symmetry

MDPI

Article

A Semi-Supervised Semantic Segmentation Method for Blast-Hole Detection

Zeyu Zhang [1], Honggui Deng [1,*], Yang Liu [1], Qiguo Xu [1] and Gang Liu [1,2]

[1] School of Physics and Electronics, Central South University, Lushan South Road, Changsha 410083, China; 192211038@csu.edu.cn (Z.Z.); 192211037@csu.edu.cn (Y.L.); 202211045@csu.edu.cn (Q.X.); 162201003@csu.edu.cn (G.L.)

[2] College of Information Science and Engineering, Changsha Normal University, Teli Road, Changsha 410100, China

* Correspondence: denghonggui@csu.edu.cn

Abstract: The goal of blast-hole detection is to help place charge explosives into blast-holes. This process is full of challenges, because it requires the ability to extract sample features in complex environments, and to detect a wide variety of blast-holes. Detection techniques based on deep learning with RGB-D semantic segmentation have emerged in recent years of research and achieved good results. However, implementing semantic segmentation based on deep learning usually requires a large amount of labeled data, which creates a large burden on the production of the dataset. To address the dilemma that there is very little training data available for explosive charging equipment to detect blast-holes, this paper extends the core idea of semi-supervised learning to RGB-D semantic segmentation, and devises an ERF-AC-PSPNet model based on a symmetric encoder–decoder structure. The model adds a residual connection layer and a dilated convolution layer for down-sampling, followed by an attention complementary module to acquire the feature maps, and uses a pyramid scene parsing network to achieve hole segmentation during decoding. A new semi-supervised learning method, based on pseudo-labeling and self-training, is proposed, to train the model for intelligent detection of blast-holes. The designed pseudo-labeling is based on the HOG algorithm and depth data, and proved to have good results in experiments. To verify the validity of the method, we carried out experiments on the images of blast-holes collected at a mine site. Compared to the previous segmentation methods, our method is less dependent on the labeled data and achieved IoU of 0.810, 0.867, 0.923, and 0.945, at labeling ratios of 1/8, 1/4, 1/2, and 1.

Keywords: hole detection; semi-supervised learning; semantic segmentation; RGB-D perception

Citation: Zhang, Z.; Deng, H.; Liu, Y.; Xu, Q.; Liu, G. A Semi-Supervised Semantic Segmentation Method for Blast-Hole Detection. *Symmetry* **2022**, *14*, 653. https://doi.org/10.3390/sym14040653

Academic Editors: João Ruivo Paulo, Cristina P. Santos, Gabriel Pires and Davide Pagano

Received: 25 February 2022
Accepted: 22 March 2022
Published: 23 March 2022

1. Introduction

Charging explosives is an essential part of mining operations [1], and conveying the pipe to the required blast-hole location is a critical link in the development of a mechanized charging system [2]; i.e., hole detection. The main objectives of this work were focused on assisting with blast-hole detection for charge blasting systems. At present, almost all mines still require a large amount of manpower during hole loading, due to manual operation, and a series of problems inevitably occur, such as inefficiency, explosive corrosion, untimely response, and physiological fatigue [3]. More seriously, the workers are exposed under the top of the loose rock, and their lives are in danger all the time. In recent years, with the development of artificial intelligence and computer vision technology, automatic hole detection has undoubtedly become a major trend for the future [4]. Unfortunately, the harsh environment in mines makes this hard to achieve. There are various methods of drilling blast-holes, and the fracture patterns of rock walls near the holes are varied. These problems seriously affect the efficiency and accuracy of automatic hole detection, which makes it a key problem that needs to be solved for explosive charging equipment. Some related research work has been carried out, as follows.

For the hole detection problem, Duda et al. [5] first used the Hough transform (HT) for circle detection. Nakanishi et al. [6] proposed a highly parallel Hough transform algorithm for high-speed extraction, using the symmetry of image space, but the algorithm is very intensive in terms of computational and storage resources. Many improvements have, therefore, been made on this basis, such as probabilistic HT [7], randomized HT [8], fuzzy HT [9], and randomized circle detection (RCD) algorithms [10], etc. These methods make the detection more efficient, but none of them has a good trade-off between accuracy and computational complexity. In the work done by Ayala-Ramirez et al. [11], a genetic algorithm (GA) detector achieved sub-pixel accuracy for round hole detection, but could not handle small round holes in images. The edge drawing parameter free (EDPF) algorithm proposed by Akinlar et al. [12] uses a parameter-free edge segment detector to achieve real-time detection of circular holes, based on previous studies; hence the name EDCircles, but it is generally ineffective for detecting defective circular holes. Inspired by Jiao et al. [13], object detection based on deep learning came into the limelight.

With the development of deep learning in computer vision, there is a clear picture of its achievements in the field of object detection. One of the generic models is the full convolutional neural network (FCN), which can be trained end-to-end [14]. Chen et al., proposed [15] introducing conditional random fields at the fully connected layer, based on FCN, and showed a better segmentation performance, as tested in PASCAL VOC 2012, but with the problems of a large number of parameters and long processing time. Lu et al. [16] raised a graph model initialized by FCN, named Graph-FCN, to achieve better segmentation performance by extracting more features with a larger receptive field; but this still does not address the problem of low processing efficiency. An efficient residual factorized convolutional neural network (ERFNet) [17] reduces the number of network parameters and processing time, by adding a residual connection layer and a dilated convolution layer in the down-sampling based on FCN, and at the same time obtaining more information, but the limited scene resolution capability restricts the segmentation effect of the network. Zhao et al. [18] suggested a pyramid scene parsing network (PSPNet), to improve the network's ability to perceive scenes; however, in practical tests, it was found that the quantity of information provided by RGB images in the mining cave environment was not sufficient for detection. Therefore, the method using RGB-D data as network input [19] captured our attention. References [20,21] achieved good segmentation with RGB-D data, but the feature extraction ratio was stationary, which is not suitable for some special cases. The study in [22] enhanced the segmentation effect, by achieving a dynamic extraction of features through an attention complementary module (ACM). However, training these networks takes a large amount of labeled data. In the mining industry, there is very little labeled data, as manual labeling is time-consuming and tedious. Large amounts of unlabeled data are usually accessible with relative ease. The semi-supervised learning (SSL) developed by Chapelle et al. [23] provides us with methods to train deep networks using vast quantities of unlabeled data.

In summary, blast-hole detection approaches suffer from two problems: the insufficient accuracy of segmentation models, and the absence of labeled data for training. We intend to train a multimodal fusion deep learning model to achieve pixel-level detection of blast-holes through a semi-supervised learning method. To accomplish this, we need to design a feature extraction network with RGB-D input and employ it to increase the information in the feature map. In addition, a new SSL method is proposed, by combining pseudo-labeling and self-training methods to train the network using unlabeled data, and without increasing the model complexity.

