applied sciences

MDPI

Article

Accurate Instance-Based Segmentation for Boundary Detection in Robot Grasping Application

Hong Hai Hoang * and Bao Long Tran

School of Mechanical Engineering, Hanoi University of Science and Technology, Hanoi 100000, Vietnam; hoanghai.aoibk@gmail.com

* Correspondence: hai.hoanghong@hust.edu.vn; Tel.: +84-093-449-3466

Abstract: With the rapid development of cameras and deep learning technologies, computer vision tasks such as object detection, object segmentation and object tracking are being widely applied in many fields of life. For robot grasping tasks, object segmentation aims to classify and localize objects, which helps robots to be able to pick objects accurately. The state-of-the-art instance segmentation network framework, Mask Region-Convolution Neural Network (Mask R-CNN), does not always perform an excellent accurate segmentation at the edge or border of objects. The approach using 3D camera, however, is able to extract the entire (foreground) objects easily but can be difficult or require a large amount of computation effort to classify it. We propose a novel approach, in which we combine Mask R-CNN with 3D algorithms by adding a 3D process branch for instance segmentation. Both outcomes of two branches are contemporaneously used to classify the pixels at the edge objects by dealing with the spatial relationship between edge region and mask region. We analyze the effectiveness of the method by testing with harsh cases of object positions, for example, objects are closed, overlapped or obscured by each other to focus on edge and border segmentation. Our proposed method is about 4 to 7% higher and more stable in IoU (intersection of union). This leads to a reach of 46% of mAP (mean Average Precision), which is a higher accuracy than its counterpart. The feasibility experiment shows that our method could be a remarkable promoting for the research of the grasping robot.

Keywords: instance segmentation; mask R-CNN; grasping; robot; 3D camera; edge region

Citation: Hoang, H.H.; Tran, B.L. Accurate Instance-Based Segmentation for Boundary Detection in Robot Grasping Application. *Appl. Sci.* **2021**, *11*, 4248. https://doi.org/10.3390/app11094248

Academic Editor: DaeEun Kim

Received: 18 March 2021
Accepted: 4 May 2021
Published: 7 May 2021

1. Introduction

In the robot grasping field, the accuracy of instance segmentation plays an important role. It decides the position controlling, the open angle of the robot's arm. Moreover, it is the first step of finding object pose estimation, which is necessary for proposing a possible grasping plan [1–3]. If the detailed information at the edge region is overlooked, it will cause the negative situation that the arm of the robot may collide with the objects. Thus, the accuracy of the segmentation results is crucial in this field.

The advent of Convolution Neural Network (CNNs) in the 1990s opened a strong step forward for handling computer vision tasks in the future. In 2016, the Region-based Convolutional Neutral Network method (R-CNN) [4] was proposed and reached excellent object detection accuracy by using a deep CNN to classify object proposals. Then, Fast R-CNN [5] and Faster R-CNN [6] introduced later greatly improved both the accuracy of target detection and time processing. For object segmentation, based on the advantages of previous detection principles, Mask R-CNN [7] upgraded object detection and segmentation in image exceptionally. Nonetheless, the state-of-the-art Mask R-CNN did not consistently make accurate predictions, especially when objects were placed obscure to each other or there was a lack of light. Solving these problems, a scoring path was added to the mask head to increase the supervision of the mask in MS R-CNN [8]. Mask Refined R-CNN (MR R-CNN) [9] substantially improved the accuracy segmentation in the entire object by combining feature layers, which focus on the global and detailed information.

Although many works based on 2D data have achieved high accuracy in segmentation, they all have the same limitations. The image brightness and object obstacle reduce the performance of these works, and the pixels at the object edge are usually misclassified. Though 3D segmentation algorithms can easily take all the pixels of objects based on spatial location of object, it is impossible to obtain a good segmentation result without considering the features of 2D image. In addition, the state-of-the-art 3D segmentation model works well at part segmentation, scene semantic parsing, but not for the entire object. Motivated by these disadvantages and advantages, we propose a method that integrates Mask R-CNN and 3D segmentation algorithms for improving the accurate instance segmentation in the object edge region. The main novelty of our works can be summarized as follows:

- Adding a 3D process branch to conserve full pixels of objects by Difference of Normals-Based Segmentation [10]. Dealing with multiple objects and occlusion, which leads to the misclassification in Mask R-CNN.
- Distinguishing the interesting edge regions based on the relationship with the original masks.
- Continuously considering the spatial distance between edge regions and each mask region to categorize the edge region by Euclidean Cluster extraction [11].

The idea of combining 2D neural network model and 3D algorithms will open many opportunities in the future. With this proposed method, robots have full and exact awareness of object region, object localization and accomplish the purpose of grasping better. In addition, it can be applied to many different segmentation models to improve their performance.

This paper is organized in the following context: Section 2 shows the research related to our purpose. Mask R-CNN pipeline and the combination of Mask R-CNN and 3D algorithms are presented in Section 3. Section 4 shows the experiments and the result of our approach. The discussion, conclusion and further work are detailed in the last two sections.

2. Related Works

2.1. 2D Approach

In recent years, many researchers, involving both 2D and 3D approaches, have been approved to upgrade the segmentation task. In particular, apart from Mask R-CNN, Unet [12] was designed for biomedical image segmentation. It consisted of a contracting path and an expansive path. The contracting path was a typical architecture of a convolutional network. The essential part was the expansive path, which aimed to upsample the feature map and improve the accuracy, in detail, in localization and segmentation.

Improvement of the Mask-RCNN object segmentation algorithm [13] handled the poor masks of Mask R-CNN by combining with Grabcut [14]. However, Grabcut depends too much on the contrast of the input images and does not consider the detailed information in segmentation generally. Moreover, it can preserve the foreground but cannot classify it.

Fast Vehicle and Pedestrian Detection Using Improved Mask R-CNN [15] introduced a novel Side Fusion feature pyramid network (SF-FPN) by adding a feature pyramid network (FPN) [16] to obtain more precise feature semantic information. This was the most important part that helped researchers increase the accuracy.

Mask Scoring R-CNN (MS R-CNN) deals with problems that the predicted mask was not qualified to; in addition, it is confident. This model has a network block to learn the quality of the predicted instance masks and calibrate the misalignment between mask quality and mask score. Therefore, the model noticeably gained the instance segmentation performance.

Mask-Refined R-CNN (MR R-CNN) introduced a new semantic segmentation layer that realized feature fusion by constructing a feature pyramid network and summing the forward and backward transmissions of feature maps of the same resolution. It was applied for the semantic segmentation, while the other FPN structures were selected for object detection. From this method, the network can determine the characteristics of various scales simultaneously and improve the sensitivity to the global and detailed information. In comparison to MS R-CNN and Path Aggregation Network (PAN) [17], MR R-CNN had a remarkable performance in the prediction of large objects.

In summary, these above models based on 2D dataset input accomplish the appreciable improvement of Mask R-CNN detailed information segmentation. However, these models only consider the 2D feature of the image. It is hard to admit that these models are the complete solutions when practicing with dimly lit environments or obstructed objects.

