electronics

MDPI

Article

Depth Estimation of Monocular PCB Image Based on Self-Supervised Convolution Network

Zedong Huang, Jinan Gu *, Jing Li, Shuwei Li and Junjie Hu

School of Mechanical Engineering, Jiangsu University, Zhenjiang 212000, China; huangzedongju@163.com (Z.H.); lijing31431@sina.com (J.L.); toughricqf@outlook.com (S.L.); hjj_melon@hotmail.com (J.H.)
* Correspondence: gjnan@ujs.edu.cn

Abstract: To improve the accuracy of using deep neural networks to predict the depth information of a single image, we proposed an unsupervised convolutional neural network for single-image depth estimation. Firstly, the network is improved by introducing a dense residual module into the encoding and decoding structure. Secondly, the optimized hybrid attention module is introduced into the network. Finally, stereo image is used as the training data of the network to realize the end-to-end single-image depth estimation. The experimental results on KITTI and Cityscapes data sets show that compared with some classical algorithms, our proposed method can obtain better accuracy and lower error. In addition, we train our models on PCB data sets in industrial environments. Experiments in several scenarios verify the generalization ability of the proposed method and the excellent performance of the model.

Keywords: unsupervised learning; depth estimation; hybrid attention mechanism; residual dense

Citation: Huang, Z.; Gu, J.; Li, J.; Li, S.; Hu, J. Depth Estimation of Monocular PCB Image Based on Self-Supervised Convolution Network. *Electronics* **2022**, *11*, 1812. https://doi.org/10.3390/electronics11121812

Academic Editor: Gwanggil Jeon

Received: 6 April 2022
Accepted: 23 May 2022
Published: 7 June 2022

1. Introduction

The proposal of "made in China 2025" indicates that China's "industry 4.0" era is coming [1,2]. In this context, the intelligent industrial robot has become an inevitable development trend. In recent years, machine vision technology has developed rapidly, which can meet the requirements of high-precision positioning in industrial scenes. The introduction of machine vision dramatically expands the application field of industrial robots and has essential significance for the development of industrial automation. At present, some important links on the assembly line of the integrated circuit board are still completed by skilled workers, such as inserting the pins of electronic components (as shown in Figure 1) into corresponding holes (as shown in Figure 2), quality control of finished products, etc. However, there are some problems in manual insertion, such as high cost, low efficiency, and poor quality.

In order to realize the automation of circuit board assembly, besides the pose estimation of parts, an important premise is the high-precision 3D reconstruction of the scene. Stereo matching algorithm is its core; only by obtaining the accurate two-dimensional information of matching points can we reconstruct the accurate three-dimensional scene. Stereo vision uses binocular cameras to obtain the left and right images of the target object in the same scene at different positions; through the stereo matching method, the homonymous points of the left and right images are matched pixel by pixel, and the disparity of the homonymous points is calculated to obtain the depth information of the object in the three-dimensional scene. At present, most stereo matching algorithms train the depth network in a supervised way.

Figure 1. Electronic components.

Figure 2. Manual Assembly.

However, the supervised training method needs to use the image data set with real depth information (ground truth) as the target training. In monocular image depth estimation, it is difficult to obtain the real depth data per pixel on a large scale. Therefore, some scholars have proposed a self-supervised monocular depth estimation method. This method only uses the image itself as the monitoring information when training the depth network without the explicit ground truth depth of the image. The self-supervised method reduces the requirements of the training data set and improves the adaptability and robustness of depth estimation. However, due to the problems of occlusion between objects and high reflectance in the existing data sets, the current self-supervised methods not only fail to make good use of the context information in the scene, but also are affected by object occlusion and inaccurate contour, and the result of depth estimation will be worse. Therefore, it is necessary to design a more effective depth estimation network structure. It should obtain more useful supervising information from the image, optimize the depth estimation

result, and achieve a more reasonable measurement; this is also the main problem faced by using the self-supervised convolution network to estimate the depth of a single image.

These show the application value of this study; the design of neural network structure has many technical difficulties, which is the main reason for the low accuracy of PCB component reconstruction. These difficulties include making better use of the contextual information in the scene, better enhancing the spread of features, and maintaining a balance between accuracy and time consumption.

This paper presents a method of depth estimation of a single PCB image based on a self-supervised convolution network. The developed method was arduously tested and compared to commonly used methodologies. The structure of this paper is as follows: in the second section, we introduce the supervised learning method and self-supervised learning method in the field of depth estimation of a single image. The third section explains the network model structure of a self-supervised residual dense network. The fourth section describes the experimental process; discusses the results of training, verification, and testing; and compares the methods in this paper with the latest ones. Section 5 summarizes the research and future work.

2. Related Work

In recent years, deep learning has gradually entered people's attention. In single-image depth estimation, neural network-based methods emerge endlessly. These methods can be divided into supervised learning methods and unsupervised learning methods.

2.1. Single-Image Depth Estimation Method Based on Supervised Neural Network

Eigen et al. [3] first applied the CNN network to monocular image depth estimation task. The model proposed by them is divided into two steps. Firstly, convolutional neural network is used for coarse-scale global prediction, and then local refinement is used to obtain a better depth map. However, this method requires the superposition of two networks and does not realize end-to-end training. Liu et al. [4] first used a convolutional neural network to convert the depth estimation problem into a continuous conditional random field learning problem. They proposed an equivalent complete convolution network and super pixel method, which increased the depth estimation speed by nearly ten times [5]. Laina et al. [6] first applied the complete convolution network based on the residual network to single-image depth estimation. At the same time, to improve the image resolution, a new up-sampling method was proposed. By introducing the inverse Huber loss function, the network has a shorter training time and better real-time performance. Li et al. [7] proposed a dual-flow network based on vgg-16; the network is divided into depth flow and gradient flow to extract depth information and depth gradient information of images, respectively. The feature fusion module and thinning module are used to extract features further. Finally, the depth map is obtained through the fusion module. Kendall et al. [8] proposed an end-to-end network based on stereo image pairs. The network has a high-level feature representation called cost-volume and uses 3D convolution to fuse the image information with the cost value. By minimizing the cost value, the disparity value is regressed, and then the depth map of the image is obtained. Fu et al. [9] proposed an idea of discretization of the distance between growth. By discretizing the depth, the depth estimation problem is transformed into an ordinal regression problem. The corresponding ordinal regression loss function trains the network to obtain outstanding depth estimation results.

2.2. Single-Image Depth Estimation Method Based on Self-Supervised Neural Network

Garg et al. [10] transformed the problem of depth estimation into image reconstruction, used binocular stereo image pairs to train the network, and realized the unsupervised training of the network by minimizing the optical difference between the actual reconstructed right image. Godard et al. [11] also used stereo images as training supervision, synthesized images by introducing the consistency of left and right disparity and estimating the stereo

disparity according to the depth network, and then trained the neural network by comparing the gray difference between the input stereo image and the synthetic image to complete the depth estimation of a single image. Zhou et al. [12] used a monocular image sequence as training data and used a depth estimation network and a pose estimation network to estimate image depth and obtain camera moving attitude, respectively. Zhang et al. [13] used binocular video sequences to train the network, obtained time information from adjacent frames of the same view video and received spatial data from different view images simultaneously, and then fused the two to achieve depth estimation. Yin et al. [14] proposed Geo-Net, which is composed of two sub-networks. The two sub-networks are trained jointly, and the depth information and pose information are estimated simultaneously. The consistency constraint between them realizes the depth estimation.