In this paper, we propose a new method for achieving the intelligent detection of blast-holes, using semi-supervised learning. We design a symmetric FCN-based ERF-AC-PSPNet model to enhance the sample extraction capability, which incorporates a residual connection layer and dilated convolution layer in down-sampling. The ACM is integrated to capture the feature map for dynamic extraction from RGB-D data, and the pyramid scene parsing network is used in decoding. After that, the model is trained with our SSL

approach, to achieve hole segmentation. The pseudo-label we designed is based on the HOG algorithm and depth and provided a good result in experiments. Compared with the currently popular segmentation methods, our approach has a prominent segmentation effect on blast-holes and less dependence on labeled data; reaching IoU of 0.810, 0.867, 0.923, 0.945, at labeling ratios of 1/8, 1/4, 1/2, and 1. The main features of our work are as follows:

1. The proposed method for blast-hole detection with our ERF-AC-PSPNet model extends the idea of SSL to RGB-D semantic segmentation and has a segmentation effect, to meet the demand for blast-hole detection without labeled datasets.
2. The model optimizes the problem of unequal information and inconsistent background distribution between RGB images and depth images.

2. Related Work

In this section, we review the relevant literature on RGBD pixel-wise semantic segmentation and semi-supervised learning.

2.1. RGBD Pixel-Wise Segmentation

The performance of semantic segmentation networks based on single-modal data tends to degrade when illumination conditions are not sufficient in the mine tunnel [24]. The auxiliary depth may reduce the uncertainty of the segmentation of objects having similar appearance information. An early attempt, in [25], was simply to connect the RGB and depth channel as four-channel inputs and feed them into a conventional RGB modal network. However, in most cases, this approach cannot take advantage of the complementary information in the depth map [26]. Recently, fully convolutional network architectures of the encoder–decoder type have been successful in the field of semantic segmentation. Hazirbas et al. proposed FuseNet [20], by fusing RGB and depth data in an encoder–decoder architecture, where the encoders use a VGG-16 [27] backbone to extract features. The feature maps from the depth encoder are fused into the RGB encoder as the network progresses, increasing the information features extracted, but the sparse fusion is less effective in extracting some types of shadow. Similarly, Sun et al. [28] designed two independent branches to extract features from RGB and depth images, respectively; they fuse the output features of the depth branch to the RGB branch using the attentional feature complementation (AFC) module. The depth map in the mine cave is sparser, with more complex shadows and noise. Most of these methods have complex structures and a fixed ratio for the fusion of RGB-D features. In this paper, we use an attention complementation network to extract features of the input, achieving competitive results with the latest techniques in RGB-D segmentation.

2.2. Semi-Supervised Learning

The primary purpose of semi-supervised learning is to use unlabeled data to build better learning programs. They can be divided into transfer learning, weakly-supervised learning, positive and unlabeled learning, and meta-learning [29]. Transfer learning [30] aims to apply knowledge from one or more source domains to knowledge from the target domain, to improve performance on the target task. Weakly-supervised learning reduces the dependence on data and can be divided into three categories: incomplete supervised data, inexact supervised data, and inaccurate supervised data. Incomplete supervised data means that only a subset of the training data is labeled. A representative approach is domain adaptation. Inexact supervised data indicates that the labels of the training instances are coarse; e.g., in the presence of multiple instance learning. Inaccurate supervised data means that the given labels have errors, for example in the situation of label noise learning. Meta-learning [31,32] aims to use previous knowledge and a few training examples to quickly learn new skills or adapt to new tasks, also known as 'learning to learn'. The meta-learning model is expected to adapt to the new environment encountered during the

training process. The adaptation process is essentially a mini-learning process that occurs during testing, but with limited exposure to the new task configuration.

Blast-hole segmentation is essentially a pixel classification problem. Based on weakly-supervised learning, we implement an SSL approach, combining incomplete supervised data and inexact supervised data. Our SSL method combines pseudo-labeling and self-training, which has two advantages over prevalent SSL methods: first, no additional network design is required, reducing parameter complexity. Second, few SSL methods can be applied to RGBD inputs, and our pseudo-labeling utilizes both RGB and depth information, resulting in a smaller error rate. However, semantic segmentation applications through SSL, especially in the mining domain, are rare and deserve more attention. Therefore, it is necessary to bridge the gap between SSL and the field of intelligent mining. We intend to extend the core idea of semi-supervised learning to RGBD semantic segmentation for pixel detection of blast-holes.

3. An SSL Method Utilizing ERF-AC-PSPNet

Inspired by [33], we designed a new model with the network seen structure in Figure 1. The model consists of two parts: a feature extraction network, and scene parsing network. It uses ERFNet for feature extraction, further extracts features using ACM, and uses PSPNet for feature fusion and scene parsing. ERFNet and PSPNet form our baseline network. RGB images of size $H \times W \times 3$ and depth images of size $H \times W \times 1$ are network inputs, and the fused features are acquired by the encoder and fed into the decoder. The output size is $H \times W \times C$, where we set $C = 2$ as representing the number of semantic categories.

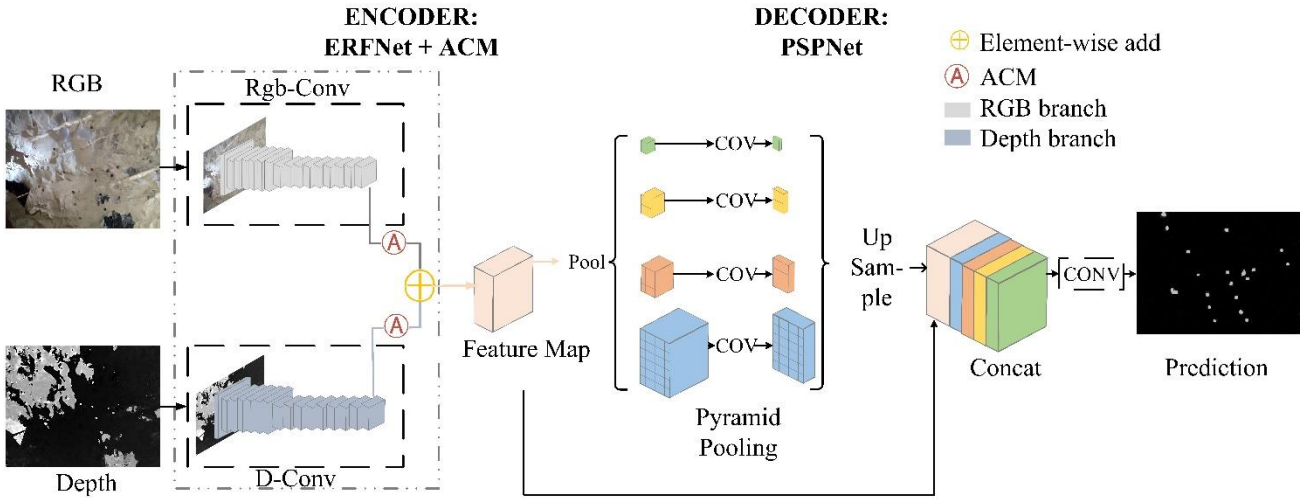

Figure 1. ERF-AC-PSPNet model structure.