2.2. 3D Approach

Dealing with point cloud data structure, PointNet [18], a novel type of neural network was introduced in 2017, which directly took point cloud and applied it to many tasks such as object detection, part segmentation and scene semantic parsing. This novel architecture reached a state-of-the-art result on standard benchmarks. To achieve this outcome, it not only considered just three coordinates (x,y,z), but also some additional dimensions such as computing normals and other local or global features. With the input data as point cloud, meshes or 3D voxel grids, the information of the border region is not declined even in harsh circumstances.

However, dealing with just 3D data establishes a gap that these structures of input data are not regular input data formats. It causes an adversity in the implementation and development of the model in a particular application. Thus, in this paper, we incorporate color image and depth information to surpass the problem of the two approaches. This will enable the advantages of achievement in speed and accuracy of 2D network and, at the same time, study the pixels at the harsh region by 3D algorithms.

3. Experimental Methodology

3.1. Mask R-CNN Pipeline

Instance segmentation is an important and difficult task since it is not only classification, localizing the objects in the image but also segmenting exactly the object region. Thus, instance segmentation is known as object detection and semantic segmentation. As suggested in Section 1, the Region-based Convolutional Neural Network (R-CNN) meets excellent object detection accuracy by using a deep convolutional network to classify object proposals. First of all, the proposals are obtained by applying the entire input image to Selective Search [19] algorithm. This step approximately extracts 2000 region proposals for each image. Then, every single region proposal is used as the input of a convolutional neural network (CNN) [20] to extract the feature and classify the region. In other words, all approximate 2000 region proposals are computed separately without sharing together, which actually takes a long time. Unlike R-CNN, instead of computing each of 2000 regions, fast R-CNN uses only one CNN for the entire input image. Therefore, fast R-CNN needs less time to achieve object detection than R-CNN. After surpassing the CNN, the Feature Map has a smaller size than the original input image. Fast R-CNN introduces RoI pooling to reshape the size of feature maps as the same size. This process helps the computation progress easily share together and reduce the space of computation. However, Selective Search needs much time to get the proposals and usually extracts too many regions. Similar to Fast R-CNN, Faster R-CNN computes the feature map on the entire image but then it used Region Proposal Network instead to propose the regions, which potentially included the objects. It dramatically upgrades the accuracy and speed of object detection tasks.

Based on state-of-the-art real-time Faster R-CNN, in aspect classification, Mask R-CNN introduced ROI Align instead of RoI Pooling in Faster R-CNN. Although it did not have a significant impact on the bounding box, RoI Pooling is not aligned according to the pixel one by one (pixel-to-pixel alignment) and has a large negative effect on predicting pixel-accurate masks. The accuracy of the mask increased significantly after using ROI Align because ROI Align can solve the problem that the network cannot consider the relationship between the pixels at the object edge, and these pixels will be misclassified. Parallel to the object detection branch, a mask branch is added for only semantic segmentation tasks.

Experiment results show that Mask R-CNN has poor accuracy in imaging the scene segmentation edge. Mask R-CNN does not usually determine full object regions because the foreground at the edge of the objects is misunderstood as background. Especially

in the robot grasping field, which has many occlusions between objects, the generated object mask usually does not fully cover the object. This poor mask will be used as the representation of the object so that a good grasp rectangle and a good grasp localization for grasping plan [21–23] cannot be proposed.

Taking the advantages of Mask R-CNN in object segmentation, we boost the accuracy in the edge areas' and border areas' segmentation by combing Mask R-CNN and depth information as shown in Figure 1.

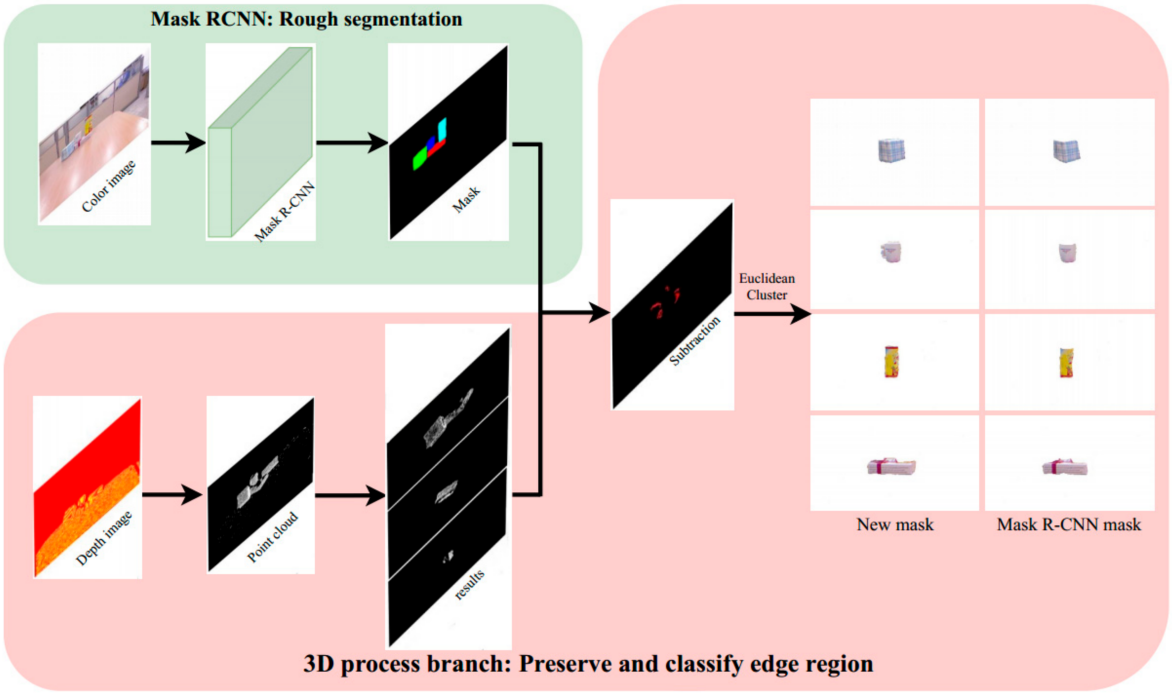


Figure 1. Diagram of combination between original Mask R-CNN and 3D segmentation algorithms. A point cloud process branch goes together with Mask R-CNN to operate 3D segmentation algorithms. The result after applying our proposed method to objects in full.

3.2. Proposed Method

We propose a method that combines Mask R-CNN and 3D segmentation algorithms, which are Difference of Normals (DoN)-Based Segmentation and Euclidean Cluster Extraction. First, we add a 3D process branch to save all pixels of objects and edge regions, which are usually treated as background if just considering only 2D features. Then, the distance between these edges and central regions is submitted to consider the relationship between edge regions and object class. In robot grasping, objects are usually disorganized, scalar, close together, overlapping, and obscured by each other. These things make an enormous hurdle for Mask R-CNN on reaching the high accurate segmentation. Taking advantage of the 3D information, 3D segmentation algorithms are applied to deal with this barrier and enhance the accuracy of Mask R-CNN.