The self-supervised method reduces the requirement of the training data set and improves the adaptability and robustness of depth estimation, but the object contour in the depth estimation map is fuzzy and the boundary is not clear. Aiming at the problem of inaccurate depth estimation values caused by fuzzy contour information in single-image depth estimation, an optimized self-supervised convolution network method for single-image depth estimation is proposed in this paper. Firstly, by fusing the residual unit and dense unit, a residual dense unit suitable for single-image depth estimation is proposed. Then, several residual dense units are used to form a residual dense module, which is applied to the codec structure with jump connection. The network has no full connection layer, which reduces the parameters and the requirements for the size of the test image. In addition, the hybrid attention module is used to improve the depth estimation network, and the attention mechanism is used to make more effective use of the context information within and between objects to enhance the feature extraction ability of the model. In the training stage, the calibrated stereo images are used as training data for self-supervised training, and the trained network can estimate the disparity between stereo images. In the test phase, the depth information of the scene in the image can be calculated by inputting a single image according to the camera calibration parameters and the predicted disparity.

3. Method

By introducing the optimized residual density model and mixed attention model, we propose a new single-image depth estimation neural network. In the case of self-supervising, a series of stereo image pairs are used to train the network; the following parts of the method are introduced in detail.

3.1. Residual Dense Module

The residual dense module used in this paper is obtained by combining the residual network with the dense network, adding the identity mapping of the residual network module based on dense network module, and then through some improvements. Its specific structure is shown in Figure 3.

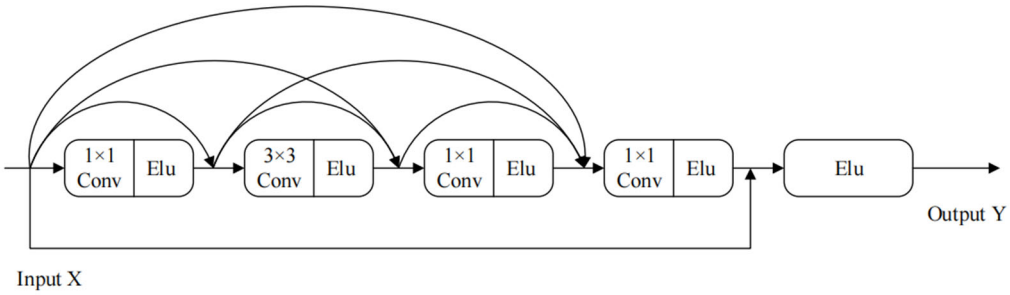

Figure 3. Residual Dense Unit.

Similar to the residual structure, the residual dense primitive is divided into a direct mapping part and a residual mapping part. The residual mapping part is divided into four convolution layers; the first layer uses 1×1 convolution core, the second layer uses

3 × 3 convolution core, the third layer uses 1 × 1 convolution core, and the fourth layer uses 1 × 1 convolution core. The first three steps are Formula 1 and the fourth step is Formula 2; each convolution layer selects ELU [15] as the activation function, as shown in Formulas (1)–(5). The dense connection is integrated into the residual mapping; after the dense connection of three convolution layers, the input features will pass through the fourth convolution layer and then add to the direct mapping. Finally, the final output is obtained through the ELU activation function. Let x be the input of the residual dense primitive, $\boldsymbol{F_i}$ be the $\boldsymbol{i}$th convolution operation, and $\boldsymbol{I_i}$ be the input of the $\boldsymbol{i}$th convolution layer.

$$\boldsymbol{I_1} = \boldsymbol{x} \tag{1}$$

$$\boldsymbol{I_2} = \boldsymbol{concat}[\boldsymbol{x}, \boldsymbol{F_1}(\boldsymbol{I_1})] \tag{2}$$

$$\boldsymbol{I_3} = \boldsymbol{concat}[\boldsymbol{x}, \boldsymbol{F_1}(\boldsymbol{I_1}), \boldsymbol{F_2}(\boldsymbol{I_2})] \tag{3}$$

$$\boldsymbol{I_4} = \boldsymbol{concat}[\boldsymbol{x}, \boldsymbol{F_1}(\boldsymbol{I_1}), \boldsymbol{F_2}(\boldsymbol{I_2}), \boldsymbol{F_3}(\boldsymbol{I_3})] \tag{4}$$

$$\boldsymbol{Out} = \boldsymbol{concat}[\boldsymbol{x}, \boldsymbol{I_4}] \tag{5}$$

Among them, concat is the connection operation, and Out is the final output of the residual dense unit. In the residual dense unit, the input of each convolution layer is the output characteristics of all previous convolution layers in the unit and the input characteristics of the unit. Through the integration of the residual network units and the dense network units that are dense unit residual, we not only can avoid the degradation of the neural network, but also make sure that the layers of the network extracted features are fully used, which makes the local characteristics of the network module show better to extract the features and also solves the problem of the dense network of hard training.

3.2. Hybrid Attention Module

Due to the poor image quality of the PCB board, high density of components, and complex scene content, it is easy to cause high similarity between the target and the surrounding background, which affects the judgment of the network on the target. Therefore, the designed model is required to distinguish the target and background features well. The most direct idea is to enhance a super-resolution of the image and then estimate the depth, but this will cause high memory consumption, increased computational complexity, and can not meet the end-to-end training and reasoning, significantly expanding the reasoning and training time. Because the attention module can enhance the feature expression ability of the model, at the same time, it can automatically find the significant region and capture the decisive local features; so as to avoid confusing the target and background and finally improve the accuracy of depth estimation, the attention module is introduced in this paper. According to the discussion of SE-Net [16], SE-Net automatically learns the importance of each channel in the network through training and then improves the critical features and suppresses other less essential elements according to the extent of each channel. Compared with SE-Net, CBAM [17] (Convolutional Block Attention Module) calculates the corresponding feature map from two dimensions of space and channel. CBAM is a relatively lightweight module that can be easily integrated into other convolutional neural network models to optimize the features extracted by the network and guide the model to pay more attention to the most discriminative areas in the image to improve the accuracy of the task. Therefore, this paper designs an optimized hybrid attention module to extract critical features. The overall structure is shown in Figure 4.

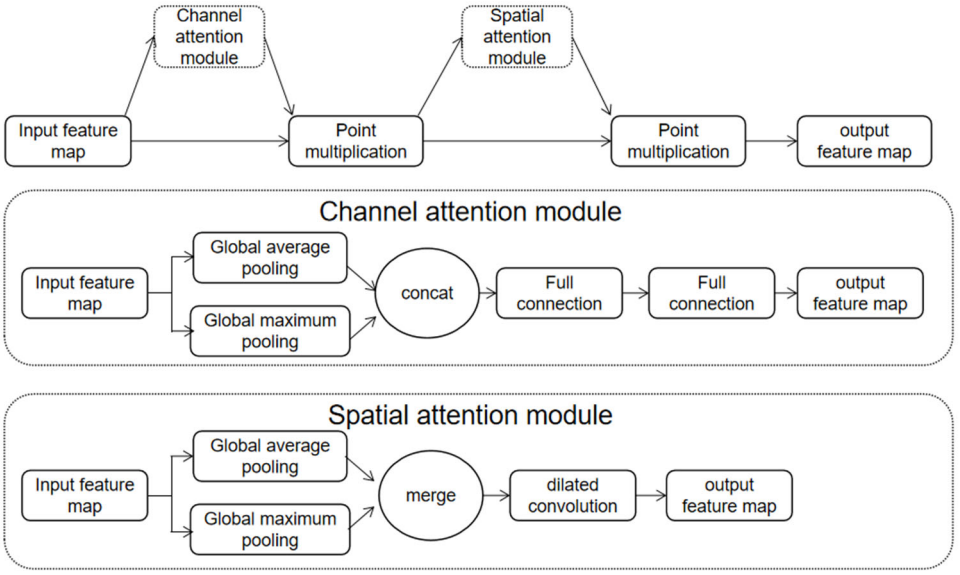

Figure 4. Improved Hybrid Attention Module.