In the following sections, we describe in detail how our model is designed and how the method achieves segmentation in a semi-supervised situation.

3.1. Feature Extraction

Specifically, an efficient residual factorized network (ERFNet) can be divided into two symmetric branches: the RGB branch, and the depth branch. They have roughly the same process: image resolution decrease via down-sampling. Despite the fact that some detail is lost in the down-sampling operation, the net consumption can be decreased by reducing the resolution of the feature map. The down-sampling operation carries out convolution, with a step size of two and average pooling in parallel. After down-sampling, the network uses a non-bottleneck connection layer. It uses a pair of 1×3 and 3×1 convolutional kernels connected in series, which reduces the number of parameters in the network, while maintaining the extraction capability. At the same time, for the smooth working of the gradient backward-propagation, the non-bottleneck connection layer is connected using

residuals, by adding the original input to the final output feature map. We chose the activate function ReLU [34], to give the resulting decomposition layer an inherently low computational cost and simplicity.

From these symmetric processes, we get two feature maps, RGB and depth, respectively. To facilitate subsequent processing, both feature maps have the same dimensions.

3.2. Attention Complementary Model

As Figure 2 shows, in some special scenes (e.g., mines, poorly lit rooms), the RGB images and the depth images contain distinct information. In order to collect as many features as possible from RGB images and depth images, we introduce a module that allows the network to concentrate on more information-rich regions, which is known as the attention complementary module (ACM); with the structure shown in Figure 3.

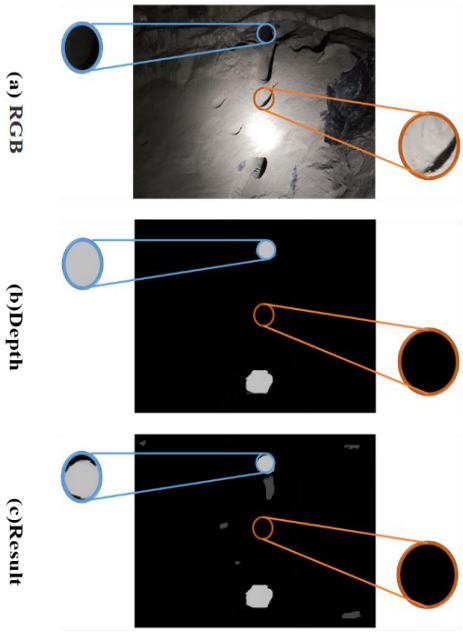

Figure 2. RGB and depth images have different feature distributions.

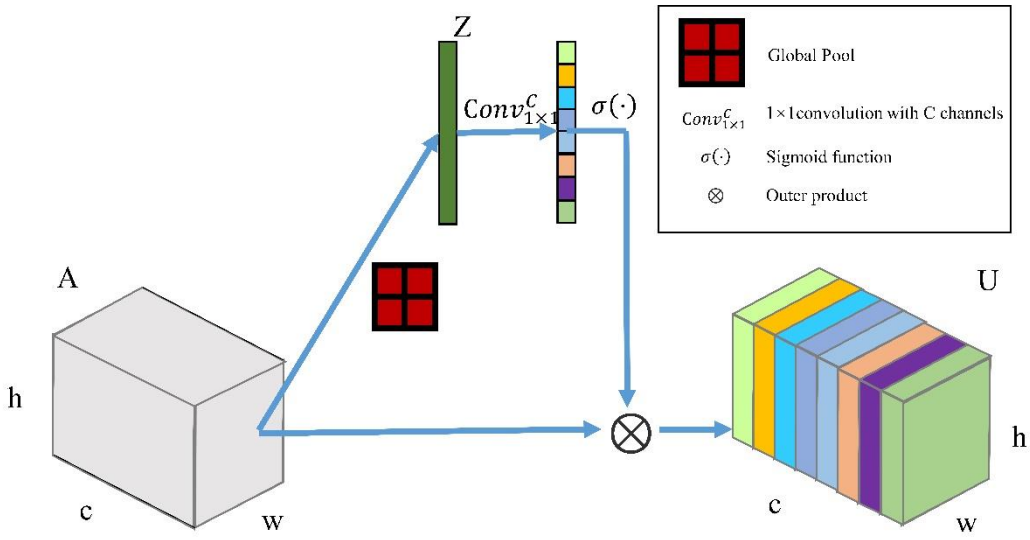

Figure 3. Attention complementary module (ACM).

Suppose input feature map $A = [A_1, \cdots, A_c] \in \mathbb{R}^{C\times H\times W}$, a global average pooling is first performed to obtain output $Z \in \mathbb{R}^{C\times H\times W}$, where C indicates the number of channels

and H, W indicate the height and width of the feature map, separately. The $k(k \in [1, C])$ channel of Z can be expressed as:

$$Z_k = \frac{1}{H \times W} \sum_{i}^{H} \sum_{j}^{W} A_k(i, j) \tag{1}$$

Z is subsequently restructured to make use of as many channels as Z with a 1×1 convolutional layer. This 1×1 convolutional layer is capable of exploiting the correlation between channels to derive the distribution of appropriate weights for these channels. The activation function sigmoid is used to process the convolution outcome, constraining the value of the weight vector $V \in \mathbb{R}^{C \times 1 \times 1}$ to values between 0 and 1. Finally, we take the outer product of A and V. The outcome $U \in \mathbb{R}^{C \times H \times W}$ can be denoted as:

$$u = A \otimes \sigma[\Phi(Z)] \tag{2}$$

where $\otimes$ denotes the outer product, σ denotes the S-shaped function, and Φ denotes the 1×1 convolution. In this way, the feature map A is transmuted into a new feature map U containing more valid information.

3.3. Pyramid Scene Parsing

At the heart of PSPNet is the fusion of multiple features. The contextual information in the feature maps cannot be fully and appropriately used if the up-sampling layer is directly connected. Therefore, we apply a PSPNet, to operate in parallel on the feature maps extracted by the ACMs.

