization algorithm is applied to detect the centerlines in the images. The positions of these lines are equivalent to the positions of the left edges of the color stripes in the deformed pattern images.

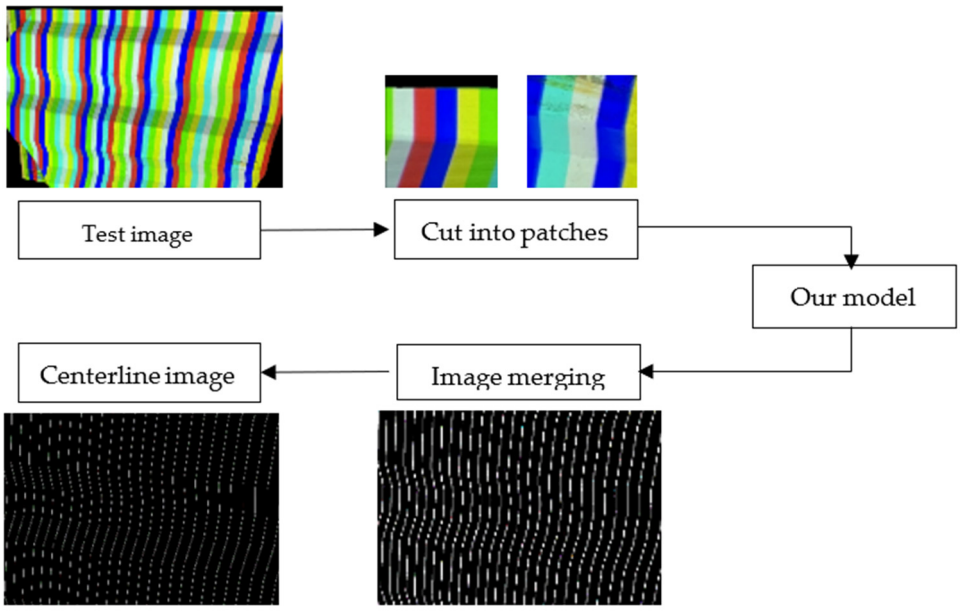


Figure 7. Block diagram of the color structured light stripe edge detection method.

Traditional edge detection methods for color stripe pattern images usually rely on some threshold to determine the existence of an edge in the images. However, the influence of noise and the complex characteristics of the objects' surface make many edges undetected or wrongly detected. To overcome these challenges, the color structured light stripe edge detection method proposed in this paper has the following improvements: (1) The specific dataset is designed with high quality; (2) the deeper network structure based on GANs with attention gates helps the model to learn more high-level features. Thus, the trained model can effectively reduce the effect of noise in the image and produce high-quality images, such as ground-truth images. As a result, the centerlines in the predicted images, corresponding to the edges in the color stripe pattern images, are accurately detected. In other words, with the proposed network, our method converts the task of detecting edges in color stripe pattern images into detecting centerlines in curved line images.

4. Experimental Results and Discussion

4.1. Experimental Setup

To achieve our goal, the system consisted of two main parts: the image-acquiring subsystem (IAS) and the data processing unit (DPU). The IAS consisted of a Canon 77D camera with a resolution of 6000 × 4000 pixels and a Canon REALiS SX7 projector with a resolution of 1400 × 1050 pixels. During the image acquisition, the projector and camera were placed 200 mm apart, while the working distance was 800 mm. The DPU was a desktop computer with an Intel Core i7-8700 microprocessor, 16 GB of RAM, and an NVIDIA GeForce RTX 2070 graphics card. In our experiments, two projection patterns were designed. The first one was a De Bruijn sequence-based color stripe pattern encoded with five colors in the HSV color space. To ensure uniqueness with a 1 × 3 property window in the entire pattern, the pattern consisted of only 66 stripes (Figure 8a). The second one was a multi-slit binary pattern, including 66 slits with a slit width of 3 pixels. It should be noted that the slits are located at positions corresponding to the left edges of the color stripes in the stripe pattern (Figure 8b). To demonstrate the robustness against the noise of the proposed method in practical applications, the experiments were conducted under room lighting conditions.