For the characteristic graph $F : F \in R^{C \times H \times W}$ of a given network, C, H, and W are the dimensions of each channel of the feature map, and R is the real number space. The channel attention map with high contribution to the target is extracted by the channel attention module M_C, and the spatial attention map is extracted by the spatial attention module M_S in a cascade way to obtain the final output. In the channel attention module of SE-Net network, the spatial information of the worldwide average pooled statistical characteristic graph is used. Different from their ideas, we think that the global maximum pooling operation can obtain the most various features between targets, which can help to infer more precise channel attention. Therefore, this paper uses global average pooling and all maximum pooling at the same time. Firstly, global average pooling and complete global pooling are used to generate different spatial description features: $M^c_{ave} \in R^{C \times 1 \times 1}$, $M^c_{max} \in R^{C \times 1 \times 1}$. Then, the fused channel description feature M^c_{merge} is obtained by pixel level addition. The fused channel description features are sent to a multi-layer perceptron to get the final channel attention map.

$$M^c_{merge}(F) = M^c_{ave}(F) + M^c_{max}(F) \tag{6}$$

The feature map optimized by the channel attention module is as follows:

$$F' = M^c_{merge}(F) \otimes F \tag{7}$$

Spatial location attention is mainly to find important key information areas in the feature map, which is a supplement to channel attention. As shown in Figure 4, the feature map output by the channel attention module is taken as the input of the spatial attention module, and the average pooling and maximum pooling are used to compress the input feature map. The input feature map F' generates two new features along the channel dimension: $M^s_{ave} \in R^{1 \times H \times W}$, $M^s_{max} \in R^{1 \times H \times W}$ global pooling and average pooling operations. Then the splicing operation is carried out, and the receptive field is extracted by 3 × 3 dilated convolution (division = 2) [18]. Compared with standard convolution, dilated convolution can expand the receptive field of convolution and capture multi-scale information without introducing additional parameters. Therefore, the process of the CBAM module can be expressed as follows:

$$F'' = M^s_{merge}\left[M^c_{merge}(F) \otimes F\right] \otimes \left[M^c_{merge}(F) \otimes F\right] \tag{8}$$

where: F'' is the characteristic graph calculated by the hybrid attention optimization module, "⊗" in Formulas (7) and (8) is the multiplication operation by element.

3.3. Implementation of Self-Supervised Training Based on Stereo Image Pairs

Compared with supervised learning, self-supervised learning does not need labeled training data sets, which significantly reduces the requirements for data sets. The self-supervised training method used in this paper only needs a series of stereo image pairs provided by the binocular camera and the baseline distance and focal length of the camera to train the neural network. Our neural network training method is similar to that of reference [11], its central idea is that stereo image pair is a pair of images obtained by shooting the same scene simultaneously from the left and right viewpoints; in this pair of images, there are geometric constraints of the scene, and the depth information of the scene can be obtained by neural network interpretation. The implementation method is as follows: first, the left-view image is used as input, and the network predicts the right disparity image and the left disparity image. Then, the obtained right disparity map and the left map are fused to get a prediction image for the right image I_r', and the obtained left disparity map and the right map are fused to get a prediction image for the left image I_l'; Finally, the loss is calculated by comparing the two predicted images with the corresponding stereo image pairs, and the loss is minimized by training the network. Through the above ways, the self-supervised training for the network is realized. The process is shown in Figure 5.

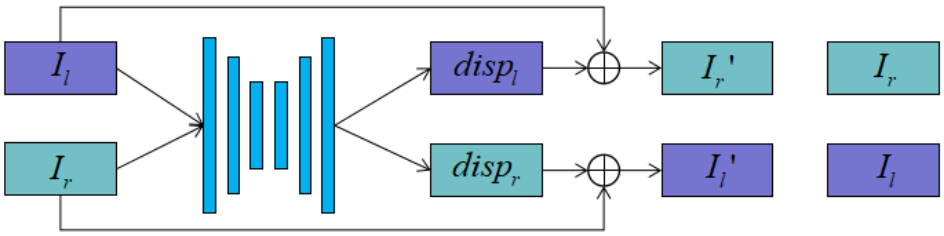

Figure 5. Image depth estimation algorithm training.

The disparity map obtained can be used to calculate the depth estimation of the image by using Formula (9)

$$D = b \bullet f / disp \tag{9}$$

where, D is the depth estimation of the input image of the neural network, b is the reference distance between the left and right cameras, f is the focal length of the camera, $disp$ is the corresponding disparity map.

3.4. Network Structure

We use the encoder and decoder structure and add a jump connection from the encoder module to the corresponding decoder module on the common encoder–decoder. The decoder can obtain higher resolution image information. The specific network structure is shown in Figure 6.

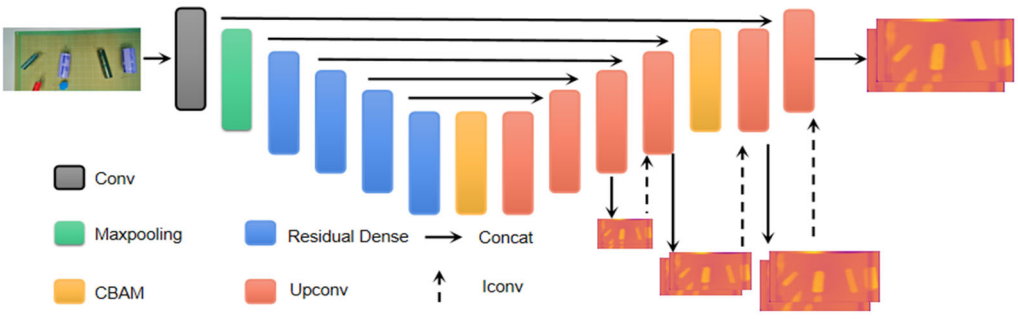

Figure 6. Depth estimation network structure.

The neural network in this paper can be divided into three parts:

(1) Encoder part: This part is divided into the global feature extraction module and the local feature extraction module. The global feature extraction module is composed of a 7 × 7 convolution layer and a top pooling layer, and the local feature extraction module is composed of four sub-modules. The number of residual dense units in each module refers to the number of each module unit in DenseNet-121. The number of residual dense units is 6, 12, 24, and 16, respectively, and the number of output channels of each module is set to 256, 512, 1024, and 1024, respectively. Increase the number of channels of the feature map layer by layer to obtain more dimensional information. The encoder firstly uses the global feature extraction module to extract the whole image information globally, and then further extracts the details of the image through the local feature extraction module to obtain more comprehensive image information.

(2) Jump connection part: In this part, the information obtained by each encoder module is directly introduced to each decoder module through a jump connection so that the decoder module can get higher resolution image information and the information of the previous encoder can be fully utilized; see Figure 6 for the specific connection method.

