agriculture

MDPI

Article

A Point Cloud Segmentation Method for Pigs from Complex Point Cloud Environments Based on the Improved PointNet++

Kaixuan Chang [1], Weihong Ma [2,3,4,*], Xingmei Xu [1], Xiangyu Qi [3], Xianglong Xue [2,3,4], Zhankang Xu [2,5], Mingyu Li [2,3,4], Yuhang Guo [2,3,4], Rui Meng [2,3,4] and Qifeng Li [1,2,3,4,*]

[1] College of Information and Technology, Jilin Agricultural University, Changchun 130118, China; a91616@163.com (K.C.); xingmeix@jlau.edu.cn (X.X.)
[2] Information Technology Research Center, Beijing Academy of Agriculture and Forestry Sciences, Beijing 100097, China; xuexl@nercita.org.cn (X.X.); xuzhankang@nwafu.edu.cn (Z.X.); limy@nercita.org.cn (M.L.); guoyh@nercita.org.cn (Y.G.); mengr@nercita.org.cn (R.M.)
[3] National Innovation Center of Digital Technology in Animal Husbandry, Beijing 100097, China; qxy@cau.edu.cn
[4] National Innovation Center for Digital Seed Industry, Beijing 100097, China
[5] College of Information and Electrical Engineering, China Agricultural University, Beijing 100083, China
* Correspondence: mawh@nercita.org.cn (W.M.); liqf@nercita.org.cn (Q.L.)

Abstract: In animal husbandry applications, segmenting live pigs in complex farming environments faces many challenges, such as when pigs lick railings and defecate within the acquisition environment. The pig's behavior makes point cloud segmentation more complex because dynamic animal behaviors and environmental changes must be considered. This further requires point cloud segmentation algorithms to improve the feature capture capability. In order to tackle the challenges associated with accurately segmenting point cloud data collected in complex real-world scenarios, such as pig occlusion and posture changes, this study utilizes PointNet++. The SoftPool pooling method is employed to implement a PointNet++ model that can achieve accurate point cloud segmentation for live pigs in complex environments. Firstly, the PointNet++ model is modified to make it more suitable for pigs by adjusting its parameters related to feature extraction and sensory fields. Then, the model's ability to capture the details of point cloud features is further improved by using SoftPool as the point cloud feature pooling method. Finally, registration, filtering, and extraction are used to preprocess the point clouds before integrating them into a dataset for manual annotation. The improved PointNet++ model's segmentation ability was validated and redefined with the pig point cloud dataset. Through experiments, it was shown that the improved model has better learning ability across 529 pig point cloud data sets. The optimal mean Intersection over Union (mIoU) was recorded at 96.52% and the accuracy at 98.33%. This study has achieved the automatic segmentation of highly overlapping pigs and pen point clouds. This advancement enables future animal husbandry applications, such as estimating body weight and size based on 3D point clouds.

Keywords: point cloud segmentation; PointNet++; 3D point cloud processing; SoftPool

check for updates

Citation: Chang, K.; Ma, W.; Xu, X.; Qi, X.; Xue, X.; Xu, Z.; Li, M.; Guo, Y.; Meng, R.; Li, Q. A Point Cloud Segmentation Method for Pigs from Complex Point Cloud Environments Based on the Improved PointNet++. *Agriculture* **2024**, *14*, 720. https://doi.org/10.3390/agriculture14050720

Academic Editors: Ali Alameer, Taha Mansouri and Francesco Marinello

Received: 27 February 2024
Revised: 6 April 2024
Accepted: 29 April 2024
Published: 2 May 2024

1. Introduction

With the rapid development of computer vision and artificial intelligence, 3D scene analysis has emerged as a critical research area across various disciplines. Among them, point cloud data, as a representation that can accurately capture 3D geometric information, has received widespread attention. Point cloud data consists of many discrete points, each containing essential information, such as the position, shape, and surface features of an object, so point cloud segmentation has been widely used in 3D computer vision scenarios [1]. For example, good results have been achieved in urban planning and design, medical data processing, architectural modeling, precision flight of unmanned aircraft, and autonomous driving [2–7]. In intelligent agriculture, a depth camera is used to collect the point clouds of plants. Then, the collected data are sent to neural networks to segment the leaves and stems of plants, which helps in selective breeding and trait analysis [8]. A secondary segmentation method for point clouds based on 2D images and depth information was proposed to extract fruits' contours in complex backgrounds [9]. In animal husbandry, an efficient and automated method was developed for measuring beef cattle body dimensions, facilitating the collection of reliable phenotypic data throughout the fattening process and directly integrating it into the breeding strategy [10]. A 3D surface reconstruction and pig size measurement system based on a multi-view RGB-D camera was developed to measure pig dimensions [11]. In these applications, accurate point cloud segmentation is one of the critical steps towards object recognition, environment perception, and scene understanding.

With the rapid development of modern animal husbandry, precise and intelligent farm management has become a critical factor in improving production efficiency, ensuring animal welfare, and achieving sustainable development. In this context, 3D point cloud data, within a data type that can provide spatial geometric information, have shown great potential for application in animal husbandry [12]. Point cloud data can effectively capture the 3D structure of pigs and their surroundings, thus providing necessary information support for behavior analysis, health monitoring, and environmental optimization, particularly in the pig farming environment.

However, in livestock applications, traditional point cloud segmentation techniques still have many limitations when dealing with the point cloud data of pigs in complex environments. Dense and disorganized point cloud data often complicate effective feature extraction and segmentation. In addition, the irregularity and diversity of pig movements further increase the difficulty of point cloud segmentation, especially in improving computational efficiency while maintaining segmentation accuracy [13]. For example, in the process of pig point cloud acquisition, it is necessary to use the railing to control the pigs within the acquisition environment; however, because pigs are unfamiliar with the environment during the acquisition process, partial occlusion and gesture changes will occur and, also, the background of the farm will affect the quality of the collected point cloud data, e.g., the pigs will lick the railing, defecate within the acquisition environment, and so on. The complexity makes it difficult for traditional rule-based and feature-engineering methods to achieve satisfactory results. Therefore, there is an urgent need for a point cloud segmentation method with more powerful feature extraction and representation capabilities.