Four convolutional layers of different step sizes are deployed on the input. Convolutional layers with smaller step sizes can acquire detailed information, while convolutional layers with larger step sizes can learn abstract information. These convolutions are then resized using bilinear up-sampling, and the result is overlaid into a new feature extraction layer. This layer is connected to a subsequent up-sampling process. The detailed network design is shown in Figure 4. The feature extraction layer down-samples 1/2, 1/4, and 1/8, respectively. There is also a branch that calculates the global feature vector by taking the average of each channel. Global feature vectors and the outputs of the 1/2, 1/4, and 1/8 feature layers are individually learned and stacked by additional convolutional layers, to form the final output. Reducing the number of channels in each convolutional result layer facilitates the integration and compression of information in the network, while leaving the information content unchanged. The activate function ReLU [34] here avoids gradient vanishing during backpropagation.

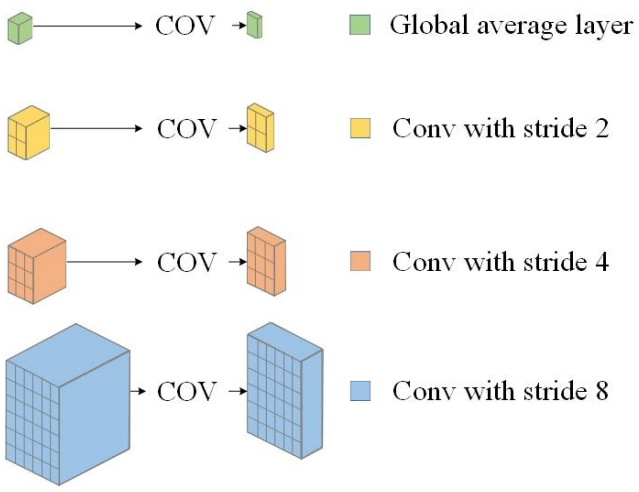

Figure 4. Structure of PSPNet.

3.4. Semi-Supervised Training

SSL is a learning method that combines supervised learning with unsupervised learning. SSL aims to make more accurate predictions with the aid of unlabeled data and some labeled data; in contrast to supervised learning, which only requires labeled data, and unsupervised learning, which only requires unlabeled data. Our method combines the pseudo-labeling method and the self-training method. The pseudo-labeling method relies on the high confidence of the pseudo-labels, which can be added to the training dataset as labeled data. The self-training algorithm uses the model's confident predictions to generate pseudo-labels of unlabeled data, adding more training data by using existing labeled data. Before the training, a temporary label will be given to the unprocessed data using a histograms of oriented gradients (HOG) algorithm [35] and the depth data. The pseudo-label given by this method works better than the label given directly using the HOG algorithm, and better labels will be beneficial to the training of the network [36]. The training process can be divided into three steps with this approach:

1. We give a temporary annotation to the RGB-D, which uses the depth and the HOG operator. It should be noted that this step is only performed on the unprocessed data and it is not the final result.
2. When the input image has a label, this is a fully supervised process and will not be repeated here. The SSL uses the previously labeled results as input for the network, and the network will feed the images several more times and repeat the training accompanied by a variant label.
3. In the process of repeated training, the output results will be compared with the temporarily labeled one again and again. Corrections will be made based on the depth data, to produce a new label for the image. This action will also be carried out several times.

4. Experiments and Results

To test the performance of the method, a series of experiments were conducted on a dataset containing 205 blast-hole images. All images in this dataset were captured from real mine scenes using a real-time acquisition system with a binocular stereo depth camera. The experimental setup is illustrated in Figure 5, and includes the explosive charging equipment and an end effector equipped with a Stereolabs industrial camera (ZED 2i, mode: 1080p, resolution: 3840 × 1080). We cropped each image to 512 × 512, to obtain the final dataset. The images of the blast-holes were taken by the real-time collection system, which is integrated with an industrial binocular stereo depth camera, and analyzed using the method above.

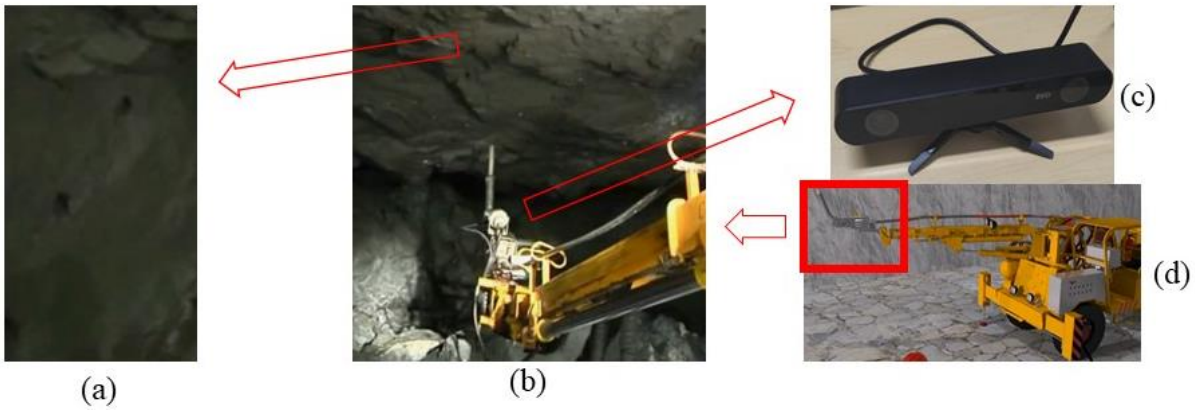

Figure 5. The experimental setup for blast-hole detection. (**a**) Blast-holes, (**b**) explosive charging end effecter, (**c**) ZED2i binocular camera, (**d**) explosive charging equipment.

4.1. Semi-Supervised Semantic Labeling and Training

There are three main stages in assigning labels to the data, HOG detection, depth-based correction, and edge extraction [37]. In Figure 6, a comparison is made between our method, HOG [35], and the edge drawing parameter free algorithm (EDCircles) [12]. The diagram shows, from left to right, the RGB, the ground truth, the HOG, the EDCircles, and our method. From (1), the results contain too many non-porous pixels in HOG, even if the target was detected. (2) and (3) took the non-hole target into account, whereas our method did not. In addition, the EDCircles showed a missed detection in (3), and the annotation in (1) is not comprehensive. Good results were achieved by all the methods in (2), similar to the ground truth. In general, our method has fewer non-porous pixel points within the target region, which is more accurate for labeling and more conducive to subsequent network training. The reasons for this problem were speculated to be as follows: 1. Inconsistent edges in RGB and depth images caused label imperfections [38]. 2. Some holes are located right at the border of the image, resulting in inaccurate detection and consequent effects on the label [39]. 3. Systematic error of the Canny operator for edge detection and contour extraction [37]. Fortunately, these labels were not the final results. The labeled data would be introduced into the model for training, to further optimize the semantic labels.