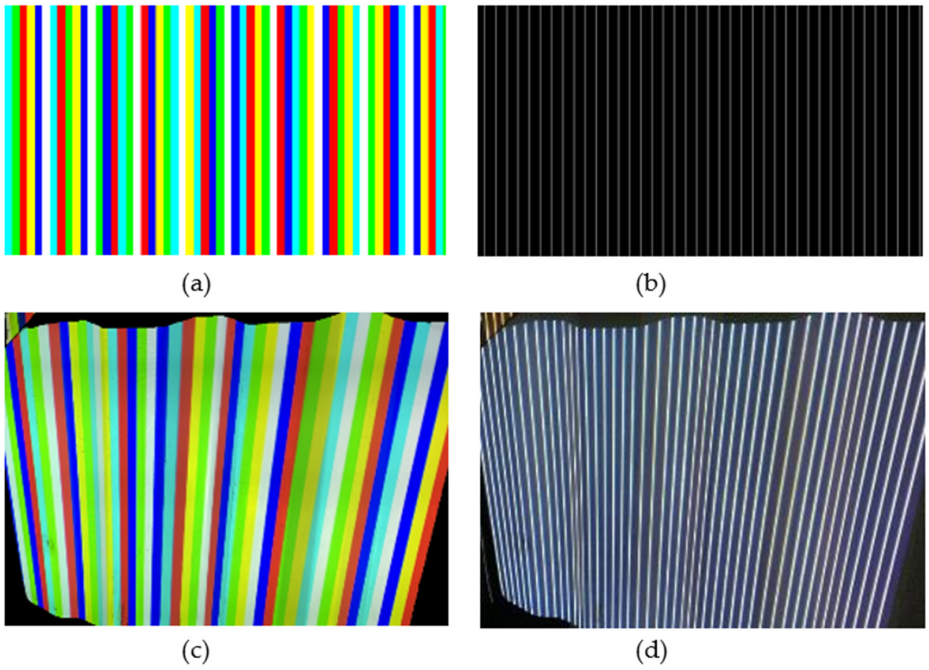


Figure 8. (**a**) Color stripe pattern and (**b**) multi-slit binary pattern of our method; (**c**,**d**) images of an object with the two patterns.

4.2. Training Process

In order for the model to converge quickly at the optimal point, it is extremely important to set the hyperparameters appropriately. According to experience from previous studies of color image semantic segmentation using GANs, we chose a batch size equal to 1 with a learning rate of 0.0001. In addition, the model was trained with 100 epochs with a step per epoch of 10,000.

Our model training process consists of two stages: (1) The input image is predicted with the generator, and (2) the discriminator evaluates the images predicted with the generator as real or fake by comparing them with real data. These two processes are not performed independently but occur in parallel during training (Figure 9).

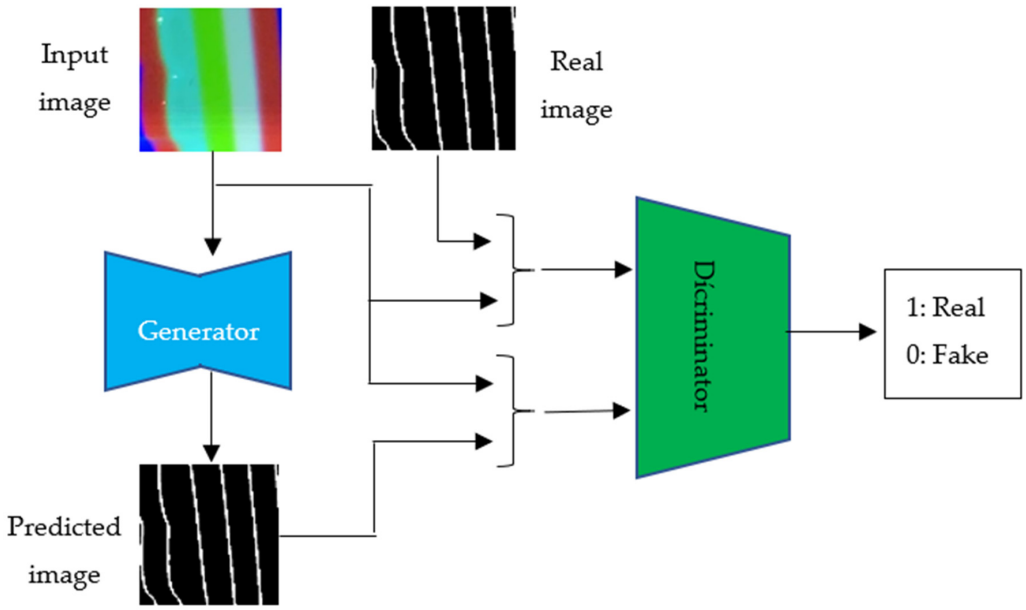


Figure 9. The training process of our model.

During training, the samples are continuously fed to the networks from the training set, where each of them consists of the sample itself and a corresponding label image. The generator extracts and then transfers many features of the input samples through the middle layers to the output. After that, the image generated by the generator is fed to the discriminator. Note that the real image and the generated image are both $128 \times 128 \times 3$. At this time, the resulting image is evaluated by the discriminator by comparing the pair of images predicted by the generator with the real data. According to the value of the loss function and the accuracy obtained after each iteration, the network parameters of each layer are continuously adjusted. In our networks, residual blocks help the model to solve the vanishing gradient problem, thus making the model converge faster at the optimal point. With attention gates, the skip connections in the generator can actively suppress activations in irrelevant regions and reduce the number of redundant features obtained. The training process is performed in multiple iterations over the entire training dataset. Finally, the model for color structured light stripe edge detection is obtained.

4.3. Segmentation Performance Evaluation of the Proposed Network

After the model is trained, test images are used to evaluate the semantic segmentation performance of the model. Note that these images are all cropped to the size of 1792×2560 pixels for ease of evaluation. Similar to the training process, the test images are also sequentially cut into small patches. Therefore, the goal of the proposed model is to individually segment each image patch. To evaluate the reliability of the model, we used two indicators, Precision (Pre) and Recall (Re), as the evaluation metrics:

$$\text{Pre} = \frac{TP}{TP + FP}, \tag{1}$$

$$\text{Re} = \frac{TP}{TP + FN}, \tag{2}$$

where true positive (*TP*) and true negative (*TN*) indicate the pixels properly classified as line and background, respectively, whereas the false positive (*FP*) and false negative (*FN*) denote misclassified pixels—background predicted as line and line predicted as background, respectively. Table 1 shows the semantic segmentation performance of the proposed model with each test image.