To address these challenges, deep learning techniques have made significant breakthroughs in point cloud segmentation [14–17]. Early deep learning methods viewed point clouds as unordered sets of points by learning features for each point, e.g., PointNet. Meanwhile, PointNet++ proposes layer-by-layer feature extraction and aggregation on this basis, which can better capture local and global geometric information. In addition, convolutional neural network (CNN)-based approaches have also achieved some success, such as using 3D convolution for processing point clouds [18–20]. In particular, PointNet++, an advanced deep learning architecture, can better cope with segmentation tasks in complex environments by extracting local and global features from point cloud data. Leveraging PointNet++, this study aims to refine the accuracy of pig point cloud segmentation in practical settings, like pig farms, thereby offering advanced management tools for the animal husbandry sector.

This study aims to leverage the advantages of PointNet++ and address the challenges posed by complex environments, exploring methods to segment live pig point clouds based on such complexities while enhancing the accuracy and robustness of segmentation. We adjusted the parameters and structure of the traditional PointNet++ model by utilizing point cloud data collected in real-world scenarios. Additionally, we introduced the SoftPool pooling method as a replacement for the original max pooling for down-sampling. This approach led to the development of an improved PointNet++ model tailored for the semantic segmentation of live pig point clouds in complex environments. The model achieved finer segmentation of railings and pig bodies. Through experiments, we validated the effectiveness of this method in addressing complexities, such as occlusion and posture changes, thereby achieving more accurate segmentation of live pig point clouds. Compared to previous methods, the SoftPool pooling technique allows for smoother feature extraction, which helps in retaining more local details. This is especially important for processing complex point cloud data. The improvement in feature extraction enables the model to capture the nuances of the pig and its surroundings more accurately, leading to better segmentation accuracy. Furthermore, the SoftPool pooling method aggregates information by weighted averaging, which makes the improved PointNet++ network better suited for handling irregularly distributed and noisy point cloud data. In farming environments, point cloud data can be highly irregular due to factors such as animal activity and equipment errors. Therefore, the SoftPool pooling method can help the model better cope with these challenges. This provides a powerful tool for processing point cloud data in precision agriculture and lays a strong foundation for future research and technology development in this field.

2. Materials and Methods

The technical pipeline in this study is illustrated in Figure 1. When pigs enter the collection area, the infrared grating is triggered to issue a synchronized acquisition command. Five Kinect DK cameras positioned at different angles (left, upper-left, right, upper-right, and overhead) simultaneously capture the local point clouds of the pigs. Subsequently, the acquisition devices transmit these point cloud data to the control computer. The computer's three-dimensional point clouds from different perspectives are aligned through contour-based analysis in respective coordinate systems. Point cloud data from other cameras are registered globally. Live pig point cloud data preprocessing is optimized through pass-through filtering, background removal, and point cloud labeling. Following this, the improved PointNet++ neural network model is utilized to segment the processed live pig point cloud data and the model is trained accordingly. Finally, the visualization of the segmentation results is generated.

2.1. Point Cloud Acquisition of the Pig Body

In this study, the Open NI (Natural Interaction) Library and Point Cloud Library (PCL) were installed on the Visual Studio 2022 platform running on the Windows 11 operating system. The C++ language was used to register, filter, and segment pig point clouds. The experimental equipment used Kinect DK as an RGB-D depth camera device, which has a depth image resolution of 512 × 512 pixels, a horizontal viewing angle of 120°, and a vertical viewing angle of 120°; it supports a maximum of 30 frames/s for depth data acquisition and the viewing distance is between 0.25 and 2.88 m.

The data were collected at Tieluqishi Guanghui Breeding Farm, Guanghui County, Mianyang City, Sichuan Province, China. Three-dimensional point cloud data of live pigs were acquired using Kinect DK cameras. Pigs aged between 60 and 120 days were selected for this study. Due to the young age of the pigs (30-day-old piglets), compactly collecting data could lead to issues such as high data repeatability. Hence, a multi-batch phased collection scheme was adopted, with the pig point cloud collection plan spanning six cycles, each lasting five days, totaling one month. This collection scheme allows space for the growth of the piglets, with a weight gain of approximately 50 pounds expected over 30 days. Multiple batches of collection ensure a sufficient volume of data.

Kinect DK cameras were set up at each of the five positions A, B, C, D, and E of the acquisition channel in the test site. To avoid incomplete data collection, only one pig was allowed to enter the acquisition channel at a time. When the target pig entered the data collection area, the infrared grating was triggered and the trigger signal reached the control box instantly, driving the RFID reader to read the ear tag number of the pig. Point clouds were simultaneously captured from five directions when the valid ear tag was read. After collecting a certain number of times per pig, the pig was herded back to the pen and the start and end times of the collection were recorded corresponding to the ear tag to prevent double collection of the same pig. Finally, pigs that had not been collected were driven into the collection environment. Upon entering the collection area, the collected pig point clouds varied significantly due to individual differences and different postures. Figure 2 shows the CAD framework of the pig three-dimensional point cloud collection device. The data collection and verification scenes are depicted in Figure 3.