Certainly, Figure 2 depicts our detailed works. The first step is rejecting almost the background to reduce the computation cost, since objects are usually located in a specified region that is limited by the workspace of the robot. Thus, based on the position of objects in space, objects out of the workspace of robots could be easily denied. This step dismissed most of the background and significantly decreased computation on the following segmentation steps. After generating the point cloud from the depth map acquired from Kinect V1 [24], the plane in which the objects are arranged is dismissed by random sample consensus (RANSAC) [25]. The numbers of the point cloud decreased dramatically, and the cost of computation also significantly decreased.

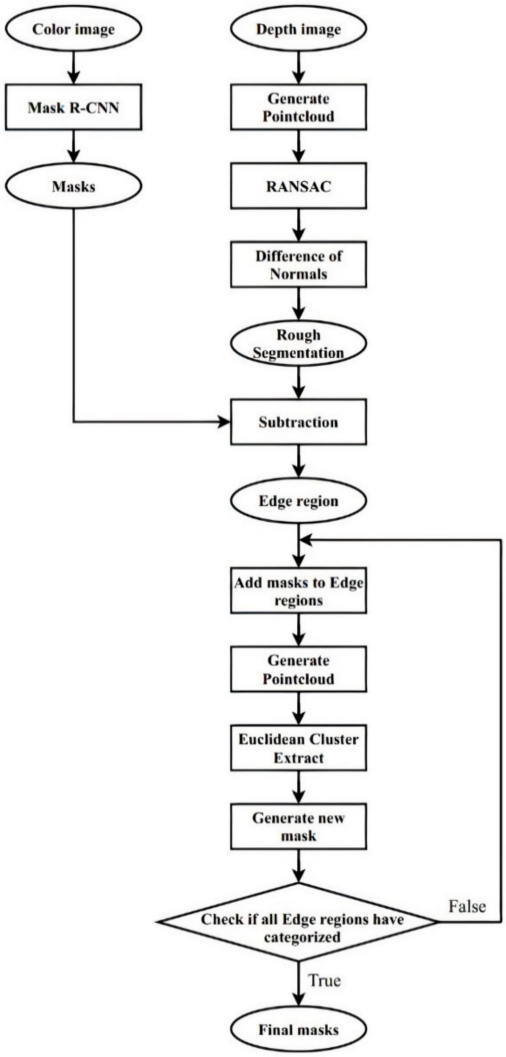


Figure 2. Flow chart of our approach for improving edge region segmentation.

The second step is applying DoN for the rough segmentation. Unlike Grabcut, DoN only depends on the Difference of Normals of points but does not depend on the color of the image. Thus, it does not miss any region even though the edge region. DoN compares, at each point **p**, the responses of the operator across two different radii $r_s < r_l$. Figure 3 illustrates the effect of the support radius on the estimated surface normal for a point cloud. Formally, the Difference of Normals operator of any point **p** in a point cloud is defined as follows [10,26].

$$\Delta_{\hat{n}}(p, r_s, r_l) = \frac{\hat{n}(p, r_s) - \hat{n}(p, r_l)}{2} \quad (1)$$

where $r_l, r_s \in \mathbb{R}$, $r_s < r_l$, and $\hat{n}(p, r)$ is the surface normal estimate at point **p**, given the support radius r. DoN is implemented with the support of PCL following these steps for segmentation [26]:

1. Estimate the normals for every point using a large support radius of r_l.
2. Estimate the normals for every point using the small support radius of r_s.
3. For every point, the normalized Difference of Normals for every point, as defined above.
4. Filter the resulting vector field to isolate the points belonging to the scale/region of interest.

Figure 3 confirms that Grabcut segmentation does not work more effectively than Difference of Normals. Practice in a situation as described in Figure 3a, Mask R-CNN gains

an unimpressive result as shown in Figure 3b. Expected to improve that result, however, Grabcut is limited by the low resolution and the contrast of input image taken from a 3D camera. Whereas, DoN can roughly separate the object without forgetting any edge regions. Compared with the combined Mask R-CNN and Grabcut method proposed by Xin Wu as Figure 3c describes, Grabcut denies the regions near the border area, while DoN, as in Figure 3d, can save more misclassified pixels.

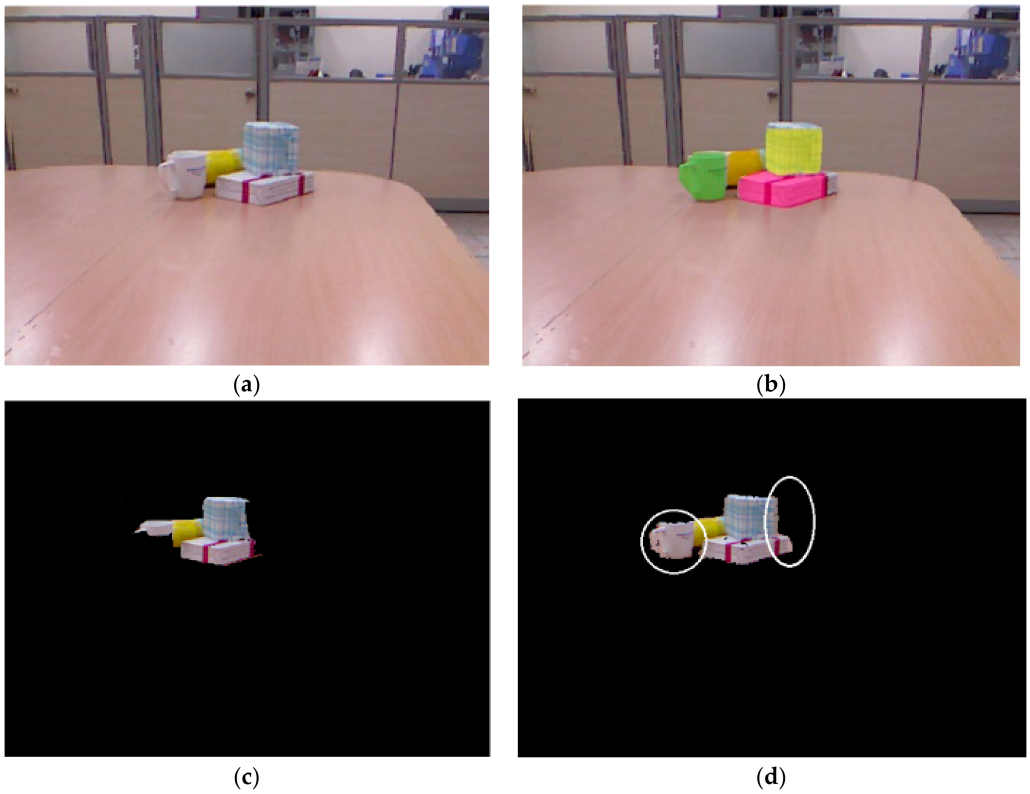
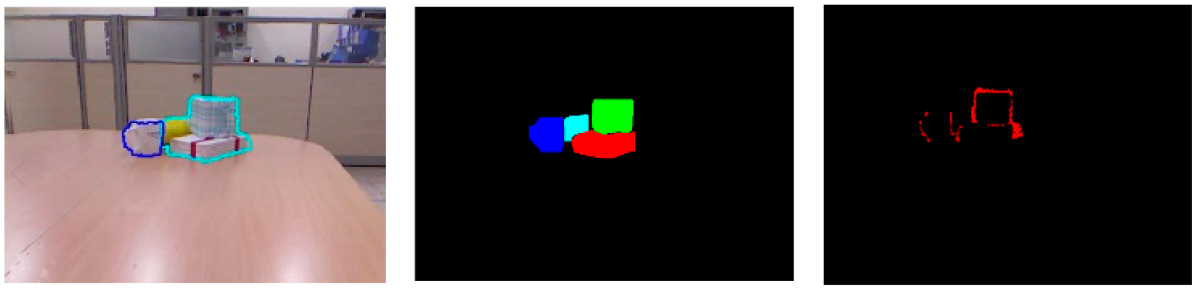