Table 1. Segmentation performance of our network with different materials (Unit: Pixels).

Object	TP	FP	FN	TN	Pre (%)	Re (%)
Colored textured plastic box	410,466	19,199	19,001	4,138,854	95.53	95.58
Wave metal sheet	423,066	11,872	12,952	4,139,630	97.27	97.03
Porcelain dish	411,728	12,032	11,153	4,152,607	97.16	97.36
Colored textured carton box	388,841	20,373	28,854	4,149,452	95.02	93.09
Open book	421,753	11,370	9447	4,144,950	97.37	97.81

It can be seen in Table 1 that the proposed method worked very well with objects with a fairly homogeneous surface (wave metal sheet, porcelain dish, and open book). However, there are still a few patches with lines unsegmented or falsely segmented in images of the objects with a multi-colored textured surface. After the image patches are predicted with the proposed model, they are merged together into large images. Finally, a morphological skeletonization algorithm is performed to obtain the centerline images (Figure 10).

To further evaluate the image segmentation performance of the proposed model, we compared it with four other GANs-based deep learning models, including Pix2pix [41], Res-Unet-based GAN (RU-GAN) [42], Upsample residual channel-wise attention Generative Adversarial Networks (URCA-GAN) [25], and the proposed networks without attention

gates (Proposed w/o AG). In addition to the above two metrics, two more metrics, including Accuracy (Acc) and the F1 score, are defined in Equations (3) and (4).

$$\text{Acc} = \frac{TP + TN}{TP + FP + TN + FN}, \tag{3}$$

$$\text{F1} = \frac{2 * TP}{2 * TP + FP + FN}, \tag{4}$$

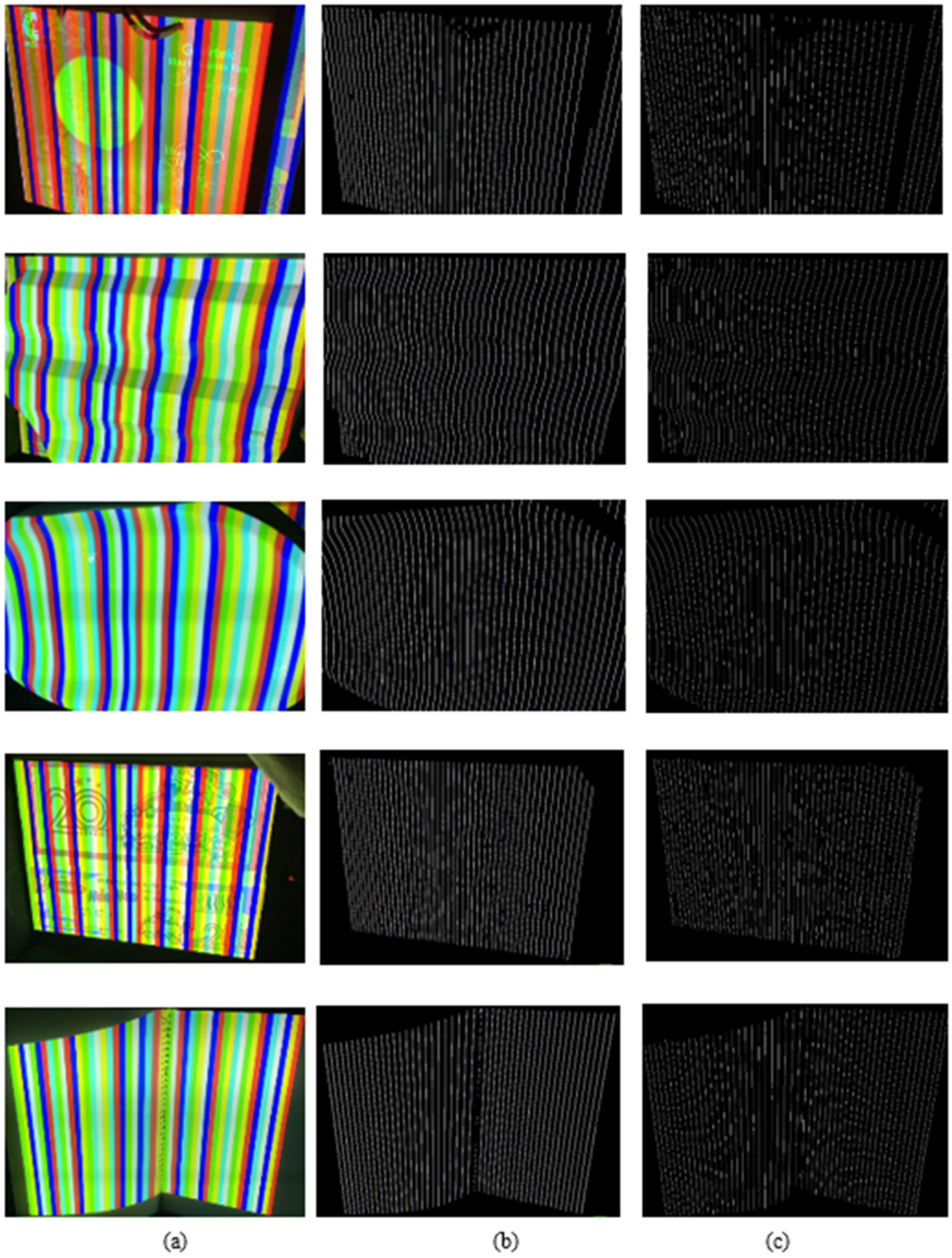


Figure 10. Segmentation results with the proposed method: (**a**) original test image; (**b**) predicted image; (**c**) centerline image.