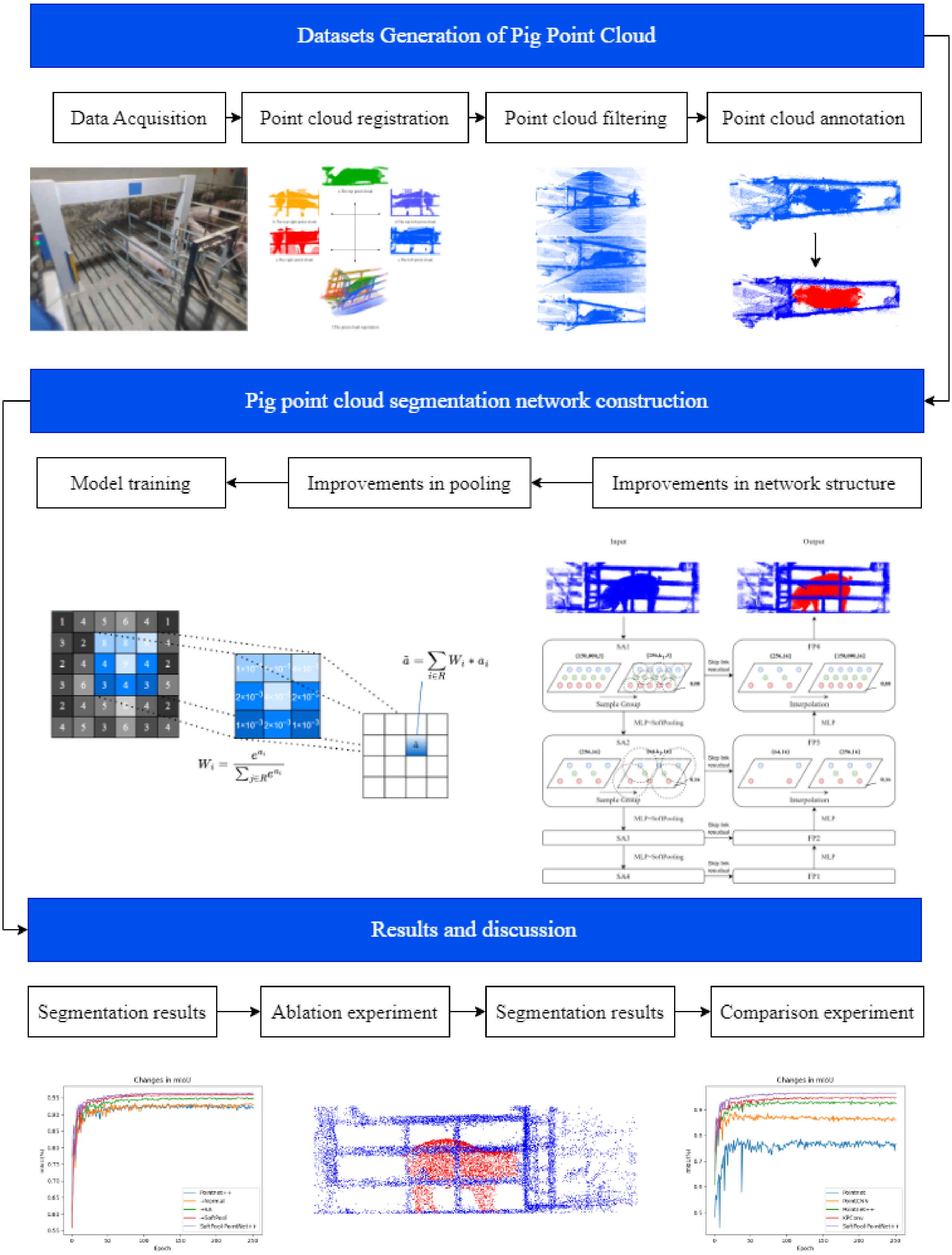


Figure 1. Flowchart of point cloud segmentation method for pigs in complex environments.

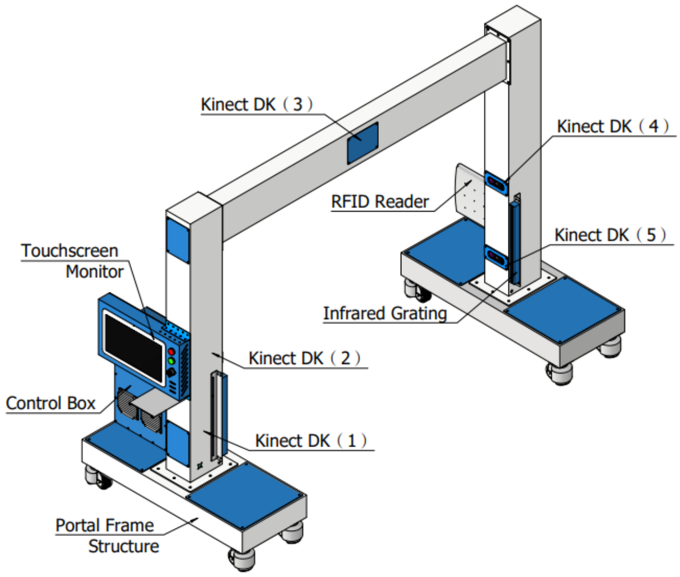


Figure 2. CAD framework diagram of pig 3D point cloud acquisition device.