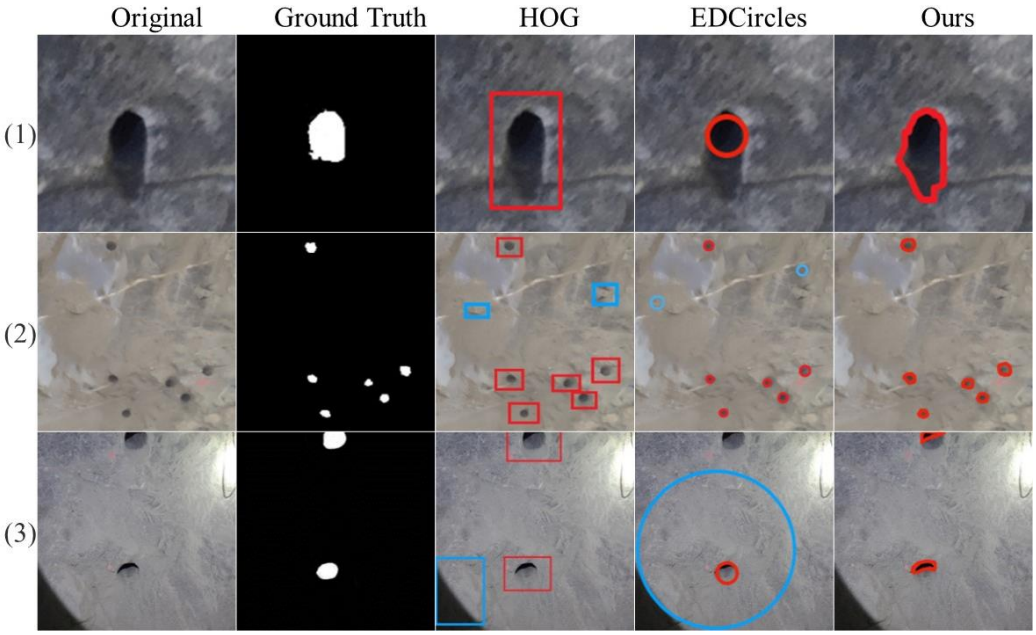

Figure 6. Comparison of the effectiveness of various methods for labeling a blast-hole. Comparison of the effectiveness of various methods for the labeled blast-hole. (**1**): The input sample contains less noise and shadows, and the hole is clear and intact; (**2**): the input sample contains a lot of noise; (**3**): the input sample contains shadows, and the blast-hole is at the edge.

The implementation of the network was based on the public platform Pytorch [40] +CUDA and CuDNN backend. We trained the model in the Windows 10 operating system with an Intel(R) Xeon(R) Gold 6248R CPU @ 3.0 GHz, 192 GB of RAM, and a single Tesla V100 NVlink GPU with 32 GB of RAM. Inspired by [15], we used a poly learning rate strategy, where the current learning rate was equal to the base learning rate multiplied by $\left(1 - \frac{epoch}{max_{epoch}}\right)^{power}$. The base learning rate was set to 0.01, epoch is the number of iterations, and max_{epoch} is the maximum of iterations; power controls the shape of the curve, and we set it to 0.9. We used a momentum of 0.9 and a weight decay of 0.0001. During the experiments, we noticed that a suitably large crop-size can produce a good performance, and that batching of the batch size in the normalization was very important too. The batch size was set to 16 for training, due to GPU limitations.

4.2. Evaluation Indices and Architecture

We obtained 205 images for training, validation, and testing. The evaluation of the segmentation effect was performed using two main indicators: intersection over union (IoU) [41] and Dice [42]. Their function expressions were, respectively: G_i denotes the ground truth pixel of hole i, P_j denotes the predicted pixel of hole j.

$$\text{IoU} = \frac{|G_i \cap P_i|}{|G_i \cup P_i|} \tag{3}$$

$$\text{Dice} = \frac{2 \times |G_i \cap P_i|}{|G_i| + |P_i|} \tag{4}$$

To facilitate the segmentation, and based on the symmetric encoder–decoder architecture of SegNet [43], we designed our networks inspired by ENet [44], ERFNet [17], and PSPNet [18]. A detailed description of the overall structure is given in Table 1. Out-F: number of feature maps at layer's output. Out-Res: output resolution for an input size of 512 × 512. RGB and depth feature fusion are accomplished in the encoder, and residual layers are also stacked in it. In general, there are two cases of residual layers used in advanced networks [44,45]: bottleneck designs, and non-bottleneck designs. Here we chose Non-bottleneck-1D, a redesigned residual layer based on a non-bottleneck that uses the 1D factor of the convolution kernel to exploit the efficiency of bottlenecks and the learning ability of non-bottlenecks in a good trade-off. It can efficiently use the minimum amount of residual layers to extract feature maps and achieve semantic segmentation. In order to gather more contextual information, while minimizing the sacrificing of learned features, we introduced the pyramid pooling module of PSPNet [18] to harvest different sub-regional representations, which form the final feature representation by up-sampling and concatenating layers.

Table 1. Layer Disposal of Our Proposed Network. 'Out-F': Number of Feature Maps at Layer's Output, 'Out-Res': Output Resolution for an Input Size of 512 × 512.

	Layer	Type	Out-F	Out-Res
ENCODER	0	Scaling	3	512 × 512
	1	Down-sampler block	16	256 × 256
	2	Down-sampler block	64	256 × 256
	3–7	5× Non-bt-1D	64	128 × 128
	8	Down-sampler block	128	128 × 128
	9	Non-bt-1D (dilated 2)	128	128 × 128
	10	Non-bt-1D (dilated 4)	128	128 × 128
	11	Non-bt-1D (dilated 6)	128	128 × 128
	12	Non-bt-1D (dilated 8)	128	128 × 128
	13	Non-bt-1D (dilated 2)	128	128 × 128
	14	Non-bt-1D (dilated 4)	128	128 × 128
	15	Non-bt-1D (dilated 6)	128	128 × 128
	16	Non-bt-1D (dilated 8)	64	256 × 256
	17	ACM fuses	64	256 × 256
DECODER	18a	Original feature map	128	256 × 256
	18b	Pooling and convolution	32	256 × 256
	18c	Pooling and convolution	32	128 × 128
	18d	Pooling and convolution	32	64 × 64
	18e	Pooling and convolution	32	32 × 32
	18	Up-sampler and concatenation	256	256 × 256
	19	Convolution	2	256 × 256
	20	Up-sampler	2	512 × 512