In this experiment, we used an image representing the images in the test set, which was an image of a paper box with a complex color pattern. The comparison of the image segmentation performance of the above networks is presented in Table 2. The segmentation

results of the image of a colored textured carton box with these methods are shown in Figure 11.

Table 2. Quantitative comparison of segmentation performance of different methods (Unit: Pixels).

Method	TP	FP	FN	TN	Acc (%)	Pre (%)	Re (%)	F1 (%)
Pix2pix	289,185	99,553	88,510	4,110,272	95.90	74.39	76.57	75.46
RU-GAN	291,195	72,268	66,500	4,157,557	96.98	80.12	81.41	80.76
URCA-GAN	378,496	48,021	39,199	4,121,804	98.10	88.74	90.62	89.67
Proposed w/o AG	383,662	59,401	24,033	4,120,424	98.18	86.59	94.11	90.19
Proposed	388,841	20,373	28,854	4,149,452	98.93	95.02	93.09	94.05

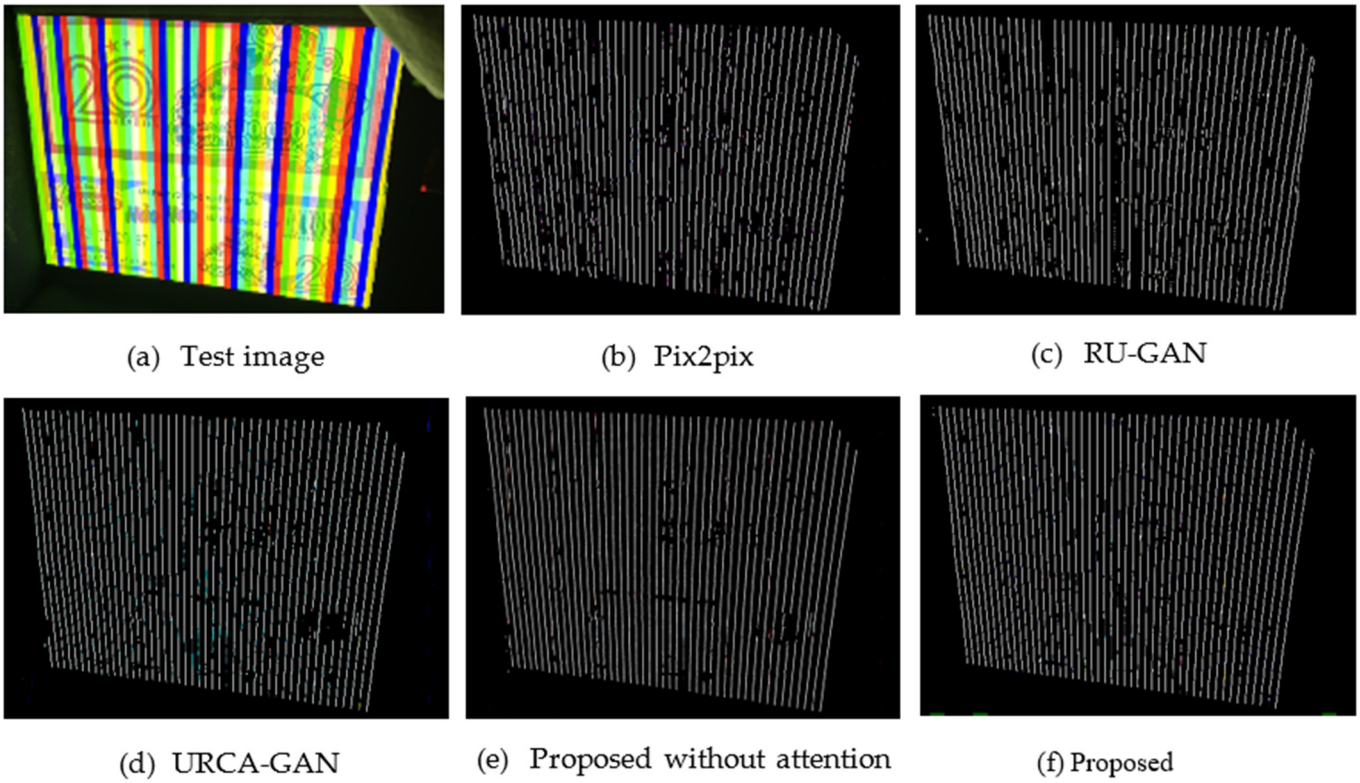


Figure 11. Segmentation results of the image of a colored textured carton box with different methods.

As can be seen in Figure 11 and Table 2, by replacing plain neural blocks with residual blocks, RU-GAN achieved higher accuracy than Pix2pix. URCA-GAN achieved better segmentation results; there were only a few areas where the texture, which was similar to the color stripes, was wrongly segmented. Although the proposed method without attention gates achieved better segmentation performance than the previous methods, there were still some broken lines and a few wrongly detected regions. Meanwhile, the proposed networks can address most of these issues, thus achieving the highest segmentation performance.