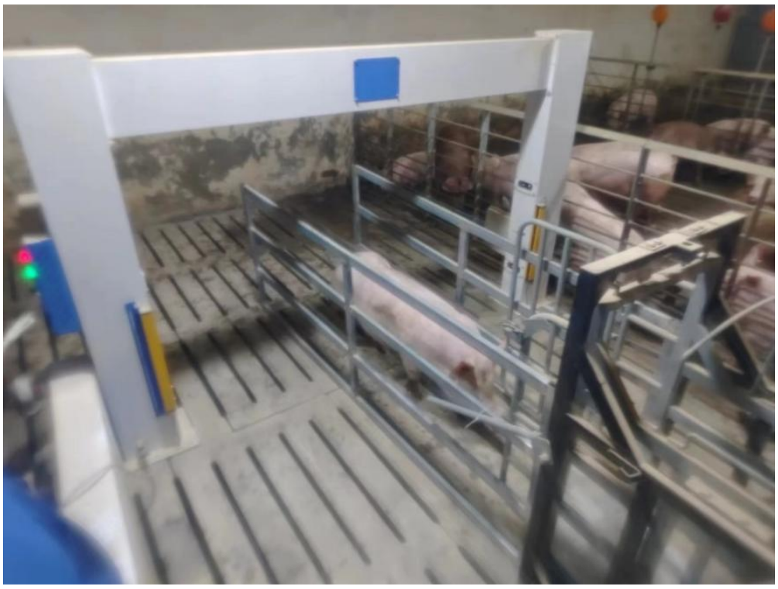


Figure 3. Pig 3D point cloud field acquisition device and acquisition environment.

2.2. Multi-Angle Pig Point Cloud Registration

Point cloud registration involves converting multiple local point clouds, acquired from various depths and viewpoints, into a unified world coordinate system. The 3D point cloud data in different viewpoint coordinate systems are converted to the same coordinate system through coordinate transformation to generate a global point cloud [21]. Point cloud registration is divided into two steps: rough registration and fine registration. Rough registration, i.e., the initial registration of the point cloud, provides an initial value of the rotation and translation matrix for fine matching so that two point clouds with different positions can be registered as much as possible. Fine registration refers to the further computation of the approximate rotational translation matrix of the two-point clouds based on the initial registration. During the rough registration process, it becomes challenging to obtain the initial estimates used for the fine registration in the absence of apparent markers, point cloud overlaps, or geometric features. Since the railings are rectangular, in this paper, the railings are used as the registration standard. Then, the matrices of the different point clouds are computationally transformed to the same coordinate system by rotational translation. The railing's point cloud is acquired from multiple viewpoints without the pig, allowing for the extraction of geometric feature points to compute the railing's rotation and translation matrix. Finally, the rotational translation matrix of the railing is utilized to register the pig point cloud. The 3D reconstructed pig point cloud map obtained after the registration fusion is shown in Figure 4f.

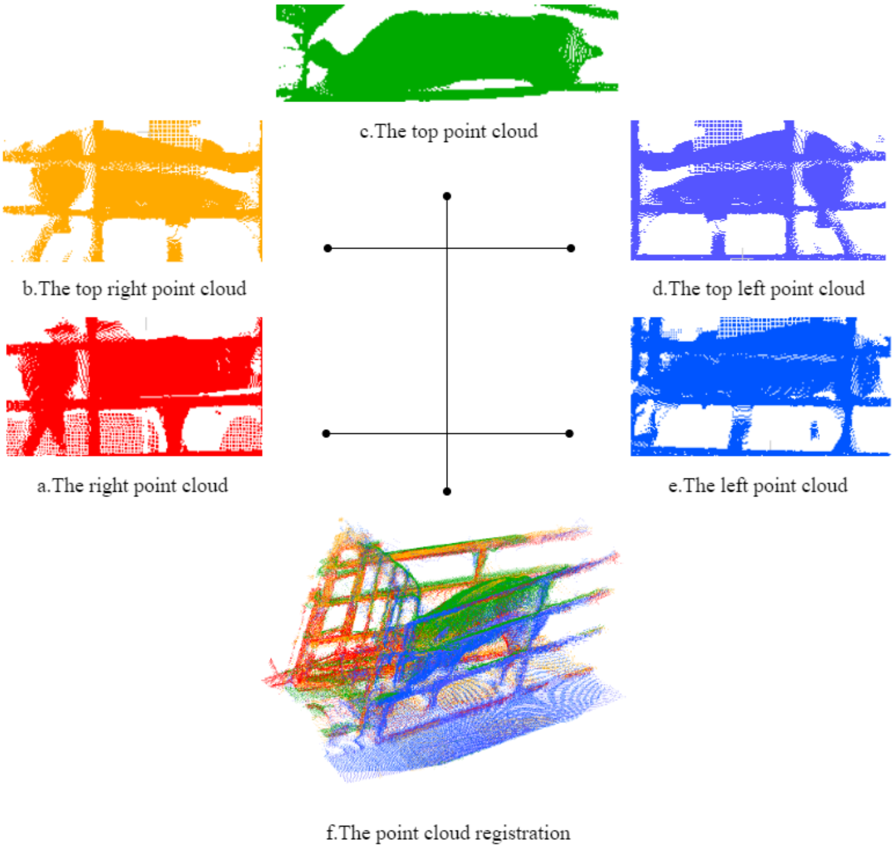

c.The top point cloud
b.The top right point cloud
d.The top left point cloud
a.The right point cloud
e.The left point cloud
f.The point cloud registration

Figure 4. Multi-view pig point cloud registration fusion map.

We define the point cloud captured by camera A as P (x, y, z) and the point clouds captured by cameras B, C, D, and E as P' (x', y', z') and the relationship between point P and point P' is denoted as (1):

$$P\prime = R \times P + T = RxRyRz \times \begin{bmatrix} x \\ y \\ z \end{bmatrix} + \begin{bmatrix} tx \\ ty \\ tz \end{bmatrix} \tag{1}$$

where R is the rotation matrix, T is the translation matrix, R_x is the rotation matrix of the x-axis, R_y is the rotation matrix of the y-axis, R_z is the rotation matrix of the z-axis, t_x is the translation distance of the x-axis, t_y is the translation distance of the y-axis, and t_z is the translation distance of the z-axis. The rotation translation matrix obtained by using the railings as a registration criterion has strong robustness and high accuracy.