(3) Decoder part: This part consists of six decoder sub-modules; each sub-module contains an upper convolution layer and a convolution layer, in which the input of the upper convolution layer is the output of the upper layer, and the input of the convolution layer is the combination of the output of the corresponding upper convolution layer and the jump connection. At the same time, the last four decoder sub-modules will output four disparity images through up-sampling. The decoder reduces the dimension of high-level information and restores the resolution of image information through each decoder module, and finally outputs the disparity pyramid composed of four disparity graphs with different resolutions, which are all used for loss calculation. The loss function used is described in detail in Section 3.5. After the training, Disp1 was used to generate the depth map of the input image. The specific parameters of this network model are shown in Table 1.

Table 1. Coding network structure parameters.

Layer	Kernal	Stride	Channel	Input
Conv	7 × 7	2	3/32	Left + Right
Max Pool	3 × 3	2	32/64	Conv
Denseblock_1	[(1 × 1), (3 × 3)] ×6	1	64/64	Max Pool
Trans_1	2 × 2 MaxPool	2	64/256	Denseblock_1
Denseblock_2	[(1 × 1), (3 × 3)] × 12	1	256/256	Trans_1
Trans_2	2×2 MaxPool	2	256/512	Denseblock_2
Denseblock_3	[(1 × 1), (3 × 3)] × 24	1	512/512	Trans_2
Trans_3	2 × 2 MaxPool	2	512/1024	Denseblock_3
Denseblock_4	[(1 × 1), (3 × 3)] × 16	1	1024/1024	Trans_3
Ca_1	7 × 7 MaxPool	1	1024/1024	Denseblock_4
Sa_1	7 × 7 MaxPool	1	1024/1024	Ca_1
Upconv_6	3	2	1024/1024	Sa_1
Iconv_6	3	1	1536/512	Upconv_6 + Denseblock_3
Upconv_5	3	2	512/256	Iconv_6
Iconv_5	3	1	512/256	Upconv_5 + Denseblock_2
Upconv_4	3	2	256/128	Iconv_5
Iconv_4	3	1	192/128	Upconv_4+Denseblock_1
Disp_4	3	1	128/2	Iconv_4
Upconv_3	3	2	128/64	Iconv_4
Iconv_3	3	1	130/64	Upconv_3 + MaxPool + Disp_4
Disp_3	3	1	64/2	Iconv_3
Ca_2	7 × 7 MaxPool	2	64/64	Iconv_3
Sa_2	7 × 7 MaxPool	1	64/64	Ca_2
Upconv_2	3	1	64/32	Sa_2
Iconv_2	3	2	98/32	Upconv_3 + Conv + Disp_3
Disp_2	3	1	32/2	Iconv_2
Upconv_1	3	1	32/16	Iconv_2
Iconv_1	3	2	18/16	Upconv_1 + Disp_2
Disp_1	3	1	16/2	Iconv_1

In the above table, the kernel is the convolution kernel width, the stride is the convolution step size, the channel is the number of input and output channels, Ca represents channel attention, Sa represents spatial attention, and Denseblock represents representation dense residual module. The "+" in the table represents the concatenation of the input on the channel dimension. Conv stands for convolution, Upconv stands for deconvolution, Iconv stands for superposition of deconvolution result and convolution of the coding network in channel dimension and then convolution. Disp stands for left and right disparity map to be output. Due to different scales, a disparity map with the original width of 1, 1/2, 1/4, 1/8 is generated.

3.5. Loss Function Design

We use binocular stereo image pairs to train the network. The final output of the dense residual network is four disparity estimation maps with different scales. Therefore, when calculating the loss, we separately calculate the four scales and then sum them up. In other words, the total loss of the network can be divided into four parts $C = \sum_{s=1}^{4} C_s$. Where C_s is the loss corresponding to each disparity estimation map. Each C_s is composed of three parts, C_{ap} represents the similarity between the input image and the reconstructed image; C_{ds} represents the smoothness of the disparity map; C_{lr} represents the similarity between the left and right disparity maps of the network output. By weighted summation of the three kinds of losses, the loss value of each scale can be obtained, and its specific expression is shown in Formula (10).

$$C_s = \omega_1 C_{ap} + \omega_2 C_{ds} + \omega_3 C_{lr} \tag{10}$$

$\omega_1, \omega_2, \omega_3$ represents the weight coefficients of the weight of three kinds of losses in the total loss. The specific set of these three coefficients in this chapter will be consistent with [11] $\omega_1 = 1,\ \omega_2 = 0.1,\ \omega_3 = 1$. In the output of each scale, there are two disparity estimation maps: left disparity estimation map and right disparity estimation map. Therefore, each kind of loss is also divided into two parts: left disparity loss and right disparity loss, which can be expressed by Formulas (11)–(13).

$$C_{ap} = C_{ap}^{l} + C_{ap}^{r} \tag{11}$$

$$C_{ds} = C_{ds}^{l} + C_{ds}^{r} \tag{12}$$

$$C_{lr} = C_{lr}^{l} + C_{lr}^{r} \tag{13}$$

l represents left disparity loss and r represents right disparity loss. The specific calculation methods of three kinds of losses will be given below.

Image reconstruction loss: the loss reflects the difference between the two-view image input by dense residual network and the corresponding view reconstruction image. The specific calculation formula of the loss is shown in Formulas (14) and (15).

$$C_{ap}^{l} = \frac{1}{N}\sum_{i,j} \alpha \frac{1 - SSIM\left(I_{i,j}^{l}, \widetilde{I}_{i,j}^{l}\right)}{2} + (1-\alpha)\left\|I_{i,j}^{l} - \widetilde{I}_{i,j}^{l}\right\| \tag{14}$$

$$C_{ap}^{r} = \frac{1}{N}\sum_{i,j} \alpha \frac{1 - SSIM\left(I_{i,j}^{r}, \widetilde{I}_{i,j}^{r}\right)}{2} + (1-\alpha)\left\|I_{i,j}^{r} - \widetilde{I}_{i,j}^{r}\right\| \tag{15}$$

where N is the total number of pixels, i, j represents the abscissa and ordinate values of the image, respectively, and α is the weight coefficient of the two-loss terms. The weight value set in this paper is the same as that in reference [11], which is 0.85. I is the input image. In the formula, the former term reflects the similarity between the original view image and the reconstructed image of the corresponding view. In contrast, the latter term represents the difference between the original view image and the reconstructed image of the corresponding view.

Disparity smoothing loss: the loss is used to local smooth the depth discontinuous points in the disparity estimation map. The depth of the discontinuous points in the disparity estimation map will produce drastic changes in the gradient domain [19]. Therefore, by increasing the smoothing loss in the gradient domain of the disparity map, the depth discontinuity area in the disparity estimation map can be smoother and a more realistic disparity estimation map can be obtained. The specific calculation method of the loss is shown in Formulas (16) and (17).

$$C_{ds}^{l} = \frac{1}{N}\sum_{i,j}\left(\left|\partial_x d_{i,j}^{l}\right| e^{-\|\partial_x d_{i,j}^{l}\|} + \left|\partial_y d_{i,j}^{l}\right| e^{-\|\partial_y d_{i,j}^{l}\|}\right) \tag{16}$$

$$C_{ds}^{r} = \frac{1}{N}\sum_{i,j}\left(\left|\partial_x d_{i,j}^{r}\right| e^{-\|\partial_x d_{i,j}^{r}\|} + \left|\partial_y d_{i,j}^{r}\right| e^{-\|\partial_y d_{i,j}^{r}\|}\right) \tag{17}$$

where ∂_x and ∂_y are the gradients of the disparity map on the x and y axes, and $d_{i,j}^{l}$ is the value of the left disparity map at coordinates (i, j). Other parameters are consistent with the meaning in Formula (14).