4.4. Pixel Location Accuracy

Figure 12 shows a comparison of the location accuracy of the color stripe edges detected with the proposed method and two other methods, including a modified Canny edge detector [16] and a modified Sobel edge detector [43]. Notice that the green lines indicate the edges in the ground-truth image, and the red lines indicate the edges detected by the three methods. The experimental results show that the methods in [16,43] could detect almost all color stripe edges in the images. There were, however, some edges that were wrongly detected (Figure 12b,c). The cause of this occurrence is that these methods

cannot reduce the effect of surface characteristics such as colored texture, inter-reflection (porcelain dish), and rust stains (metal sheet). Meanwhile, the above problems are well-handled with the proposed method (Figure 12d).

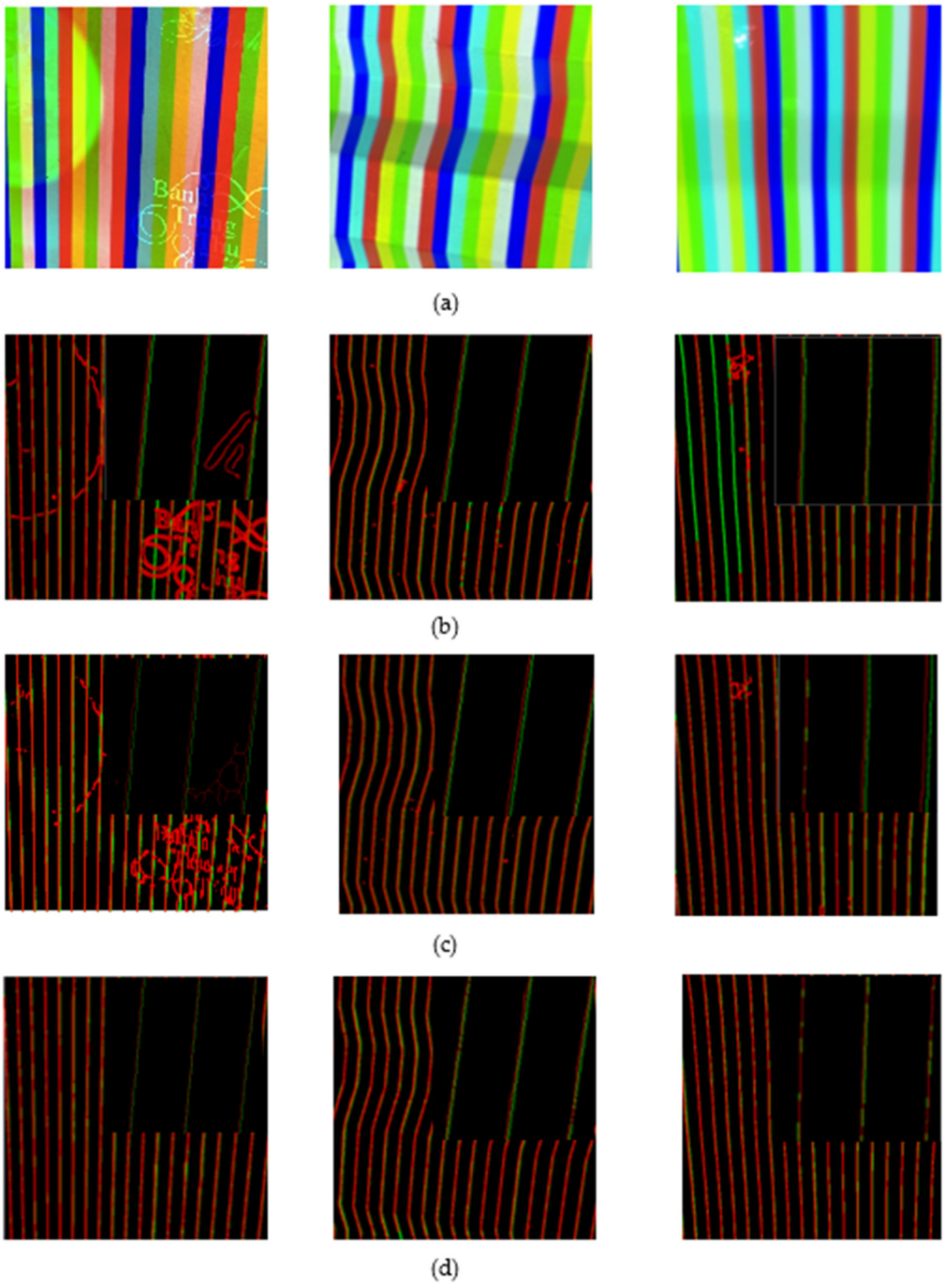


Figure 12. Comparison of pixel location accuracy of color stripe edge detection methods: (**a**) part of the original image; (**b**) results with modified Canny in [16], (**c**) with modified Sobel in [43], and (**d**) with the proposed method.