2.3. Target Point Cloud Extraction

After registration, the original point cloud data of the pig point cloud are too large, with many irrelevant point cloud data, so the non-essential point cloud needs to be deleted. First, the original point cloud is filtered using pass-through filtering to obtain the point cloud of the best observation area; the remaining point cloud mainly consists of the target pig to be measured, the railing, and the ground, in addition to some noise points. Here, the Random Sample Consensus (RANSAC) detects the ground point cloud [22].

The RANSAC algorithm uses a fixed threshold parameter set according to how well different parameters remove the ground. The results show that the RANSAC algorithm effectively removes the ground point cloud and extracts the target point cloud. Figure 5 shows the comparison results of preprocessing before and after point cloud filtering; most of the noise is eliminated and the overall outline of the pig is clearly retained.

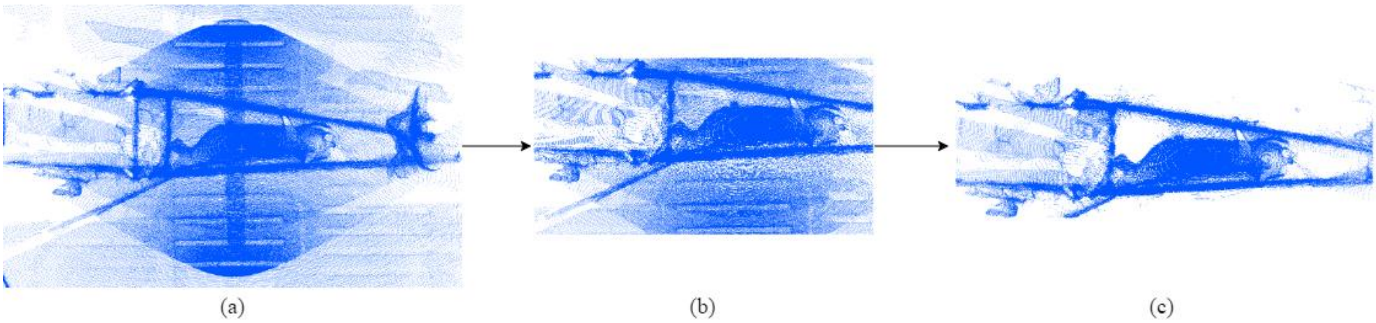


Figure 5. Comparison results of preprocessing before and after point cloud filtering. (**a**) Raw data; (**b**) pass-through filtering results; (**c**) RANSAC results.

2.4. Pig Point Cloud Segmentation Model Based on the Improved PointNet++

Due to pigs biting and licking the railing during data collection, manifested in the point cloud as the pig point cloud and the railing point cloud being connected, the RANSAC algorithm cannot remove the railing, resulting in extracted target point clouds containing pig bodies and railing. Given this complex situation, an improved PointNet++ model is employed in this study to segment both pig body and railing point clouds.

The improvements made to the classical PointNet++ model in this paper mainly include (1) adjustments to the model structure, such as the number of network layers, sampling radius, and feature extraction quantity, to make the model more suitable for live pig point cloud data and (2) in terms of pooling methods, SoftPool being utilized as the point cloud feature down-sampling method, replacing the original max pooling, to enhance the model's ability to extract point cloud features.

2.4.1. Classical PointNet++ Point Cloud Segmentation Models

PointNet++ introduces a recursive hierarchical structure, continuously subdividing local regions and aggregating features to achieve multi-scale feature extraction [23]. This enables PointNet++ to handle point cloud data better in complex scenarios, such as occlusion and pose variations. PointNet++ can extract rich local feature representations without losing global structural information through layer-by-layer feature transformation and pooling.

The network architecture of PointNet++ is illustrated in Figure 6. PointNet++ utilizes a hierarchical network structure consisting of set abstraction (SA) layers, sampling layers, grouping layers, and feature extraction layers, forming hierarchical point set feature learning to gradually extract both local and global features of the point cloud. This allows PointNet++ to capture better the hierarchical structure and local information of point clouds. The set abstraction layer of PointNet++ performs regional feature modeling on each subsampled point cloud, thus better capturing local information. This improves performance in tasks such as point cloud classification and segmentation. The sampling layer subsamples the input point cloud to reduce its complexity, effectively reducing computational load. The grouping layer is a critical operation used for local feature modeling. It divides the input point cloud into multiple groups for local feature extraction. The sampling layer utilizes Farthest Point Sampling (FPS) to select partial feature points of the point cloud data. The grouping layer selects neighboring points of the sampling center points using the Ball Query method, sending the chosen subpoint clouds to the feature extraction layer to encode information from the local regions into feature vectors. The feature extraction layer employs Multi-Layer Perceptrons (MLPs) to extract features from the local point sets computed by the grouping layer.