Disparity consistency loss: this loss reflects the left and right consistency between disparity graphs. The second estimation image of the disparity estimation image can be obtained by left–right fusion and right–left fusion, respectively. The disparity consistency loss can be obtained by comparing the first estimate image with the second estimate image. The specific calculation method is shown in Formulas (18) and (19).

$$C_{lr}^{l} = \frac{1}{N}\sum_{i,j}\left|d_{i,j}^{l} - d_{i,j+d_{i,j}^{l}}^{r}\right| \tag{18}$$

$$C_{lr}^{r} = \frac{1}{N}\sum_{i,j}\left|d_{i,j}^{r} - d_{i,j+d_{i,j}^{r}}^{l}\right| \tag{19}$$

3.6. Evaluation Metrics

The evaluation criteria are mainly divided into error rate and accuracy rate. The error rate includes average relative error ***Abs Rel***, square root relative error ***Sq Rel***, linear root mean square error ***RMSE***, and logarithmic root mean square error ***RMSE log***. The accuracy rate includes three threshold indicators, which are 1.25, 1.25^2 and 1.25^3 respectively. The lower the error, the better the accuracy. The specific calculation formula is as follows [3]:

(1) Average relative error ***Abs Rel***: The ratio of the absolute value of the difference between the estimated depth value and the true depth value to the true depth value. The specific calculation method is shown in the formula.

$$\textbf{\textit{Abs Rel}} = \frac{1}{T}\sum_{i=1}^{T}\frac{\|d_i^* - d_i\|}{d_i} \tag{20}$$

Among them T is the total number of pixels, d_i is the estimated depth value of pixels, and d_i^* is the real depth value of corresponding pixels. The same letters in the following formulas have the same meaning.

(2) Square root-relative error ***Sq Rel***: The ratio of the square of the difference between the estimated depth value and the true depth value to the true depth value. The specific calculation method is shown in the formula.

$$\textbf{\textit{Sq Rel}} = \frac{1}{T}\sum_{i=1}^{T}\frac{\|d_i^* - d_i\|^2}{d_i} \tag{21}$$

(3) Linear root mean square error ***RMSE***: Used to describe the root mean square of the difference between the estimated and true depths.

$$\textbf{\textit{RMSE}} = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(d_i^* - d_i)^2} \tag{22}$$

(4) Log root mean square error ***RMSE log***: It is the expression of root mean square error in the log field, which is used to describe the root mean square of the difference between the logarithm of the estimated depth value and the logarithm of the true depth value.

$$RMSE\ log = \sqrt{\frac{1}{T}\sum_{i=1}^{T}(logd_i^* - logd_i)^2} \tag{23}$$

(5) Accuracy: Used to describe the percentage of the ratio of the estimated depth to the true depth within a fixed threshold.

$$\delta = max\left(\frac{d_i^*}{d_i}, \frac{d_i}{d_i^*}\right) < thr, \quad thr = 1.25, 1.25^2, 1.25^3 \tag{24}$$

Among them, thr represents the given threshold.

4. Experiments and Discussion

In this part, the experiment of this paper will be explained in detail and compared with several representative image depth estimation methods on the KITTI data set, Cityscapes data set, and PCB data set we collected, including the supervised learning method of Eigen et al. [3] and the unsupervised learning method of Godard et al. [11]. The results verify the effectiveness of this method in error, accuracy, and visual depth effect feasibility.

4.1. Datasets

PCB data set: We use a 2.5 megapixel binocular camera to make the PCB data set, the baseline distance of the camera is 2.5 cm and the focal length is 4.2 mm. The data set includes nine categories: square capacitance, varistor, medium capacitance, triode, I-shaped inductance, relay, resistance, large capacitance, and circuit board. The training set contains 9000 pairs of images, the verification set contains 1000 pairs of images, the test set contains 1000 pairs of images, and the original image resolution is 1700 × 830; as shown in Figure 7, the first row is the left image and the second row is the right image.

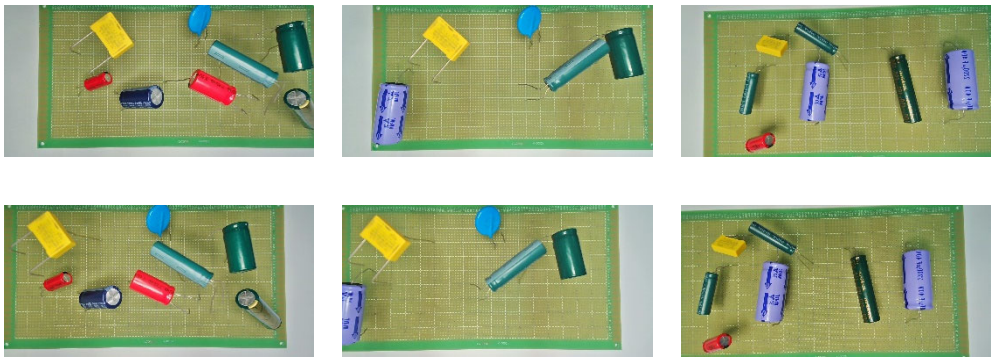

Figure 7. Samples of PCB data set.

Eigen Split set: In this paper, the algorithm is trained and tested on the KITTI 2015 data set, using the data set segmentation method proposed by Eigen et al. [3]. The data set contains 42,382 pairs of modified images from 61 scenes, and the original image resolution is 1242 × 375. The image with a resolution of 512 × 256 is obtained from the original image processing. Eigen et al. selected 697 pictures from 29 scenes in the original KITTI 2015 data set as the test set. Among the 23,488 images in the remaining 32 scenes, 22,600 images are selected for training and the rest for evaluation. To compare with most of the methods using this data set, we use the clipping method proposed by Eigen et al. [3] to test this data set and evaluate the resolution of the input image.

The Cityscapes data set is the image data set collected by the Benz group company when driving driverless. The Cityscapes data set is collected in 50 cities with different

scenes and different seasons. It is divided into schools, streets, towns, work areas, etc., and provides 5000 fine label images and 20,000 rough labels images, and 30 kinds of labeled objects.

4.2. Implementation Details

The proposed model is implemented with the PyTorch framework and trained on two NVIDIA GTX 2080Ti GPUs. In the model training, after epochs reach 45, loss starts to maintain dynamic balance. After weighing, this paper finally sets it to 50, and the number of batches is 16. The initial learning rate was set to 0.0001, and the learning rate was maintained in the first 30 epochs, while the learning rate was halved in 30–40 epochs and then halved in 40–50 epochs. Adam optimizer is used for optimization, and the optimizer parameter is set to $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$. The initial learning rate was 0.0001, using the same data enhancement and post-processing techniques as in reference [11]. The data are randomly enhanced during the training process to increase the diversity of the data, such as brightness, contrast, saturation adjustment, and horizontal flip.