4.3. Segmentation Accuracy with Blast-Hole

We began our test with different proportions of labeled and unlabeled training data; 1/8, 1/4, 1/2, and 1 represent the proportion of labeled images in the dataset, respectively, and the rest of the images were unlabeled. The images with labels were selected randomly, to ensure the objectivity and fairness of the results. We compared the proposed method with FCN [14], U-Net [46], ENet [44], PSPNet [18], ERFNet [17], ACNet [22], Swin Transformer [47], ERF-PSPNet [48], and SegNet [43], and the results are presented in Table 2. When using 1/8, 1/4, $\frac{1}{2}$, and 1 scaled labeled blast-holes, we predicted IoU values of 0.810, 0.867, 0.923, 0.945, and Dice values of 0.849, 0.904, 0.958, 0.981, respectively. It can be seen that the accuracy of segmenting the blast-holes with our SSL method exceeded that of the other methods, and our method was less dependent on data labeling, making it more suitable for practical applications.

Table 2. Intersection over Union (IoU) and Dice index under different methods and different label settings.

Method	Labeled Data							
	1/8		1/4		1/2		Full	
	IoU	Dice	IoU	Dice	IoU	Dice	IoU	Dice
FCN	0.482	0.614	0.672	0.762	0.726	0.813	0.839	0.879
U-Net	0.498	0.647	0.668	0.781	0.731	0.804	0.844	0.880
ENet	0.521	0.647	0.738	0.829	0.801	0.850	0.916	0.954
PSPNet	0.524	0.650	0.741	0.833	0.807	0.848	0.928	0.967
ERFNet	0.530	0.673	0.745	0.830	0.798	0.860	0.921	0.958
ACNet	0.538	0.699	0.729	0.843	0.813	0.889	0.918	0.956
Swin Transformer	0.543	0.703	0.744	0.853	0.825	0.903	0.925	0.960
ERF-PSPNet	0.540	0.661	0.763	0.856	0.832	0.912	0.937	0.976
SegNet	0.502	0.661	0.671	0.801	0.748	0.829	0.853	0.891
Ours	0.810	0.849	0.867	0.904	0.923	0.958	0.945	0.981

Owing to limitations, we were only able to capture a restricted number of images of the holes, which made the training more difficult. The symmetric encoder–decoder architecture we adopted has the advantage of a small number of training parameters, which compensates well for this shortcoming. On the one hand, our proposed method employs unlabeled blast-hole images in the model training, and these images can generate additional supervised signals to train the model. On the other hand, the introduced ACM allows us to extract features from RGB-D data at a dynamic scale, whereas this scale is fixed for the other methods, which allows us to extract more features for segmentation. The results shown in Table 2 indicate that our method was significantly less dependent on the labeled data and outperformed the others for the same settings. For specific analysis, Figure 7 shows the pixel-wise level results presented by the various approaches when training the network with 1/4 annotated data; from left to right, RGB image, depth image, annotation, PSPNet, ERFNet, ERF-PSPNet, and our approach. The detection results in the first line are approximately the same, with only PSPNet showing a false detection in the second line. PSPNet, ERFNet, and ERF-PSPNet all show an erroneous detection in the fourth, and the third line even shows a missed check. The reasons for these situations are threefold in our analysis: 1. The effect of noise, occlusion, and shadows on RGB images on the network [38]. 2. The information content of RGB and depth is unequal, the ratio is fixed for feature extraction [22]. 3. Different training effects on the model during SSL [49]. On the one hand, our approach produces more accurate segmentation results by extending the idea of SSL to RGB-D semantic segmentation. On the other hand, it shows that using RGB-D data for scene perception in this uniform framework can solve the problem of unequal information and inconsistent background distribution between RGB and depth.

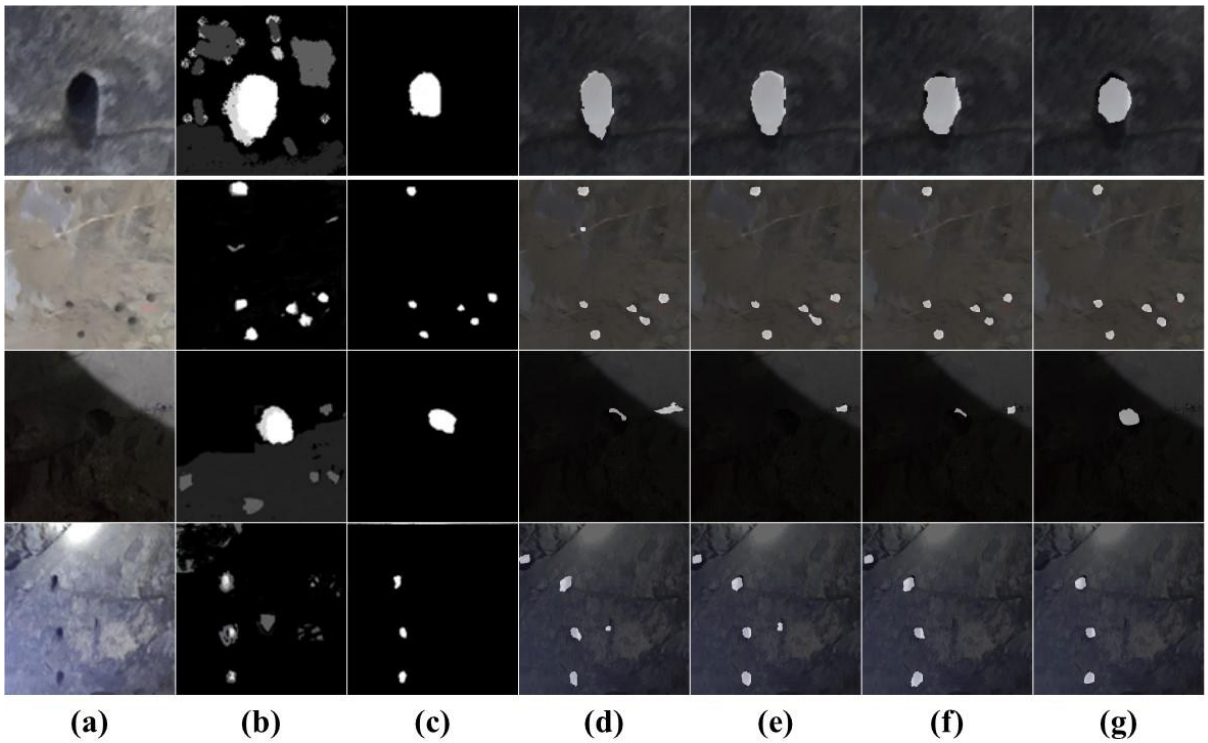

Figure 7. Qualitative examples of the segmentation of our captured image produced by our approach compared with the ground-truth annotation, PSPNet [18], ERFNet [17], and ERF-PSPNet [33]. From left to right: (**a**) RGB image, (**b**) depth image, (**c**) annotation, (**d**) PSPNet, (**e**) ERFNet, (**f**) ERF-PSPNet, (**g**) our approach.