Figure 3. Comparison between results of Mask R-CNN (original) with GrabCut and DoN. The weakness of Grabcut compared with DoN is shown inside the white circle. (**a**) Original image. (**b**) Mask R-CNN. (**c**) Grabcut. (**d**) DoN.

The third step is distinguishing the border region and the missing parts that object Mask R-CNN does not cover. These regions are simply defined by subtraction of the regions covered by DoN contours and all Mask R-CNN masks as in Figure 4. The subtracted parts include all the areas near the border objects, which are treated as background by Mask R-CNN. Most of the entire objects are protected though the contours are not smooth because of the downsampling step.

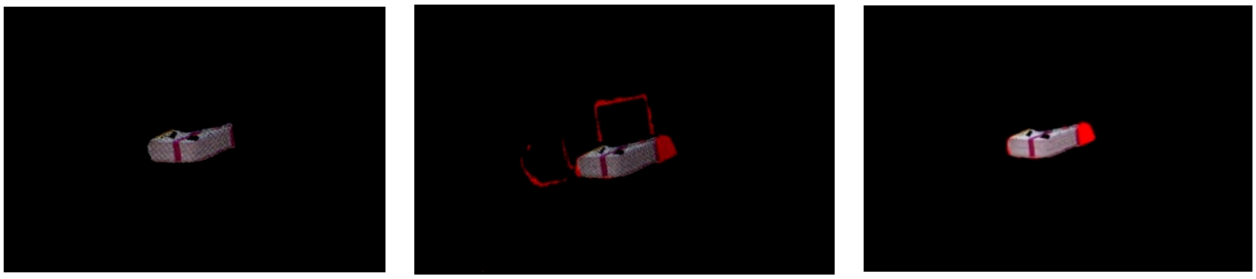
The final step is classifying these edge parts and putting them into the corresponding masks generated by Mask R-CNN. To overcome this issue, Euclidean Cluster Extraction is considered suitable for this mission. As every Mask R-CNN mask has interaction areas with subtracted regions, Euclidean distance is reasonable solution. Thus, each Mask R-CNN mask region was then sequentially matched to the uncategorized region. The matched image will contain only one object mask and the subtracted region. It will be used as the reference to extract the corresponding depth image. An example of this process is presented in Figure 5. Obviously, we obtain the separated object masks from Mask R-CNN, and Figure 5a shows the white box mask of one of them. The edge regions obtained in the previous process is put into one mask region as Figure 5b. To highlight the edge region, we set it in red color. These regions possibly belong to many objects, and it can be seen that they may include the box edge region and the cup bounder region. However, they have a spatial relationship with the classified Mask R-CNN mask. Clearly, the white box edge

region is closer to the white box more than the other regions. Applying Euclidean Cluster Extraction will achieve the result as in Figure 5c, which contains the white box mask and red edge region. The full object region is now covered, and a new full mask is generated.



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

Figure 4. Illustration of subtraction between (**a**): contours generated by DoN, (**b**): contours generated by Mask R-CNN and (**c**) subtracted region. The misclassified regions of Mask R-CNN are then segmented by Euclidean Cluster Extraction.



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

Figure 5. Illustration of classifying the edge region (**a**): original white box Mask R-CNN mask, (**b**): mask region and edge region, (**c**): edge region of the box is classified.

Euclidean Cluster Extraction clusters the fully three-dimensional object, which is used to recover 2D images and extract new masks of the object. However, in the next iteration, the previous mask is replaced by the new mask to reduce the computation cost. Next, the edge regions and next mask regions continuously repeat the process until the final one. The classifying edge region stops when the final edge categorizing is complete. Adapted from the Point Cloud Library (PCL) tutorial that operated Euclidean Cluster Extraction by following this procedure [8], this work is achieved.

1. Create a Kd-tree representation for the input point cloud dataset P.
2. Set up an empty list of clusters C, and a queue of the points that need to be checked Q.
3. Then, for every $p_i \in P$, perform the following steps:
   - Add p_i to the current queue Q.
   - For every point $p_i \in Q$ do:
     - Search for the P_i^k set of point neighbors of p_i in a sphere with radius $r < d_{th}$.
     - For every neighbor $p_i^k \in P_i^k$, check if the point has already been processed, and if not add it to Q.
   - When the list of all points in Q has been processed, add Q to the list of clusters C, and reset Q to an empty list.
4. The algorithm terminates when all points $p_i \in P$ have been processed and are now part of the list of point clusters C.

4. Experimental Results and Analysis

4.1. Experiment Preparation

The experiments were built with a 3D camera, Kinect V1, object including a red box, a cup, a cylinder, a white box, a computer equipped Intel core i5 9th Gen and Nvidia GeForce GTX 1650 graphics card. Objects were disorderly located from 0.4 m to 3.5 m far from the camera since the camera could not capture the distance of the object out of this space. An application was developed by support of C#, Kinect for Window SDK 1.8 to connect to the 3D camera and acquire both color image and depth image. The application would associate with Kinect V1, and both of the two types of image would be directly delivered to the viewer. It made us easily capture the image for the experiments. The captured depth image would be served for 3D algorithms. Meanwhile, the captured color images, which have the same 640 × 480 resolution, were used as the dataset of Mask R-CNN. However, before using the dataset of Mask R-CNN, the color image had to surpass an essential step—the calibration step. This essential step not only calibrates the color images as 2D image processing, but also maps the different coordinates of the color image and depth image. Two-dimensional cameras, IR Emitter and IR Depth Sensor, are arranged differently in Kinect V1. Moreover, the 2D and 3D cameras do not have the same resolution. In particular, the 2D camera has a 640 × 480 resolution, and the 3D camera has a 320 × 240 resolution. Thus, this calibration and mapping coordinate step is the very first step in combining 2D algorithms and 3D algorithms.