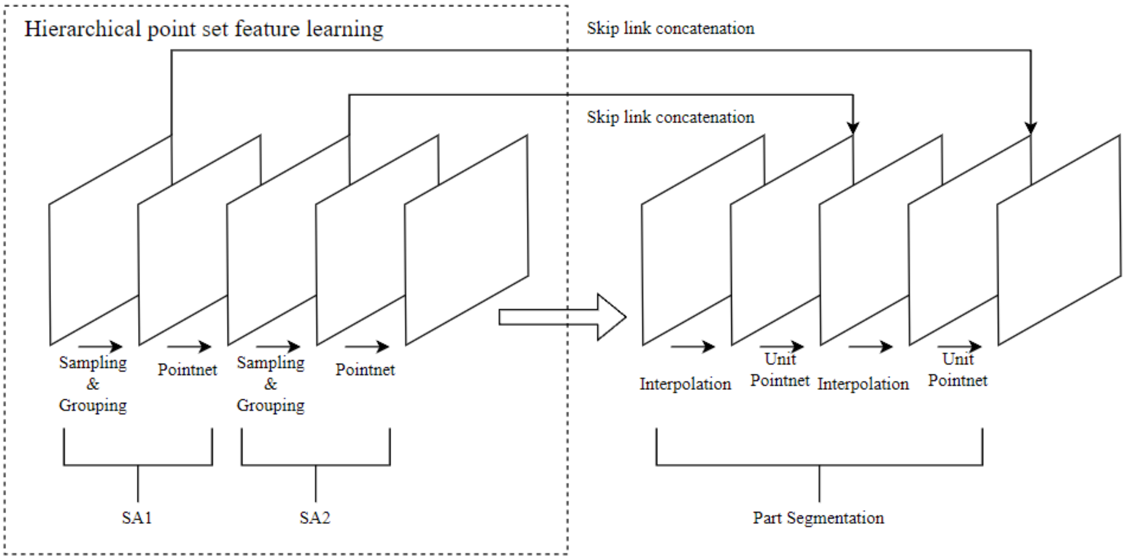


Figure 6. PointNet++ segmentation network structure.

For segmentation networks, PointNet++ adopts an Encoder–Decoder structure, where features are connected through skip link concatenation. Interpolation targets segmentation networks because segmentation results require assigning a semantic label to each point cloud. However, only the features of the down-sampled point cloud are obtained after the SA layer. The purpose of interpolation is to obtain the features of the points ignored during the down-sampling process. Unit PointNet consists of the MLP and Rectified Linear Unit (ReLU) and constitutes the subsequent MLP operation after the max pooling in the PointNet layer.

2.4.2. Improved Point Cloud Segmentation Model for Pigs with PointNet++ Improvements Based on the Set Abstraction Layer

To improve PointNet++ based on the SA layer, we have to understand the role of the SA layer, which is the critical component in PointNet++. It selects a certain number of key points by sampling and then defines a spherical region around these critical points. The points within this region are used as a grouping, to which PointNet is applied for feature extraction. In order to improve the model's ability to capture the local details of the pig body, the structure of multi-level feature extraction is further strengthened based on PointNet++. Repeatedly applying SA layers at different scales can capture features from local to global more efficiently, which is especially important for point cloud data processing in complex environments. The classical PointNet++ segmentation model is designed for the ShapeNet dataset, where the target size to be segmented is small [24]. The model uses three SA modules and three FP modules, where the second SA module has the largest sampling radius of 0.4 m. The last layer of the SA modules only performs MLP operations. This results in the classical PointNet++ segmentation model with a maximum sensory field of 0.4 m, which cannot satisfy the semantic segmentation task of the pig point cloud.

Therefore, the original network structure of the classical PointNet++ segmentation model is not directly applicable to the pig point cloud data. After analyzing the pig point cloud data, this study adjusts the model structure of the classical PointNet++ segmentation model; the adjusted model parameters are shown in Table 1.

Table 1. The layers, radius, and number of samplings of the proposed model.

Layer	PointNet++			Improved PointNet++		
	Npoint	Radius	Nsample	Npoint	Radius	Nsample
SA1	512	0.2	32	256	0.08	16
SA2	128	0.4	64	64	0.16	32
SA3	None	None	None	16	0.32	64
SA4	None	None	None	None	None	None
FP4	None	None	None	None	None	1280
FP3	None	None	1280	None	None	384
FP2	None	None	384	None	None	320
FP1	None	None	150	None	None	150

As can be seen from Table 1, the improvement of the classical PointNet++ is mainly conducted in 2 aspects: (1) increasing the depth of the model as the deeper model has stronger feature extraction ability and, also, facilitates the smooth reduction in the radius of the receptive field; (2) reducing the radius of the model's receptive field as the smaller receptive field can improve the model's ability to perceive the detailed features of the pigs. The structure of the improved point cloud segmentation model for pigs with PointNet++ is shown in Figure 7.