To evaluate the pixel location accuracy of the edges detected with different methods, the images, including the edge images obtained with the three methods and the corresponding ground-truth images, were cropped into small images with a size of 1000 × 1000 pixels in the experiments. Notice that the edges of the color stripes in the ground-truth images are 1 pixel wide. As observed in Figure 12, the edges detected by different methods do not

coincide with the edges in the ground-truth image due to the influence of the characteristics of the object's surface. This means the number of correctly detected pixels corresponding to TPs in the above metrics may be very small, even zero. Therefore, the above metrics cannot be used to evaluate the local pixel accuracy of the different methods. Thus, the experiments were conducted using the mean square error (*MSE*), which is defined in Equation (5), as the evaluation metric:

$$MSE = \frac{\sum_{i=1}^{N} (C_d - C_g)^2}{N} \tag{5}$$

where C_d is the coordinates of the points on the edges detected with different methods, C_g is the coordinates of the points on the edges in the ground-truth image, and N is the number of points in each ground-truth image.

In order to confidently conclude that the color structure light stripe edge detection method proposed in this paper is better than the methods in [16,43], we performed a statistical significance test with a comparison of MSE, which was achieved with these methods, as shown in Table 3.

Table 3. Comparison of pixel location accuracy of color stripe edge detection methods (Unit: Pixels).

Object	Method in [16] (C)	Method in [43] (S)	Proposed Method (P)	MSE Difference between C & P	MSE Difference between S & P
Colored textured plastic box	0.0383	0.0350	0.0302	0.0081	0.0048
Waved metal sheet	0.0359	0.0332	0.0286	0.0073	0.0046
Porcelain dish	0.0331	0.0301	0.0275	0.0056	0.0026
Colored textured carton box	0.0437	0.0412	0.0306	0.0131	0.0106
Open book	0.0343	0.0315	0.0271	0.0072	0.0044
Sample mean				0.0082	0.0054
Sample variant				0.0028	0.0031
Standard error				0.0012	0.0014

Using the data in Table 3, a 90% confidence interval for the true mean differences can be obtained. The approximate $100\%(1-\alpha)$ confidence interval for τ is defined as Equation (6):

$$\overline{\tau} - g_{\frac{\alpha}{2},(n-1)}\Omega(\overline{\tau}) \leq \tau \leq \overline{\tau} + g_{\frac{\alpha}{2},(n-1)}\Omega(\overline{\tau}) \tag{6}$$

where $\overline{\tau}$ denotes a sample mean of τ based on a sample of size n. In this case $n = 5$. $\Omega(\overline{\tau})$ is the standard error of $\overline{\tau}$, and $g_{\alpha/2,(n-1)}$ indicates the $100\%(1-\alpha)$ percentage point of a t-distributed with $n-1$. Thus, $g_{\alpha/2,(n-1)} = g_{0.05,4} = 2.132$, which was obtained from the t-distribution table.

With the data in Table 3, a 90% confidence interval for the MSE difference between the method in [16] and the proposed method is given by

$$0.0082 \pm 2.132 * 0.0012 \text{ or} \tag{7}$$

$$0.0082 \pm 0.0026 \tag{8}$$

So, the 90% confidence interval is (0.0056, 0.0108).

Thus, we are 90% confident that the average MSE difference between the method in [16] and the proposed method is between 0.0056 and 0.0108.

Similarly, a 90% confidence interval for the MSE difference between the method in [43] and the proposed method is given by

$$0.0054 \pm 2.132 * 0.0014 \text{ or} \tag{9}$$

$$0.0054 \pm 0.0029 \tag{10}$$

So, the 90% confidence interval is (0.0025, 0.0083).

Thus, we are 90% confident that the average MSE difference between the method in [43] and the proposed method is between 0.0025 and 0.0083.

It can be seen in Table 3 that the MSEs achieved with the methods in [16,43] were pretty large, especially for the objects with complex surface characteristics, such as the plastic box and the carton box, while deviations of the MSEs with different objects achieved by the proposed method were small. Furthermore, the 90% confidence intervals in Equations (8) and (10) were always above zero. This means that the proposed method achieved higher pixel location accuracy compared to its counterparts.

5. Conclusions

In this paper, a color structure light stripe edge detection method based on generative adversarial networks, which is named horizontal elastomeric attention residual Unet-based GAN (HEAR-GAN), is proposed. Additionally, a specific dataset was designed for training and testing the model. Firstly, a De Bruijn sequence-based color stripe pattern and a multi-slit binary pattern were designed. After that, the images of objects with these patterns were captured under room lighting conditions. Among them, the images with a multi-slit pattern after being preprocessed were selected as the ground-truth images. The images with a color stripe pattern and their respective ground-truth images were cut into patches. Finally, some data augmentation techniques were applied to these patches, and then they were divided into a training set and a validation set. Using the images of objects with a multi-slit pattern as ground-truth images reduced not only the labor of manual annotation but also enhanced the quality of the training set. Patches of the test images, after being predicted with the trained model, were merged together into large images. The centerlines in these resulting images were detected as the edges in the color structured light stripe images. The experimental results prove that the proposed method can overcome the challenges from ambient light noise and the effect of the complex characteristics of the object surface so that most of the edges in the images of a color stripe pattern are detected. Additionally, the comparison with other GANs-based networks shows that our network had a higher color stripe segmentation performance than its counterparts. Moreover, the results of a statistical significance test with the MSE of different methods demonstrate that the proposed method achieved higher pixel local accuracy than the manual-based edge detection methods, i.e., the modified Canny and modified Sobel edge detectors.