We equipped all the experimental instruments with 4 classes of object. Using the Keras ImageDataGenerator library, we generate new images by horizontal and vertical flipping, rotating (±2 degree), shifting height and width, changing brightness. Both original and generated images are used together to form a dataset. The dataset has 1000 color images, in total, of four classes: blue box, cup, cylinder, and white box. We trained this dataset with COCO [27] pre-trained, which is trained with 80 classes. For training, we choose some arguments for configuration such as the number of epochs is 30 with 100 steps each, learning rate equals is 0.001. We used a mini-batch size of 1 image per GPU and trained the model for 11k iterations. We used a weight decay of 0.0001, a learning momentum of 0.9 and a Gradient norm clipping rate of 5.0. The training process took approximately four hours under a GTX 1650 GPU with 12GB of VRAM and this setting. The training process is illustrated as shown in Figure 6. We implemented experiments on both backbone Resnet 50 and Resnet 101 to consider what network will be effective in combining with 3D algorithms.

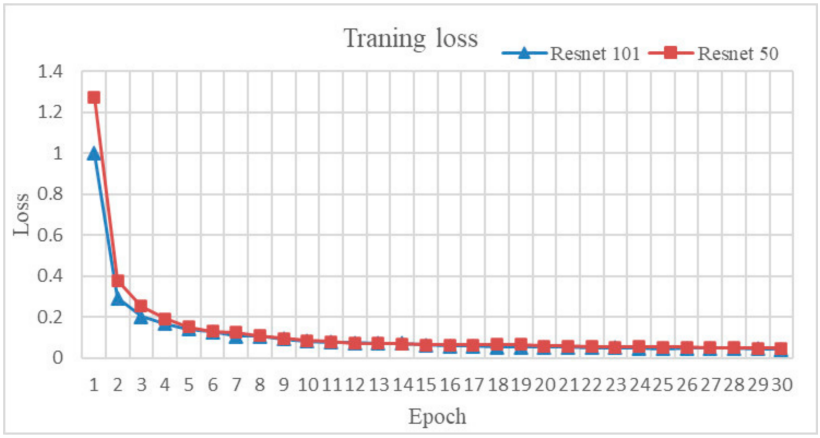
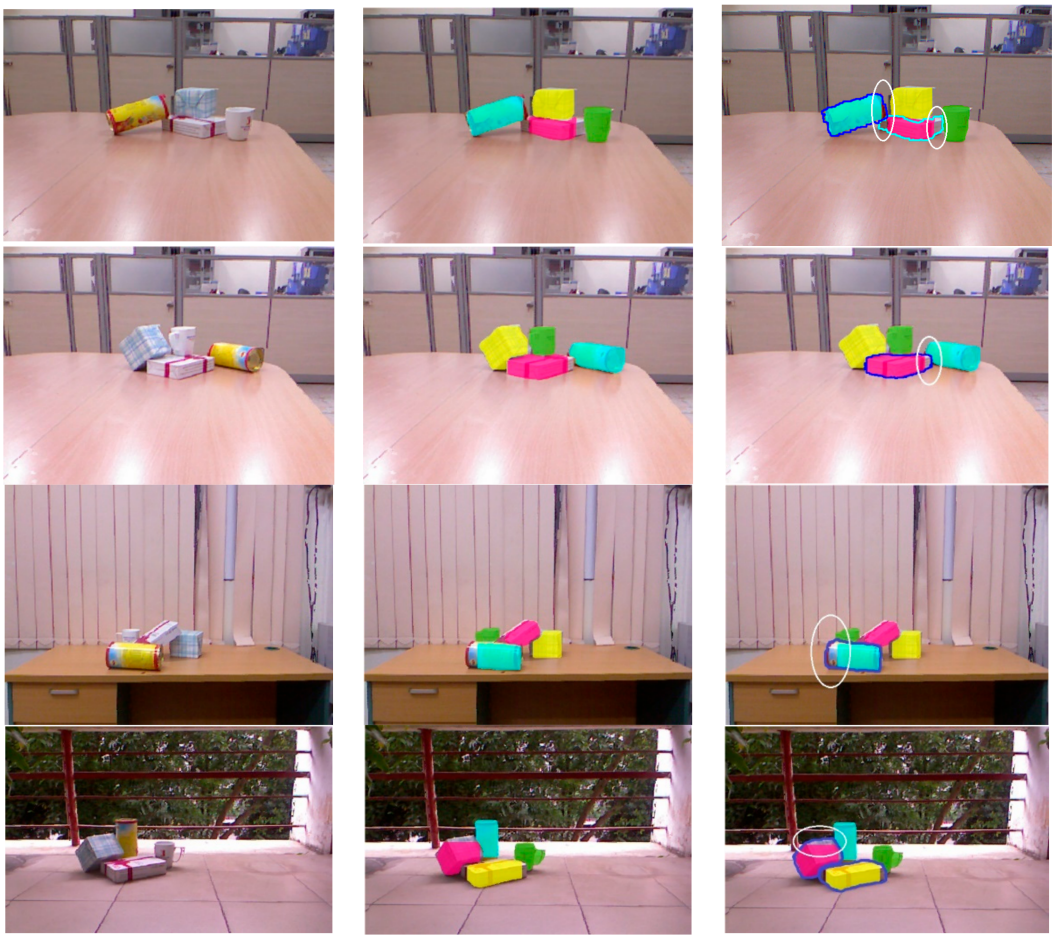


Figure 6. Training loss original network with Resnet 101 and Resnet 50.

4.2. Result and Evaluation

The following Figure 7 illustrates the comparison between the results of the original Mask R-CNN and after applying our approach. Our goal could be observed that it reaches segmentation accuracy better than the original framework. The object mask generated by

Mask R-CNN could not deliver enough information about the object while our approach covers most of the entire object even though the objects were obscurely located. The border regions or even the regions that have less brightness are usually challenging for 2D algorithms to approach, but all of them are kept in our research as below. The increased accuracy of covering all regions of the object will submit a prosperous pose estimation of objects to determine the grasping position and grasping orientation for the robot arm.



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

Figure 7. The comparison image between (**a**) original images, (**b**) Mask R-CNN and (**c**) our approach. Our approach can find all the borders of the object, as shown in the white circle.

As shown in Figure 7, our approach could both cover and segment the edge region. The result shows that it classified the edge regions and added these regions to the corresponding mask generated by Mask R-CNN. All regions of the objects are obtained completely and then can approve the grasping plan. The poor masks usually happen when the objects are located close together. Thus, to prove the performance of our approach, we focus on 50 experiments with overlapping located objects as shown in Figure 7a. Figure 7b shows that the original masks generated by Mask R-CNN are poor; this means these masks could not contain all of the information of the objects. This overlooked information is separated after the subtraction step as shown in Figure 7c. As described above, for segmenting this information, we continuously place each original mask to the subtraction result and applied Euclidean Cluster Extraction. Despite the fact that the size of the dataset is modest and the limitation mentioned before, our approach enables the hardest region to segment as shown in Figure 7c.