4.3. Ablation Study

The ablation study was also taken in the experiment, and we choose Monodepth [11] as the baseline method. To verify the model's generalization ability, we tested it with the Eigen Split set, the Cityscapes data set, and the PCB data set without any adjustment to the model parameters. Monodepth-D upgrades four residual modules in Monodepth to four dense residual modules; Monodepth-C adds optimized hybrid attention modules to Monodepth in the encoding and decoding stages, respectively; Monodepth-DC adds optimized hybrid attention modules to Monodepth-D in the encoding and decoding stages, respectively; the experimental results are shown in Table 2. At the same time, this paper visualizes some data for a more intuitive comparison; the first column in Figure 8 is the predicted results of the four algorithms on the Eigen Split set; the second column is the expected results of the four algorithms on the Cityscapes data set; the third column is the predicted results of the four algorithms on the PCB data set.

Our depth estimation result shows clearer edges on some thin structures, such as trunks shown in Figure 8. We also found that, due to the deep network, Monodepth-D provides a clearer depth map than Monodepth in a variety of environments and can accurately predict distant targets. Monodepth-C has sharper edges than Monodepth predicted depth maps because the hybrid attention mechanism provides better feature fusion and the network focuses more on the local information of the image. In the areas with more complex details, the target edge predicted by Monodepth-DC method is the clearest, such as the telegraph pole, the human's head, and the reflection part of electronic components in the second column of images. Of the four networks, each index of the depth map predicted by Monodepth-D and Monodepth-C is better than Monodepth; Monodepth-DC works best, both in numerical metrics and visual depth map results. It can be found from Table 2 that the ***RMSE*** of this algorithm is 4.984, which is 15.9% lower than that of Monodepth. The threshold accuracy of $\delta \leq 1.25$ is 0.858, which is 6.8% higher than Monodepth and 3.2% higher than Monodepth-D. It can be seen that the depth error of the image predicted by this algorithm is smaller and the accuracy is higher.

Table 2. Ablation study on the Eigen Split set.

Method	Lower Is Better				Higher Is Better		
	Abs Rel	*Sq Rel*	*RMSE*	*RMSE log*	$\delta \leq 1.25$	$\delta \leq 1.25^2$	$\delta \leq 1.25^3$
Monodepth	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Monodepth-D	0.123	1.029	5.392	0.236	0.831	0.953	0.970
Monodepth-C	0.135	1.312	5.671	0.242	0.829	0.935	0.966
Monodepth-DC	0.119	0.987	4.984	0.225	0.858	0.962	0.974

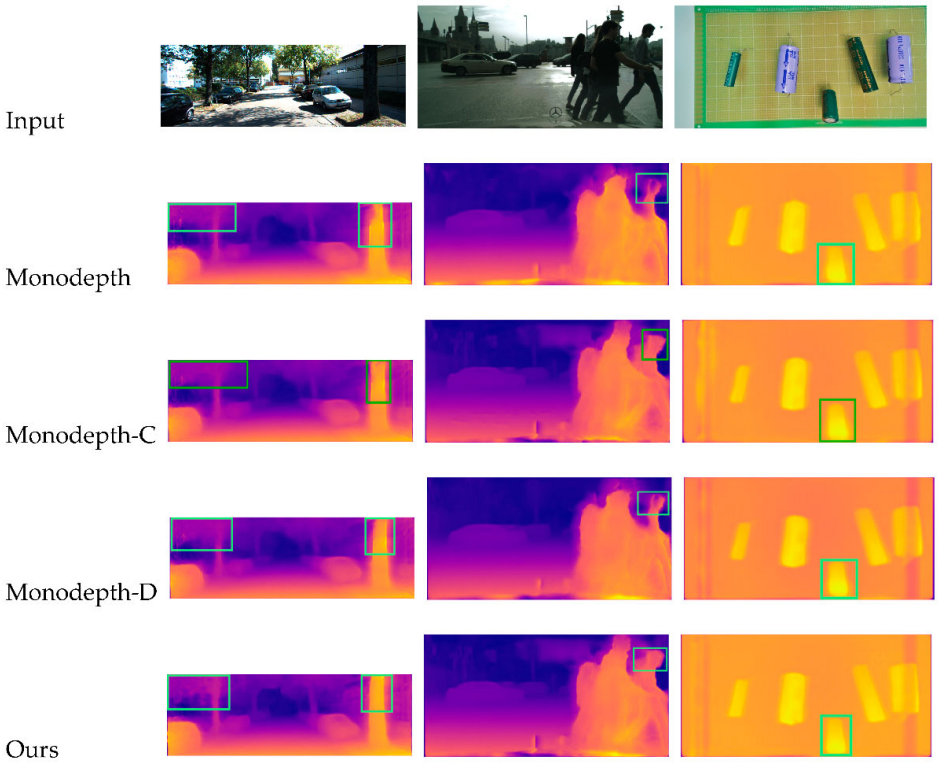

Figure 8. Results of three algorithms on the Eigen split set, Cityscapes data set, and PCB data set.

4.4. Performance Comparison with State-of-the-Art

We verify the effectiveness of the model proposed in this paper on the Eigen Split set; the comparison results with other existing methods are shown in in Table 3. The depth estimation results of several representative images in the Eigen Split set, the Cityscapes data set, and the PCB data set are shown in Figures 9–11.

Table 3. Results on the Eigen Split set.

Method	Super-Vision	*Abs Rel*	*Sq Rel*	*RMSE*	*RMSElog* (log)	$\delta \leq 1.25$	$\delta \leq 1.25^2$	$\delta \leq 1.25^3$
Eigen [3]	Yes	0.203	1.548	6.307	0.282	0.702	0.890	0.890
Liu [4]	Yes	0.201	1.584	6.471	0.273	0.680	0.898	0.967
Zhou [12]	No	0.208	1.768	6.856	0.283	0.678	0.885	0.957
DF-Net [20]	No	0.150	1.124	5.507	0.223	0.806	0.933	0.973
Ranjan [21]	No	0.148	1.149	5.464	0.226	0.815	0.935	0.973
Garg [10]	No	0.169	1.080	5.104	0.273	0.740	0.904	0.962
Godard [11]	No	0.148	1.344	5.927	0.247	0.803	0.922	0.964
Xu [22]	No	0.132	0.911	/	0.162	0.804	0.945	0.981
GASDA [23]	No	0.149	1.003	4.995	0.227	0.824	0.941	0.973
Ours	No	0.119	0.987	4.984	0.225	0.858	0.962	0.974

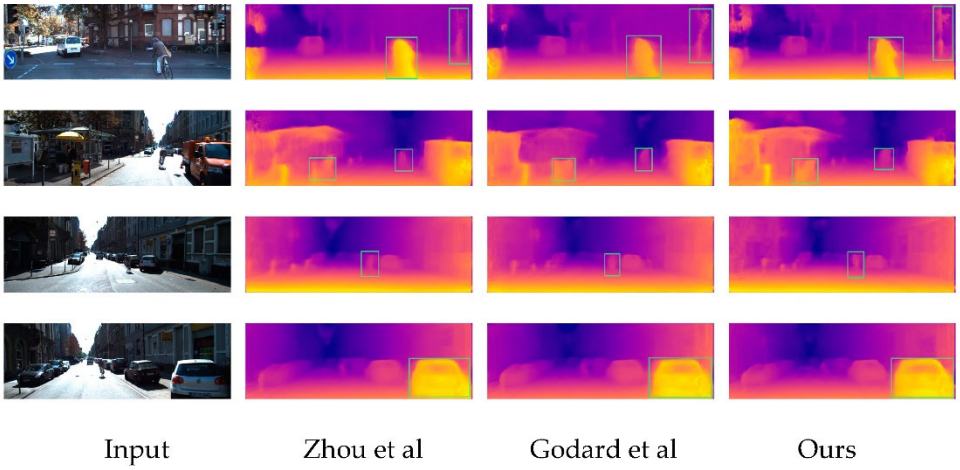

Figure 9. Results compared with other methods on the Eigen split set.