Author Contributions: Methodology, C.X.; Validation, M.H.; Investigation, D.P.; Resources, D.P.; Data curation, D.P. and M.H.; Writing—original draft preparation, D.P. and M.H.; Writing—review and editing, C.X.; Supervision, C.X. 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, grant number 62073128.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The participants of this study did not give written consent for their data to be shared publicly, so supporting data is not available.

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

References

1. Pages, J.; Salvi, J.; Forest, J. A new optimised De Bruijn coding strategy for structured light patterns. In Proceedings of the 17th International Conference on Pattern Recognition, San José, CA, USA, 26–26 August 2004; Volume 4, pp. 284–287.
2. Donlic, M.; Petkovic, T.; Pribanic, T. 3D surface profilometry using phase shifting of De Bruijn pattern. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 963–971.
3. Petković, T.; Pribanić, T.; Đonlić, M. Single-shot dense 3D reconstruction using self-equalizing De Bruijn sequence. *IEEE Trans. Image Process.* **2016**, *25*, 5131–5144. [CrossRef]

4. Je, C.; Lee, K.H.; Lee, S.W. Multi-projector color structured-light vision. *Signal Process. Image Commun.* **2013**, *28*, 1046–1058. [CrossRef]
5. Lee, K.H.; Je, C.; Lee, S.W. Color-stripe structured light robust to surface color and discontinuity. In Proceedings of the Asian Conference on Computer Vision, Tokyo, Japan, 18–22 November 2007; pp. 507–516.
6. Zhang, C.; Xu, J.; Xi, N.; Zhao, J.; Shi, Q. A robust surface coding method for optically challenging objects using structured light. *IEEE Trans. Autom. Sci. Eng.* **2014**, *11*, 775–788. [CrossRef]
7. Rocchini, C.M.P.P.C.; Cignoni, P.; Montani, C.; Pingi, P.; Scopigno, R. A low cost 3D scanner based on structured light. In *Computer Graphics Forum*; Wiley-Blackwell: Hoboken, NJ, USA, 2001; Volume 20, pp. 299–308.
8. Dhankhar, P.; Sahu, N. A review and research of edge detection techniques for image segmentation. *Int. J. Comput. Sci. Mob. Comput.* **2013**, *2*, 86–92.
9. Magnier, B. Edge detection: A review of dissimilarity evaluations and a proposed normalized measure. *Multimed. Tools Appl.* **2018**, *77*, 9489–9533. [CrossRef]
10. Owotogbe, J.S.; Ibiyemi, T.S.; Adu, B.A. Edge detection techniques on digital images-a review. *Int. J. Innov. Sci. Res. Technol.* **2019**, *4*, 329–332.
11. Azeroual, A.; Afdel, K. Fast image edge detection based on faber schauder wavelet and otsu threshold. *Heliyon* **2017**, *3*, e00485. [CrossRef]
12. Lopez-Molina, C.; De Baets, B.; Bustince, H.; Sanz, J.A.; Barrenechea, E. Multiscale edge detection based on Gaussian smoothing and edge tracking. *Knowl. Based Syst.* **2013**, *44*, 101–111. [CrossRef]
13. Ahmad, I.; Moon, I.; Shin, S.J. Color-to-grayscale algorithms effect on edge detection—A comparative study. In Proceedings of the 2018 International Conference on Electronics, Information, and Communication (ICEIC), Honolulu, HI, USA, 24–27 January 2018; IEEE Press: New York, NY, USA, 2018; pp. 1–4.
14. Cheon, Y.; Lee, C. Color edge detection based on Bhattacharyya distance. In Proceedings of the 14th International Conference on Informatics in Control, Automation and Robotics (ICINCO), Madrid, Spain, 26–28 July 2017; Volume 2, pp. 368–371.
15. Xin, G.; Ke, C.; Xiaoguang, H. An improved Canny edge detection algorithm for color image. In Proceedings of the IEEE 10th International Conference on Industrial Informatics, Beijing, China, 25–27 July 2012; pp. 113–117.
16. Qin, X. A modified Canny edge detector based on weighted least squares. *Comput. Stat.* **2021**, *36*, 641–659. [CrossRef]
17. Rashid, H.; Tanveer, M.A.; Khan, H.A. Skin lesion classification using GAN based data augmentation. In Proceedings of the 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Berlin, Germany, 23–27 July 2019; IEEE Press: New York, NY, USA, 2019; pp. 916–919.
18. Yang, S.; Jiang, L.; Liu, Z.; Loy, C.C. Unsupervised Image-to-Image Translation with Generative Prior. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; IEEE Press: New York, NY, USA, 2022; pp. 18332–18341.
19. Guo, Y.; Liu, Y.; Georgiou, T.; Lew, M.S. A review of semantic segmentation using deep neural networks. *Int. J. Multimed. Inf. Retr.* **2018**, *7*, 87–93. [CrossRef]
20. Zhang, Y.; Miao, S.; Mansi, T.; Liao, R. Unsupervised X-ray image segmentation with task driven generative adversarial networks. *Med. Image Anal.* **2020**, *62*, 101664. [CrossRef]
21. Zhang, K.; Zhang, Y.; Cheng, H.D. CrackGAN: Pavement crack detection using partially accurate ground truths based on generative adversarial learning. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 1306–1319. [CrossRef]
22. Yu, Y.; Gong, Z.; Zhong, P.; Shan, J. Unsupervised representation learning with deep convolutional neural network for remote sensing images. In Proceedings of the International Conference on Image and Graphics, Shanghai, China, 13–15 September 2017; Springer: Berlin/Heidelberg, Germany, 2017; pp. 97–108.
23. Zhu, J.-Y.; Park, T.; Isola, P.; Efros, A.A. Unpaired image-to-image translation using cycle-consistent adversarial networks. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2223–2232.
24. Zhang, K.; Zhang, Y.; Cheng, H.D. Self-supervised structure learning for crack detection based on cycle-consistent generative adversarial networks. *J. Comput. Civ. Eng.* **2020**, *34*, 04020004. [CrossRef]
25. Nie, X.; Ding, H.; Qi, M.; Wang, Y.; Wong, E.K. Urca-gan: Upsample residual channel-wise attention generative adversarial network for image-to-image translation. *Neurocomputing* **2021**, *443*, 75–84. [CrossRef]
26. Pham, D.; Ha, M.; San, C.; Xiao, C.; Cao, S. Accurate stacked-sheet counting method based on deep learning. *J. Opt. Soc. Am. A* **2020**, *37*, 1206–1218. [CrossRef]
27. Poma, X.S.; Riba, E.; Sappa, A. Dense extreme inception network: Towards a robust cnn model for edge detection. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 2–5 March 2020; pp. 1923–1932.
28. Dhar, P.; Guha, S.; Biswas, T.; Abedin, M.Z. A system design for license plate recognition by using edge detection and convolution neural network. In *2018 International Conference on Computer, Communication, Chemical, Material and Electronic Engineering (IC4ME2), Rajshahi, Bangladesh, 8–9 February 2018*; IEEE Press: New York, NY, USA, 2018; pp. 1–4.
29. Li, X.; Jiao, H.; Wang, Y. Edge detection algorithm of cancer image based on deep learning. *Bioengineered* **2020**, *11*, 693–707. [CrossRef]
30. Hou, S.M.; Jia, C.L.; Wanga, Y.B.; Brown, M. A review of the edge detection technology. *Sparklinglight Trans. Artif. Intell. Quantum Comput. (STAIQC)* **2021**, *1*, 26–37. [CrossRef]