We evaluate the success of the method by average precision (AP) and IoU (Intersection over Union) [28] indicator. The AP is evaluated at an IoU threshold of 0.5 to 0.95 with

an interval of 0.05. The final mAP (mean Average Precision) result is the average of the 10 measurements.

$$\text{IoU} = \frac{\text{area of overlap}}{\text{area of union}} \tag{2}$$

True positive, false positive, true negative, and false negative are abbreviated as TP, FP, TN and FN, respectively, accuracy [28] is:

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

Our method obtained not only higher IoU but also higher mAP than the original Mask RCNN result. To verify the effectiveness of our method, we practice throughout both backbone Resnet 101 and Resnet 50. This experiment will ensure that the improvements come from the difference of backbone Resnet or from our approach. The comparison is presented as Tables 1 and 2, which compare the IoU and mAP between each object cup, blue box, cylinder and white box through Mask R-CNN with backbone Resnet 101, Mask R-CNN with backbone Resnet 50, our approach applying for Mask R-CNN Resnet 101 and our approach applying for Mask R-CNN Resnet 50. When using only 3D traditional segmentation algorithms, it evidently achieves a low level of IoU. The combination improved considerably compared to initial 3D segmentation algorithms.

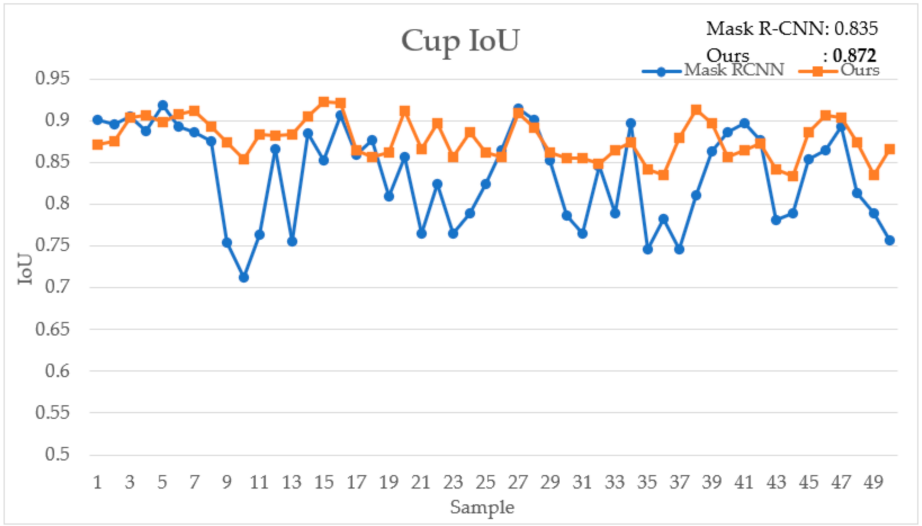
Table 1. IoU of our approach in comparison to original Mask R-CNN and 3D traditional segmentation algorithms.

Model	Cup IoU	Blue Box IoU	Cylinder IoU	White Box IoU
Mask RCNN (Resnet 101)	0.835	0.772	0.825	0.812
Mask RCNN (Resnet 50)	0.821	0.768	0.826	0.792
DoN	0.264	0.209	0.314	0.195
Euclidean Cluster Extraction	0.310	0.287	0.257	0.358
Ours (Resnet 101)	0.872	0.837	0.865	0.881
Ours (Resnet 50)	0.870	0.834	0.859	0.879

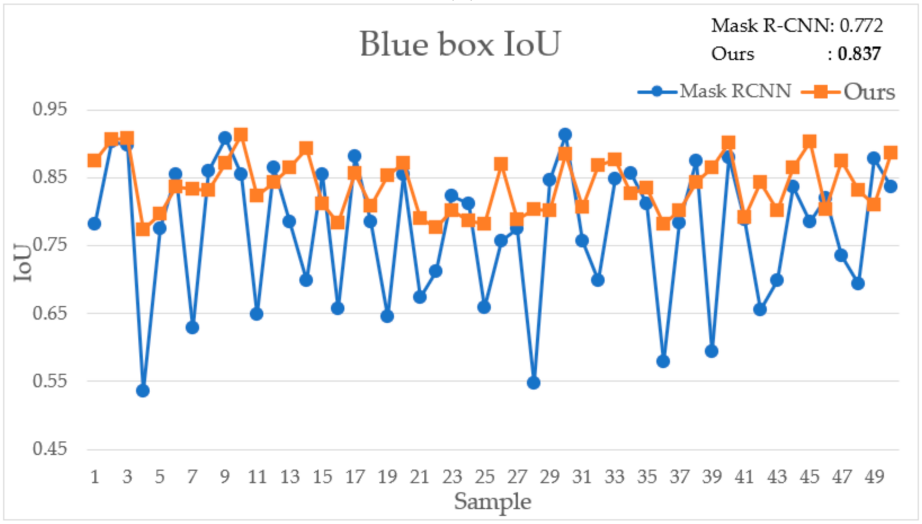
Table 2. Mean average precision of our approach in comparison to original Mask R-CNN.

Model	mAP	mAP_{cup}	$\text{mAP}_{\text{blue box}}$	$\text{mAP}_{\text{cylinder}}$	$\text{mAP}_{\text{white box}}$
Mask RCNN (Resnet 101)	0.39	0.399	0.336	0.465	0.365
Mask RCNN (Resnet 50)	0.38	0.387	0.345	0.448	0.365
Ours (Resnet 101)	0.46	0.446	0.429	0.524	0.437
Ours (Resnet 50)	0.46	0.456	0.421	0.534	0.447

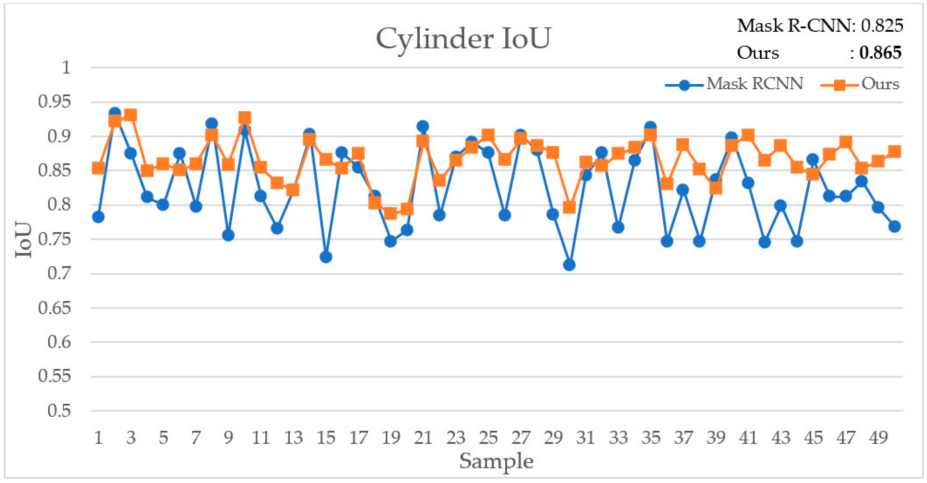
To detail our exercise, Figure 8 illustrates the IoU of all four types of objects through Mask R-CNN and our proposed approach. The difference of the backbone network does not make a change in the result. The original network is robust in the object segmentation task but testing with only the state of the overlapping objects shows its weakness. In general, Mask R-CNN grants an unsteady result. Sometimes the IoU of each object exceeds 0.9. It happens four times with Cup, four times with Cylinder, only two times with White box and zero times with Blue box. It generally achieves low and strong volatility and is unstable. The Cup IoU grants under 0.8 many times and even reaches 0.7. The IoU indicator of the Blue box usually is under 0.7 and sometimes under 0.6. The IoU result of the Cylinder and White box IoU are more moderate fluctuations. The IoU is mostly higher than 0.7, but the movement of each two continuous samples is a huge number. All the fluctuations are caused by the overlapping, obscurity, closed location of objects.