5. Conclusions

In this paper, we present an SSL method for the segmentation of blast-holes. Compared with the common semantic segmentation models in the state-of-the-art, our approach focuses on improving the capacity of the model to extract features from RGB-D data and reducing the reliance of the model on labeled data. We propose introducing ACM into the symmetric encoder–decoder framework design, to reduce the loss of details in feature extraction. An SSL approach using pseudo-labeling and self-training is suggested to train the model with unlabeled data, which reduces the model's dependence on labeling. Experiments showed that our segmentation achieved an IoU of 0.810, 0.867, 0.923, 0.945, with 1/8, 1/4, 1/2, and full labeled data ratios. A balance was reached between accuracy and reliance on the label. This indicates its suitability for blast-hole detection in explosive loading applications.

Future works will involve in-depth experiments regarding the power consumption of the model and compression techniques (e.g., weight sharing) to further reduce the computational resources of the model. In addition, we would like to deploy this work on FPGA and extend it to other tasks, such as crack detection and urban mesh semantic segmentation.

Author Contributions: Conceptualization, Z.Z. and Y.L.; methodology, Z.Z., H.D. and Y.L.; software, Z.Z. and Q.X.; validation, Z.Z. and G.L.; formal analysis, Z.Z. and H.D.; investigation, Z.Z. and Y.L.; resources, Z.Z., H.D. and Y.L.; data curation, Z.Z.; writing—original draft preparation, Z.Z. and Y.L.; writing—review and editing, Z.Z., H.D. and Q.X. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

Acknowledgments: We are grateful to the High Performance Computing Center of Central South University for assistance with the computations.

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

References

1. Ranängen, H.; Lindman, Å. A Path towards Sustainability for the Nordic Mining Industry. *J. Clean. Prod.* **2017**, *151*, 43–52. [CrossRef]
2. Golik, V.; Komashchenko, V.; Morkun, V.; Irina, G. Improving the Effectiveness of Explosive Breaking on the Bade of New Methods of Borehole Charges Initiation in Quarries. *Metall. Min. Ind.* **2015**, *7*, 383–387. [CrossRef]
3. Lala, A.; Moyo, M.; Rehbach, S.; Sellschop, R. Productivity in Mining Operations: Reversing the Downward Trend. *AusIMM Bull.* **2016**, 46–49. [CrossRef]
4. Yang, D.; Zhao, Y.; Ning, Z.; Lv, Z.; Luo, H. Application and Development of an Environmentally Friendly Blast Hole Plug for Underground Coal Mines. *Shock. Vib.* **2018**, *2018*, e6964386. [CrossRef]
5. Duda, R.O.; Hart, P.E. Use of the Hough Transformation to Detect Lines and Curves in Pictures. *Commun. ACM* **1972**, *15*, 11–15. [CrossRef]
6. Nakanishi, M.; Ogura, T. Real-Time CAM-Based Hough Transform Algorithm and Its Performance Evaluation. *Mach. Vis. Appl.* **2000**, *12*, 59–68. [CrossRef]
7. Shaked, D.; Yaron, O.; Kiryati, N. Deriving Stopping Rules for the Probabilistic Hough Transform by Sequential Analysis. *Comput. Vis. Image Underst.* **1996**, *63*, 512–526. [CrossRef]
8. Xu, L.; Oja, E.; Kultanen, P. A New Curve Detection Method: Randomized Hough Transform (RHT). *Pattern Recognit. Lett.* **1990**, *11*, 331–338. [CrossRef]
9. Han, J.H.; Kóczy, L.; Poston, T. Fuzzy Hough Transform. *Pattern Recognit. Lett.* **1994**, *15*, 649–658. [CrossRef]
10. Chen, T.-C.; Chung, K.-L. An Efficient Randomized Algorithm for Detecting Circles. *Comput. Vis. Image Underst.* **2001**, *83*, 172–191. [CrossRef]
11. Ayala-Ramirez, V.; Garcia-Capulin, C.H.; Perez-Garcia, A.; Sanchez-Yanez, R.E. Circle Detection on Images Using Genetic Algorithms. *Pattern Recognit. Lett.* **2006**, *27*, 652–657. [CrossRef]
12. Akinlar, C.; Topal, C. EDCircles: A Real-Time Circle Detector with a False Detection Control. *Pattern Recognit.* **2013**, *46*, 725–740. [CrossRef]
13. Jiao, L.; Zhang, F.; Liu, F.; Yang, S.; Li, L.; Feng, Z.; Qu, R. A Survey of Deep Learning-Based Object Detection. *IEEE Access* **2019**, *7*, 128837–128868. [CrossRef]
14. Long, J.; Shelhamer, E.; Darrell, T. Fully Convolutional Networks for Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Boston, MA, USA, 7–12 June 2015; pp. 3431–3440.
15. Chen, L.-C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic Image Segmentation with Deep Convolutional Nets and Fully Connected CRFs. *arXiv* **2016**, arXiv:1412.7062.
16. Lu, Y.; Chen, Y.; Zhao, D.; Chen, J. Graph-FCN for Image Semantic Segmentation. In Proceedings of the Advances in Neural Networks—ISNN 2019, Moscow, Russia, 10–12 July 2019; Lu, H., Tang, H., Wang, Z., Eds.; Springer International Publishing: Cham, Switzerland, 2019; pp. 97–105.
17. Romera, E.; Álvarez, J.M.; Bergasa, L.M.; Arroyo, R. ERFNet: Efficient Residual Factorized ConvNet for Real-Time Semantic Segmentation. *IEEE Trans. Intell. Transp. Syst.* **2018**, *19*, 263–272. [CrossRef]
18. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid Scene Parsing Network. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 2881–2890.
19. Cheng, Y.; Cai, R.; Li, Z.; Zhao, X.; Huang, K. Locality-Sensitive Deconvolution Networks With Gated Fusion for RGB-D Indoor Semantic Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3029–3037.
20. Hazirbas, C.; Ma, L.; Domokos, C.; Cremers, D. FuseNet: Incorporating Depth into Semantic Segmentation via Fusion-Based CNN Architecture. In Proceedings of the Computer Vision—ACCV 2016, Taipei, Taiwan, 20–24 November 2016; Lai, S.-H., Lepetit, V., Nishino, K., Sato, Y., Eds.; Springer International Publishing: Cham, Switzerland, 2017; pp. 213–228.
21. Park, S.-J.; Hong, K.-S.; Lee, S. RDFNet: RGB-D Multi-Level Residual Feature Fusion for Indoor Semantic Segmentation. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 4980–4989.
22. Hu, X.; Yang, K.; Fei, L.; Wang, K. ACNET: Attention Based Network to Exploit Complementary Features for RGBD Semantic Segmentation. In Proceedings of the 2019 IEEE International Conference on Image Processing (ICIP), Taipei, Taiwan, 22–25 September 2019; pp. 1440–1444.
23. Chapelle, O.; Scholkopf, B.; Zien, A. Semi-Supervised Learning (Chapelle, O. et al., Eds.; 2006). *IEEE Trans. Neural Netw.* **2009**, *20*, 542. [CrossRef]
24. Mo, Y.; Wu, Y.; Yang, X.; Liu, F.; Liao, Y. Review the State-of-the-Art Technologies of Semantic Segmentation Based on Deep Learning. *Neurocomputing* **2022**, *in Press*. [CrossRef]
25. Couprie, C.; Farabet, C.; Najman, L.; LeCun, Y. Indoor Semantic Segmentation Using Depth Information. *arXiv* **2013**, arXiv:1301.3572.
26. Ngiam, J.; Khosla, A.; Kim, M.; Nam, J.; Lee, H.; Ng, A.Y. Multimodal Deep Learning. In Proceedings of the 28th International Conference on Machine Learning, ICML 2011, Bellevue, WA, USA, 28 June 28–2 July 2 2011.
27. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2015**, arXiv:1409.1556.