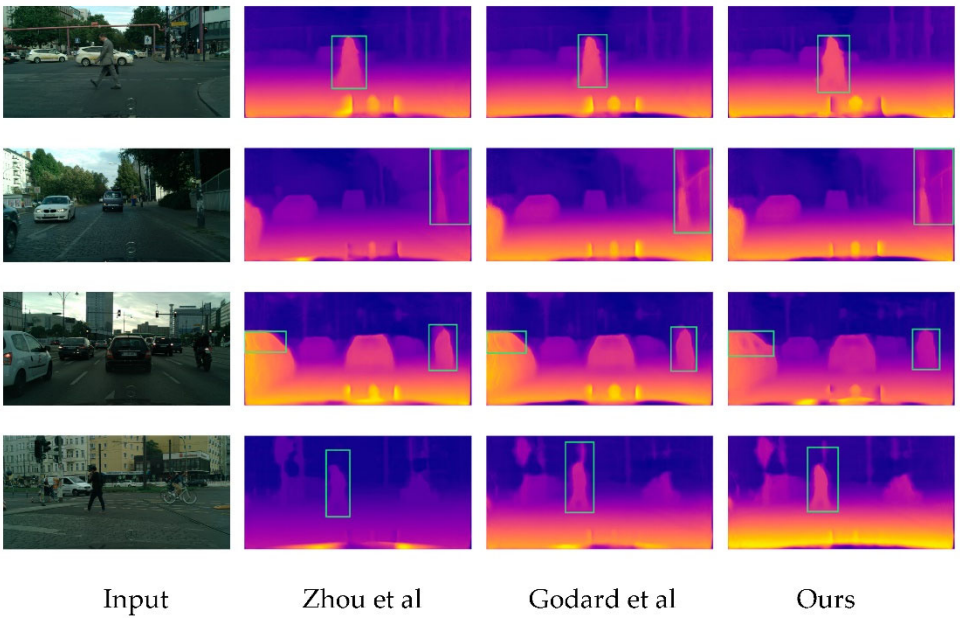

Figure 10. Results compared with other methods on the Cityscapes data set.

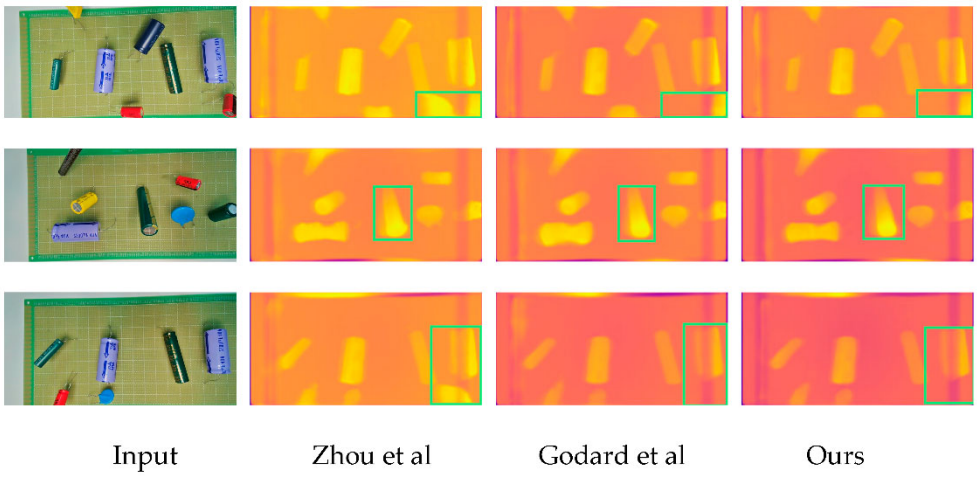

Figure 11. Results compared with other methods on the PCB data set.

(1) Quantitative analysis

As shown in Table 3, compared with the method proposed by Eigen et al. [3], the Abs Rel of our method is lower than their 0.4%, and the RMS log is lower than their 5.5% on the Eigen Split set. Although our method does not use the ground truth depth map as supervision, it considers the similarity between pixels so as to make the error smaller; when $\delta \leq 1.25$, the threshold accuracy is 15.6% higher than them, and 8.5% higher than them when $\delta \leq 1.253$; the higher the accuracy, the more depth image information will be estimated.

Compared with the unsupervised learning method proposed by Zhou et al. [12], the Abs Rel of our method is 0.3% lower than them, the threshold accuracy of $\delta \leq 1.25$ is 18% higher than them, and the threshold accuracy $\delta \leq 1.25^3$ is also 1.7% higher than them. We use dense connections to enhance feature reuse and feature forward propagation, so we can effectively improve the accuracy of image depth prediction. The spatial context in the image is used more effectively through the hybrid attention mechanism to enhance the feature extraction ability of the model. In the process of image depth estimation, we can get more position information and make the boundary of the object more obvious. In general, our model has small error and high prediction accuracy.

Because the distance between the electronic components and the camera is far, it is difficult to directly measure the distance between the electronic components and the camera. Therefore, we judge the depth accuracy in the Z direction by measuring the maximum distance between the electronic components and the desktop. The depth estimation experimental results of five electronic components are shown in Table 4, and the numbers in parentheses are errors. From the data in the table, the average error value of five electronic components is about 0.35 mm, the minimum error value is 0.28 mm, and the maximum error value is 0.44 mm by the Monodepth-DC method. Through the Monodepth method, the average error value of five electronic components is about 0.60 mm, the minimum error value is 0.49 mm, and the maximum error value is 0.66 mm. Compared with the benchmark algorithm, our accuracy index has been greatly improved.

Table 4. Results on the PCB data set.

Method	Square Capacitor (mm)	Medium Capacitor (mm)	Large Capacitor (mm)	Medium Varistor (mm)	Large Varistor (mm)
Ground Truth	12.28	20.61	32.17	15.88	21.66
Eigen [3]	13.81 (1.53)	22.17 (1.56)	33.52 (1.35)	17.63 (1.75)	22.57 (0.91)
Zhou [12]	14.05 (1.77)	22.65 (2.04)	34.31 (2.14)	17.92 (2.04)	22.81 (1.15)
Monodepth	12.93 (0.65)	21.18 (0.57)	32.66 (0.49)	16.52 (0.64)	22.32 (0.66)
Monodepth-DC	12.72 (0.44)	21.02 (0.41)	32.53 (0.36)	16.18 (0.30)	21.94 (0.28)

(2) Qualitative analysis

As can be seen from Figure 9, the depth map predicted by us is more perfect in detail than that predicted by the method of Godard et al. [11] and Zhou et al. [12], especially in areas with highly similar colors. In the third line of comparison, due to the influence of shadow, the method of Zhou et al. [12] confuses pedestrian and vehicle shadow when predicting the depth value of pedestrian in the middle of the image. However, our method is not affected by shadow and also performs a better prediction for distant vehicles. In the comparison of the first line, the depth map predicted by our method can not only well separate the trees, street lights, and background in the foreground, but also completely reflect the depth changes of the road from near to far; while the depth map predicted by Zhou et al. [12] has a fuzzy separation effect on the front and back scenes. In Figure 10, due to the poor contrast of the scene without strong light, the separation effect of the front and back scenes of the depth map predicted by Zhou et al. [12] is more fuzzy, and the target contour edges in the depth map predicted by our method are more accurate and the

prediction effect of details is better. Meanwhile, the prediction was better in challenging areas such as pedestrian heads, moving vehicles, tree trunks, signs, and traffic lights.