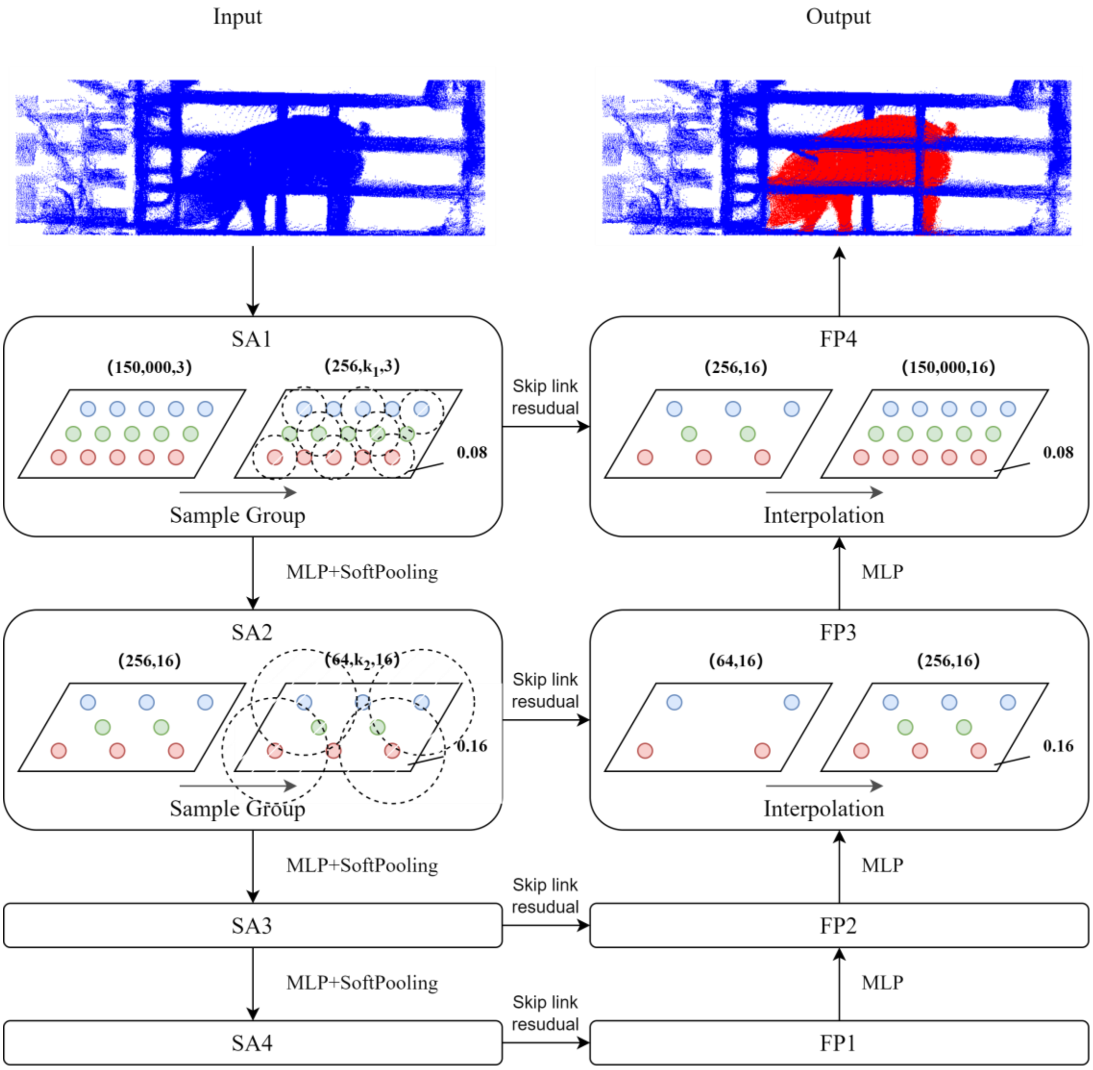


Figure 7. Improved point cloud segmentation model of pigs with PointNet++.

Improvements Based on Pooling Methods

The pooling layer is a commonly used layer structure in deep learning and its central role is to reduce the computation while keeping the main features, reduce the redundancy of the features, and prevent the model from overfitting. In addition, keeping the transformations non-deformed, including translation, scale, and rotation, is also one of its roles. When dealing with complex pig point cloud segmentation tasks, PointNet [25] and PointNet++ [13], two state-of-the-art deep learning models, usually rely on max pooling to perform pooling operations. Although these methods have achieved remarkable success in several domains, their limitations in terms of fineness of feature extraction, adaptability to complex data distributions, robustness, and preservation of spatial relationships remain problematic. To address these issues, we use the SoftPool pooling method to replace max pooling to serve as the pooling layer of the network and improve the model's performance in handling the task of pig segmentation in a complex environment.

SoftPool is a fast and efficient exponentially weighted activation down-sampling method [26]. SoftPool is differentiable and is based on a softmax weighting method to preserve the basic attributes of the inputs while amplifying the activation of features of greater intensity. Relative to other pooling methods, SoftPool is a method for pooling that assigns weights instead of simply selecting the maximum or average value. This approach preserves more information and makes the model differentiable.

This way, SoftPool is better able to retain information about the location of features and is more sensitive to anomalous data. The ratio of the natural index of the activation values in the local domain to the sum of the indices of all activation values in the domain is the weight and all weights in the domain are weighted and activated cumulatively to obtain the output of the pooling operation [27]. The weight calculation formula and weighted activation formula are (2) and (3):

$$Wi = \frac{e^{a_i}}{\sum\limits_{j \in R} e^{a_i}} \tag{2}$$

$$\tilde{a} = \sum_{i \in R} W_i * a_i \tag{3}$$

where R is the maximum approximation in the activation domain; a_i is the activation values of the ith element; W_i is the weight of the ith element after activation; and $\tilde{a}$ is the output of the pooling operation.

SoftPool uses a normalization factor to calculate the weights of activation values. This ensures that all weights of activation values sum to 1. The weight of each activation value depends on its value relative to its neighboring activation values. Unlike maximum pooling, it does not completely ignore the other values. So, the larger the activation value, the higher the corresponding weight; however, it still considers other values. After calculating the weights, they are applied to the respective activation values, resulting in a weighted activation output. SoftPool retains more feature information than a maximum value.

Figure 8 shows the SoftPool down-sampling process. A feature map is input and the colored portion of the map represents the 3 × 3 sized region being sampled. Using the weighting formula, the weight of each element in the selection is calculated and each weight is multiplied with the corresponding activation value and accumulated to obtain the final result, in which the weights are non-linearly transformed along with the corresponding activation values.

The improved PointNet++ network model based on the SoftPool pooling method used in this paper has the following four main features: (1) smoothing feature extraction: by calculating the weighted average within a neighborhood to perform information aggregation, SoftPool pooling can capture the local details in the point cloud in a more subtle way, which reduces the loss of information; this smoothing feature is crucial for accurately depicting the morphological and structural characteristics of the pigs; (2) adapt to irregular distribution: the natural state of the pigs in the farming environment results in a non-uniform and anisotropic distribution of the point cloud data; SoftPool pooling's weighted averaging mechanism can naturally cope with such a situation and its flexibility allows the model to process such data more efficiently and accurately; (3) enhance robustness: due to the inevitable noise and outliers in the aquaculture environment, the robustness of the model becomes an important indicator for assessing its usefulness; the use of SoftPool pooling reduces the influence of outliers and enhances the model's immunity to noise, thus demonstrating better stability in the face of real-world data; (4) preservation of spatial relationships: unlike the traditional pooling approach that only considers feature values, SoftPool pooling also considers the spatial relationships between feature points; this retention of spatial information is critical to understanding the 3D structure of the point cloud data and it helps the model to distinguish better and recognize the pigs and their environment.