(**a**)

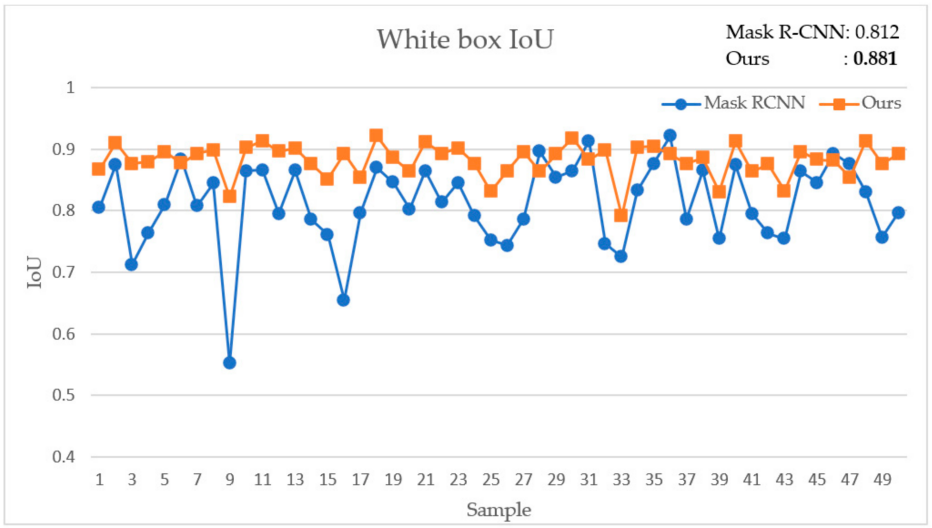


(**b**)



(**c**)

Figure 8. *Cont.*



(**d**)

Figure 8. The comparison between Mask R-CNN and our proposed method through four objects. (**a**) Cup IoU, (**b**) Blue box IoU, (**c**) Cylinder IoU, (**d**) White box IoU.

Meanwhile, after using our method for doing experiments, the results have a slight fluctuation, greater IoU and are steadier. The IoU of the Blue box is now higher and around 0.8. In all likelihood, the IoU of the Cylinder, Cup and White box are greater or approximate to 0.85. In detail, our exceptional result in obtaining higher IoU is the case for the majority. At some points, our score is less accurate than the original network, but it is close to IoU of the original network result. With regard to this issue, it is driven up by the scabrous of object contours, which come from the downsampling step in point cloud processing. As opposed to this case, when our approach achieves greater results, it reaches a more significantly advanced IoU than the original Mask R-CNN. Our proposed approach upgrades the ability of Mask R-CNN to detect and segment the edge areas more accurately and with more stability, even in the difficult cases.

The more stable IoU indicator is the premise leading to the greater mAP. Table 2 denotes the effectiveness of our approach in effort to improve the accuracy of segmentation. Although all practices are implemented with severe cases to emphasize the ability of dealing with edge region. The backbone Resnet 101 and Resnet 50 does not make too much of a difference in the result throughout both the original model and our method. Our approach achieves has the exceptional consequence of improving the performance of the original Mask R-CNN. In addition, the mean average precision increased from 0.39 to 0.46.

5. Discussion

According to the experiments, the combination between Mask R-CNN and 3D traditional segmentation algorithms advances the determination of the edge and border regions. In other words, the proposed method undoubtedly improves the performance in the instance details. Typical 2D architectures do not effectively work in border segmentation without an enormous dataset. However, even though having a good preparation of the dataset, the accuracy of segmentation could not reach an excellent result. As it depends on the contrast, the lightness of the image, the location of the objects, and especially overlapped located objects had an impact. Notably, the heavy point cloud networks can overcome these issues; however, they are hard to implement because of the complex and not regular input data format. The addition of the 3D process branch brings away the disadvantages of two styles of approach. Without attentively preparing the experiment setup for the 2D approach or complex input data for the 3D approach, this model develops the original network to be more sensitive to the instance detail. The performance is remarkably improved compared to the state-of-the-art Mask R-CNN.

Tables and charts provide concrete evidence for the outperformance. The IoU indicators, which directly reflect the proportion of the predicted mask on the truth mask, are precarious. this is because some objects are easily obvious, and others are obscure in harsh testing conditions. After applying the 3D process branch, the IoU becomes more stable in higher levels leading to exceptional improvement of accuracy segmentation indicator mAP in severe circumstances. In other words, the typical limitation of the 2D segmentation model pipeline does not harm the performance of our method. These things obviously denote not only the important role of spatial information but also the effectiveness of the proposed combining.

Practicing in the circumstance equipped by the hardware mentioned in the experiment preparation, it takes around 1 s for the original Mask R-CNN inferring an image by GTX 1650 GPU. Based on the result of Mask R-CNN, our method needs the original masks as the reference for classifying the edge region. In other words, we necessarily operate the original model before accomplishing the outcome. This means our research spends more time processing than the original model. In particular, we need around 3 s to finish the processing. An amount of 2 s is operated for 3D algorithms. Compared to the original Mask R-CNN result, it takes only 195 ms per image with model equipment [7]. Therefore, the inferencing time can be significantly decreased by the upgrading of hardware. Although our performance is hard to reach real-time requirement, but it is potential in applying for robot systems. Vision processing does not need to reach real-time requirements for a robot operating continuously, vision tasks can be taken while the robot is inspecting, grasping, or moving objects. With respect to robot manipulation, the recent state-of-the art research [29,30] shows that our method can satisfy their requirements.

6. Conclusions

In this paper, we have proposed the combination of Mask R-CNN and 3D segmentation algorithms as Difference of Normals-based segmentation and Euclidean Cluster Extraction to effectively segment the objects in a grasping environment. By adding the 3D process branch, in addition to predicting the mask, this model can safeguard objects in full and distinguish, classifying the edge region. Based on the spatial relationship between pixels in the edge and central regions, the complete mask is reached by applying the cluster algorithm. Hence, the mask prediction covers all the objects without considering the severity of the environment or object overlapping. Our solution is easy to implement and performs well; in addition, the IoU indicator is increased and more stable. This leads to a significant rise from 0.39 to 0.46 in the mAP indicator. Our approach performs remarkably well in specifying the contours of the objects. However, the contours are still not smooth, which comes from the downsampling step in the point cloud process, which has a bad effect on the localization accuracy. We will apply an upsampling process to our method to improve the localization accuracy in future work.