31. Cai, S.; Wu, Y.; Chen, G. A Novel Elastomeric UNet for Medical Image Segmentation. *Front. Aging Neurosci.* **2022**, *14*, 841297. [CrossRef]
32. Xiao, X.; Lian, S.; Luo, Z.; Li, S. Weighted res-unet for high-quality retina vessel segmentation. In Proceedings of the 2018 9th International Conference on Information Technology in Medicine and Education (ITME), Hangzhou, China, 19–21 October 2018; IEEE Press: New York, NY, USA; pp. 327–331.
33. Luo, Z.; Zhang, Y.; Zhou, L.; Zhang, B.; Luo, J.; Wu, H. Micro-vessel image segmentation based on the AD-UNet model. *IEEE Access* **2019**, *7*, 143402–143411. [CrossRef]
34. Wang, Z.; Wang, L.; Duan, S.; Li, Y. An image denoising method based on deep residual GAN. *J. Phys. Conf. Ser.* **2020**, *1550*, 032127. [CrossRef]
35. Wolterink, J.M.; Leiner, T.; Viergever, M.A.; Išgum, I. Generative Adversarial Networks for noise reduction in low-dose CT. *IEEE Trans. Med. Imaging* **2017**, *36*, 2536–2545. [CrossRef]
36. Alsaiari, A.; Rustagi, R.; Thomas, M.M.; Forbes, A.G. Image denoising using a generative adversarial network. In Proceedings of the 2019 IEEE 2nd International Conference on Information and Computer Technologies (ICICT), Kahului, HI, USA, 14–17 March 2019; IEEE Press: New York, NY, USA, 2019; pp. 126–132.
37. Je, C.; Lee, S.W.; Park, R.H. Colour-stripe permutation pattern for rapid structured-light range imaging. *Opt. Commun.* **2012**, *285*, 2320–2331. [CrossRef]
38. Li, Y.; Wang, Z. 3D reconstruction with single-shot structured light RGB line pattern. *Sensors* **2021**, *21*, 4819. [CrossRef]
39. Je, C.; Choi, K.; Lee, S.W. Green-blue stripe pattern for range sensing from a single image. *arXiv* **2017**, arXiv:1701.02123.
40. Ha, M.; Pham, D.; Xiao, C. Accurate feature point detection method exploiting the line structure of the projection pattern for 3D reconstruction. *Appl. Opt.* **2021**, *60*, 2926–2937. [CrossRef]
41. Isola, P.; Zhu, J.Y.; Zhou, T.; Efros, A. AImage-to-image translation with conditional adversarial networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; IEEE Press: New York, NY, USA, 2017; pp. 1125–1134.
42. Lv, J.; Wang, P.; Tong, X.; Wang, C. Parallel imaging with a combination of sensitivity encoding and generative adversarial networks. *Quant. Imaging Med. Surg.* **2020**, *10*, 2260. [CrossRef]
43. Tian, R.; Sun, G.; Liu, X.; Zheng, B. Sobel edge detection based on weighted nuclear norm minimization image denoising. *Electronics* **2021**, *10*, 655. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.