28. Sun, L.; Yang, K.; Hu, X.; Hu, W.; Wang, K. Real-Time Fusion Network for RGB-D Semantic Segmentation Incorporating Unexpected Obstacle Detection for Road-Driving Images. *IEEE Robot. Autom. Lett.* **2020**, *5*, 5558–5565. [CrossRef]
29. Yang, X.; Song, Z.; King, I.; Xu, Z. A Survey on Deep Semi-Supervised Learning. *arXiv* **2021**, arXiv:2103.00550.
30. Tan, C.; Sun, F.; Kong, T.; Zhang, W.; Yang, C.; Liu, C. A Survey on Deep Transfer Learning. In Proceedings of the Artificial Neural Networks and Machine Learning—ICANN 2018, Rhodes, Greece, 4–7 October 2018; Kůrková, V., Manolopoulos, Y., Hammer, B., Iliadis, L., Maglogiannis, I., Eds.; Springer International Publishing: Cham, Switzerland, 2018; pp. 270–279.
31. Vanschoren, J. Meta-Learning: A Survey. *arXiv* **2018**, arXiv:1810.03548.
32. Hospedales, T.; Antoniou, A.; Micaelli, P.; Storkey, A. Meta-Learning in Neural Networks: A Survey. *arXiv* **2020**, arXiv:2004.05439. [CrossRef] [PubMed]
33. Zhou, K.; Wang, K.; Yang, K. A Robust Monocular Depth Estimation Framework Based on Light-Weight ERF-Pspnet for Day-Night Driving Scenes. *J. Phys. Conf. Ser.* **2020**, *1518*, 012051. [CrossRef]
34. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet Classification with Deep Convolutional Neural Networks. In Proceedings of the Advances in Neural Information Processing Systems, Lake Tahoe, NV, USA, 3–6 December 2016; Curran Associates, Inc.: Red Hook, NY, USA, 2012; Volume 25.
35. Dalal, N.; Triggs, B. Histograms of Oriented Gradients for Human Detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893.
36. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. Mixup: Beyond Empirical Risk Minimization. *arXiv* **2018**, arXiv:1710.09412.
37. Rong, W.; Li, Z.; Zhang, W.; Sun, L. An Improved Canny Edge Detection Algorithm. In Proceedings of the 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China, 3–6 August 2014; pp. 577–582.
38. Sun, L.; Zhao, C.; Stolkin, R. Weakly-Supervised DCNN for RGB-D Object Recognition in Real-World Applications Which Lack Large-Scale Annotated Training Data. *IEEE Sens. J.* **2019**, *19*, 3487–3500. [CrossRef]
39. Zou, Z.; Shi, Z.; Guo, Y.; Ye, J. Object Detection in 20 Years: A Survey. *arXiv* **2019**, arXiv:1905.05055.
40. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L.; et al. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In Proceedings of the Advances in Neural Information Processing Systems, Vancouver, BC, Canada, 8–14 December 2019; Curran Associates, Inc.: Red Hook, NY, USA, 2019; Volume 32.
41. Everingham, M.; Eslami, S.M.A.; Van Gool, L.; Williams, C.K.I.; Winn, J.; Zisserman, A. The Pascal Visual Object Classes Challenge: A Retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [CrossRef]
42. Maimon, O.; Rokach, L. (Eds.) *Data Mining and Knowledge Discovery Handbook*; Springer US: Boston, MA, USA, 2010; ISBN 978-0-387-09822-7.
43. Badrinarayanan, V.; Handa, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Robust Semantic Pixel-Wise Labelling. *arXiv* **2015**, arXiv:1505.07293.
44. Paszke, A.; Chaurasia, A.; Kim, S.; Culurciello, E. ENet: A Deep Neural Network Architecture for Real-Time Semantic Segmentation. *arXiv* **2016**, arXiv:1606.02147.
45. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
46. Ronneberger, O.; Fischer, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In Proceedings of the Medical Image Computing and Computer-Assisted Intervention—MICCAI 2015, Munich, Germany, 5–9 October 2015; Navab, N., Hornegger, J., Wells, W.M., Frangi, A.F., Eds.; Springer International Publishing: Cham, Switzerland, 2015; pp. 234–241.
47. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical Vision Transformer Using Shifted Windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10012–10022.
48. Sun, L.; Wang, K.; Yang, K.; Xiang, K. See Clearer at Night: Towards Robust Nighttime Semantic Segmentation through Day-Night Image Conversion. In Proceedings of the Artificial Intelligence and Machine Learning in Defense Applications, Strasbourg, France, 19 September 2019; SPIE: Bellingham, WA, USA, 2019; Volume 11169, pp. 77–89.
49. Van Engelen, J.E.; Hoos, H.H. A Survey on Semi-Supervised Learning. *Mach. Learn.* **2020**, *109*, 373–440. [CrossRef]