Author Contributions: Supervision: H.H.H.; conceptualization: B.L.T.; methodology: H.H.H. and B.L.T.; validation: H.H.H. and B.L.T.; investigation: B.L.T.; resources: B.L.T. and H.H.H.; writing—original draft preparation: B.L.T. and H.H.H.; writing—review and editing: H.H.H. and B.L.T.; project administration: H.H.H.; funding acquisition: H.H.H. All authors have read and agreed to the published version of the manuscript.

Funding: This work was funded by Vietnam Ministry of Education and Training under project number B2020-BKA-02.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: MDPI Research Data Policies.

Conflicts of Interest: The authors declare that there is no conflict of interest.

References

1. Brachmann, E.; Krull, A.; Michel, F.; Gumhold, S.; Shotton, J.; Rother, C. Learning 6D object pose estimation using 3D object coordinates. In *European Conference on Computer Vision (ECCV)*; Springer: Cham, Switzerland, 2014; pp. 536–551.
2. Hu, Y.; Hugonot, J.; Fua, P.; Salzmann, M. Segmentation-Driven 6D object pose estimation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019; pp. 3385–3394.
3. Deng, X.; Xiang, Y.; Mousavian, A.; Eppner, C.; Bretl, T.; Fox, D. Self-supervised 6D object pose estimation for robot manipulation. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020.
4. Girshick, R.; Donahue, J.; Darrell, T.; Malik, J. Region-based convolutional networks for accurate object detection. *IEEE Trans. Pattern Anal. Mach. Intell.* **2015**, *38*, 142–158. [CrossRef] [PubMed]
5. Girshick, R. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015.
6. Ren, S.; He, K.; Girshick, R.; Sun, J. Faster R-CNN: Towards real-time object detection with region proposal networks. *arXiv* **2015**, arXiv:1506.01497. [CrossRef] [PubMed]
7. He, K.; Gkioxari, G.; Dollár, P.; Girshick, R. Mask rcnn. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2980–2988.
8. Huang, Z.; Huang, L.; Gong, Y.; Huang, C.; Wang, X. Mask Scoring R-CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 16–20 June 2019.
9. Zhang, Y.; Chu, J.; Leng, L.; Miao, J. Mask-Refined R-CNN: A Network for Refining Object Details in Instance Segmentation. *Sensors* **2020**, *20*, 1010. [CrossRef] [PubMed]
10. Ioannou, Y.; Taati, B.; Harrap, R.; Greenspan, M. Difference of Normals as a Multi-scale Operator in Unorganized Point Clouds. In Proceedings of the 2012 Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission, Zurich, Switzerland, 13–15 October 2012.
11. pcl.readthedocs.io. Available online: https://pcl.readthedocs.io/en/latest/cluster_extraction.html (accessed on 1 March 2021).
12. Ronneberger, O.; Fisher, P.; Brox, T. U-Net: Convolutional Networks for Biomedical Image Segmentation. In *Computer Vision and Pattern Recognition*; Springer: Cham, Switzerland, 2015.
13. Wu, X.; Wen, S.; Xie, Y. Improvement of Mask-RCNN Object Segmentation Algorithm. In *ICRIA 2019: Intelligent Robotics and Applications*; Springer: Cham, Switzerland, 2019.
14. Rother, C.; Kolmogorov, V.; Blake, A. GrabCut: Interactive Foreground Extraction Using Iterated Graph Cuts. *ACM Trans. Graph.* **2004**, *23*, 309–314. [CrossRef]
15. Xu, C.; Wang, G.; Yan, S.; Yu, J.; Zhang, B.; Dai, S.; Li, Y.; Xu, L. Fast Vehicle and Pedestrian Detection Using Improved Mask R-CNN. *Math. Probl. Eng.* **2020**, *2020*, 5761414. [CrossRef]
16. Lin, T.Y.; Dollár, P.; Girshick, R.; He, K.; Belongie, S. Feature pyramid networks for object detection. In Proceedings of the CVPR, IEEE, Honolulu, HI, USA, 21–26 July 2017.
17. Liu, S.; Qi, L.; Qin, H.; Jia, J.S.J. Path Aggregation Network for Instance Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, SU, USA, 18–22 June 2018.
18. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017.
19. Uijlings, J.R.R.; van de Sande, K.E.A.; Gevers, T.; Smeulders, A.W.M. Selective Search for Object Recognition. *Int. J. Comput. Vis.* **2012**, *104*, 154–171. [CrossRef]
20. Albawi, S.; Mohammed, T.A.; I-Zawi, S.A. Understanding of a convolutional neural network. In Proceedings of the International Conference on Engineering and Technology (ICET), Antalya, Turkey, 21–23 August 2017.
21. Redmon, J.; Angelova, A. Real-time grasp detection using convolutional neural networks. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015.
22. Rao, D.; Le, Q.V.; Phoka, T.; Quigley, M.; Sudsang, A.; Ng, A.Y. Grasping novel objects with depth segmentation. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), Taipei, Taiwan, 18–22 October 2020.
23. Uckermann, A.; Elbrechter, C.; Haschke, R.; Ritter, H. 3D scene segmentation for autonomous robot grasping. In Proceedings of the IEEE International Conference on Intelligent Robots and Systems (IROS), Vilamoura-Algarve, Portugal, 7–12 October 2012.
24. Sarbolandi, H.; Lefloch, D.; Kolb, A. Kinect Range Sensing: Structured-Light versus Time-of-Flight Kinect. *Comput. Vis. Image Underst.* **2015**, *139*, 1–20. [CrossRef]
25. Kurban, R.; Skuka, F.; Bozpolat, H. Plane Segmentation of Kinect Point Clouds using RANSAC. In Proceedings of the 2015 7th International Conference on Information Technology, ICIT, Huangshan, China, 13–15 November 2015.
26. pcl.readthedocs.io. Available online: https://pcl.readthedocs.io/projects/tutorials/en/latest/don_segmentation.html (accessed on 1 March 2021).
27. Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft COCO: Common objects, in context. In *ECCV*; Springer: Cham, Switzerland, 2014.

28. Rahman, M.A.; Wang, Y. Optimizing intersection-over-union in deep neural networks for image segmentation. In *International Symposium on Visual Computing*; Springer: Cham, Switzerland, 2016; pp. 234–244.
29. Lundell, J.; Verdoja, F.; Kyrki, V. Beyond Top-Grasps through Scene Completion. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 31 May–31 August 2020; pp. 545–551.
30. Gualtieri, M.; Pas, A.t.; Saenko, K.; Platt, R. High precision grasp pose detection in dense clutter. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016.