In Figure 11, the depth map of electronic components predicted by Zhou et al. [12] and Godard et al. [11] has serious fuzziness, and in the depth estimation map of Zhou et al. [12], some backgrounds are misjudged as prospects. The boundary of electronic components in the depth map indicated by us is clear, and there is almost no place for the wrong estimation. Due to the idea of DenseNet, our method can alleviate the problem of gradient disappearance and strengthen feature propagation. Therefore, the depth map has less error estimation and is relatively clear on the whole. Meanwhile, the hybrid attention mechanism is improved, and the object boundary ambiguity caused by Zhou et al. [12] in the process of image estimation is solved.

5. Conclusions

In this paper, the improved Monodepth model for depth estimation of monocular electronic components was proposed. In order to improve the accuracy of image depth estimation, we improved the network by introducing dense residual module into the coding and decoding structure. In order to solve the problem of boundary ambiguity in monocular image depth estimation, we added an improved hybrid attention module to the coding and decoding structure, respectively. We collected 11,000 pairs of images containing nine electronic components (square capacitance, varistor, medium capacitance, triode, I-shaped inductance, relay, resistance, large capacitance, and circuit board.), Data augmentation was accomplished by adding noise (Gaussian blur, gaussian noise, motion blur, and salt and pepper noise) to the original set of images.

In order to prove the effectiveness of our proposed method, it was compared with some of the latest monocular image depth estimation methods. The experimental results showed that our model has similar accuracy in depth estimation compared with Eigen's supervised model. In addition, compared with Zhou and other unsupervised models, it has significant advantages in the prediction of object edge contour information.

The monocular image depth estimation method proposed in this paper can be used for three-dimensional reconstruction of electronic components, but there is still a certain gap between its performance and real-time detection. In the future, we will focus on optimizing the existing models so that the depth of electronic components in video can be estimated to meet the real-time requirements.

Author Contributions: Conceptualization, Z.H.; data curation, Z.H. and J.L.; formal analysis, Z.H. and J.L.; funding acquisition, J.G. and Z.H.; investigation, Z.H. and J.L.; methodology, Z.H. and J.H.; resources, Z.H. and J.L.; software, Z.H., J.L. and S.L.; supervision, J.G.; writing—original draft, Z.H. and J.L.; writing—review and editing, Z.H., J.L. and J.G. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, grant number 51875266.

Data Availability Statement: Requested data will be available by contacting the first author or corresponding author, if there are reasonable grounds or that the confidentiality of some data is not violated.

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

References

1. Avinash, C.T. Towards a second green revolution. *Irrig. Drain.* **2016**, *65*, 388–389.
2. Butollo, F.; Luthje, B. Made in China 2025: Intelligent manufacturing and work. In *The New Digital Workplace How New Technologies Revolutionise Work*; Macmilan: London, UK, 2017; Volume 20, pp. 42–61.
3. Eigen, D.; Puhrsch, C.; Fergus, R. Depth map prediction from a single image using a multi-scale deep network. *Adv. Neural Inf. Processing Syst. (NIPS)* **2014**, *27*, 2366–2374.
4. Liu, F.; Shen, C.; Lin, G. Deep convolutional neural fields for depth estimation from a single image. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, MA, USA, 19–24 June 2015; pp. 5162–5170.

5. Liu, F.; Shen, C.; Lin, G.; Reid, I. Learning depth from single monocular images using deep convolutional neural fields. *IEEE Trans. Pattern Anal. Mach. Intell.* **2016**, *38*, 2024–2039. [CrossRef] [PubMed]
6. Laina, I.; Rupprecht, C.; Belagiannis, V. Deeper depth prediction with fully convolutional residual networks. In Proceedings of the Fourth International Conference on 3D vision (3DV), Stanford, CA, USA, 25–28 October 2016; pp. 239–248.
7. Li, J.; Klein, R.; Yao, A. A two-streamed network for estimating fine-scaled depth maps from single RGB images. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 3392–3400.
8. Kendall, A.; Martirosyan, H.; Dasgupta, S. End-to-end learning of geometry and context for deep stereo regression. In Proceedings of the International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017.
9. Fu, H.; Gong, M.; Wang, C. Deep ordinal regression network for monocular depth estimation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2002–2011.
10. Garg, R.; Bg, V.K.; Carneiro, G. Unsupervised CNN for single view depth estimation: Geometry to the rescue. In Proceedings of the European Conference on Computer Vision (ECCV), Amsterdam, The Netherlands, 23–28 August 2016; pp. 740–756.
11. Clément, G.; Mac, A.O.; Brostow, G.J. Unsupervised monocular depth estimation with left-right consistency. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 6602–6611.
12. Zhou, T.; Brown, M.; Snavely, N. Unsupervised learning of depth and ego-motion from video. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 27–30 June 2017; pp. 6612–6619.
13. Zhan, H.; Garg, R.; Weerasekera, C.S. Unsupervised learning of monocular depth estimation and visual odometry with deep feature reconstruction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 340–349.
14. Yin, Z.; Shi, J. Geonet: Unsupervised learning of dense depth, optical flow and camera pose. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 1983–1992.
15. Clevert, D.-A.; Unterthiner, T.; Hochreiter, S. Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs). In Proceedings of the 4th International Conference on Learning Representations, ICLR 2016, San Juan, PR, USA, 2–4 May 2016.
16. Hu, J.; Shen, L.; Albanie, S. Squeeze-and-excitation networks. In Proceedings of the IEEE Transactions on Pattern Analysis and Machine Intelligence, Salt Lake, UT, USA, 18–23 June 2018; pp. 7132–7141.
17. Woo, S.; Park, J.; Lee, J.Y. *CBAM: Convolutional Block Attention Module*; Springer: Cham, Switzerland, 2018.
18. Quan, Y.; Li, Z.; Zhang, C. Object detection by combining deep dilated convolutions network and light-weight network. In Proceedings of the International Conference on Knowledge Science, Engineering and Management, Singapore, 6–8 August 2022; Springer: Cham, Switzerland, 2019; pp. 452–463.
19. Heise, P.; Klose, S.; Jensen, B. PatchMatch with Huber Regularization for Stereo Matching. In Proceedings of the 2013 IEEE International Conference on Computer Vision, Sydney, Australia, 1–8 December 2013.
20. Zou, Y.; Luo, Z.; Huang, J.B. *DF-Net: Unsupervised Joint Learning of Depth and Flow Using Cross-Task Consistency*; Springer: Cham, Switzerland, 2018.
21. Ranjan, A.; Jampani, V.; Balles, L. Competitive collaboration: Joint unsupervised learning of depth, camera motion, optical flow and motion segmentation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.
22. Ricci, E.; Ouyang, W.; Wang, X.; Sebe, N. Monocular depth estimation using multi-scale continuous CRFs as sequential deep networks. *IEEE Trans. Pattern Anal. Mach. Intell.* **2019**, *41*, 426–1440.
23. Zhao, S.; Fu, H.; Gong, M.; Tao, D. Geometry-aware symmetric domain adaptation for monocular depth estimation. In Proceedings of the 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 15–20 June 2019.