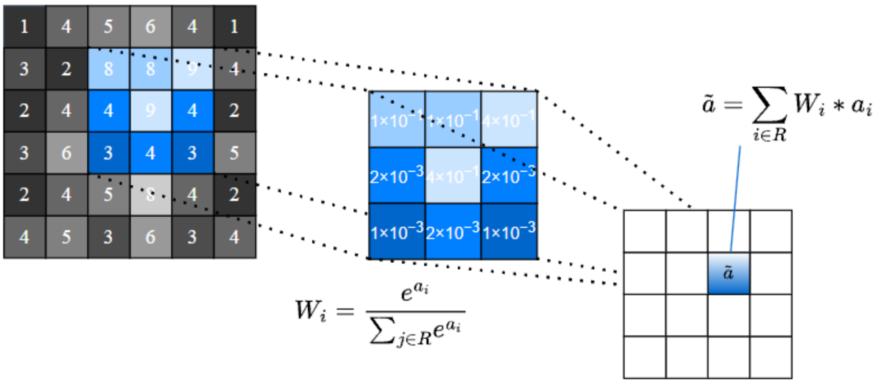

1 4 5 6 4 1
3 2 8 8 9 4
2 4 4 9 4 2
3 6 3 4 3 5
2 4 5 8 4 2
4 5 3 6 3 4
1×10^{-1} 1×10^{-1} 4×10^{-1}
2×10^{-3} 4×10^{-1} 2×10^{-3}
1×10^{-3} 2×10^{-3} 1×10^{-3}
$\tilde{a} = \sum_{i \in R} W_i * a_i$
$\tilde{a}$
$W_i = \frac{e^{a_i}}{\sum_{j \in R} e^{a_i}}$

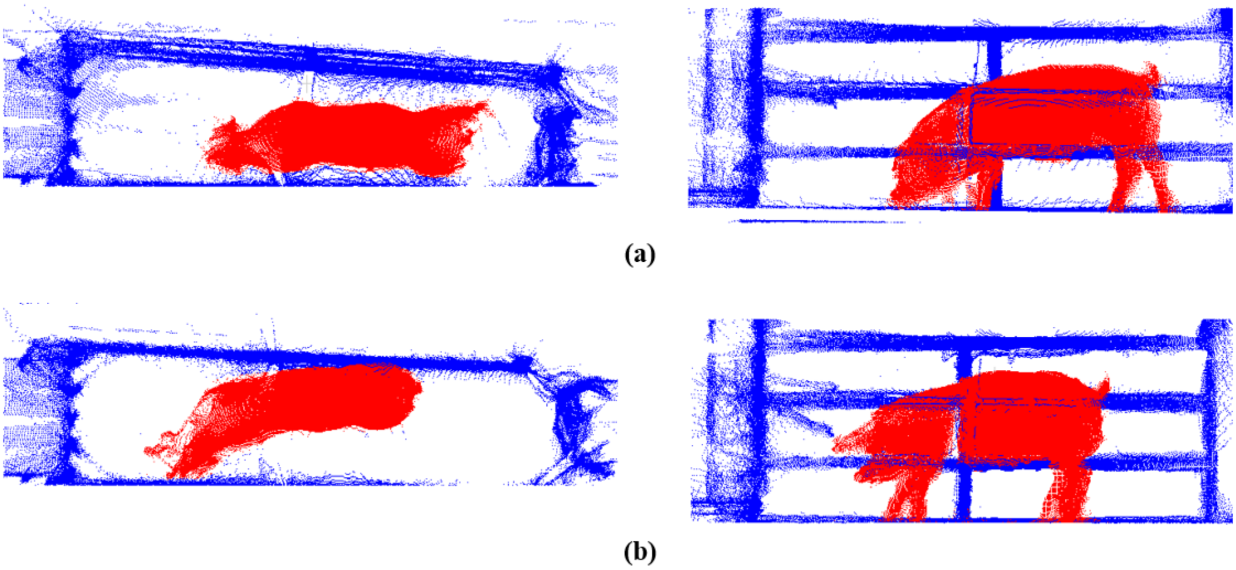
Figure 8. SoftPool down-sampling process.

In summary, the improvement based on SoftPool pooling brings multiple advantages to the PointNet++ network, especially in smoothing feature extraction, adapting to irregular data distribution, improving anti-interference ability, and better utilizing spatial context information. These improvements enhance the accuracy and efficiency of pig point cloud segmentation and provide a reliable basis for further research and applications.

2.5. Data Annotation

In deep learning point cloud segmentation methods, making datasets requires manual labeling of the original data. The manually labeled data are affected by subjective factors and different poses of the pig and the point cloud data labeled by two people will have slight differences. In order to reduce the influence of manual subjective factors and posture changes, the manually labeled experimental dataset is labeled by the same person.

The training dataset was prepared and labeled using CloudComparev2.13 software. According to the localization of the key points, a total of two parts of the pig body and the railings were labeled, sequentially, with the railings marked with "0" and the pig body marked with "1". As shown in Figure 9, the blue area is the point cloud of the railings containing the acquisition environment and the red area is the annotated point cloud of the pig body. A challenge, as depicted in Figure 9b, arises when the pig body comes into contact with the railings during the point cloud annotation process, leading to inaccuracies in the annotations, Fortunately, this phenomenon exists less and, for the deep learning model, it is still possible to learn the full picture of the point cloud of the railings shown in Figure 9a and it does not have much impact on the results.


(a)

(b)

Figure 9. Annotation results for some pig bodies; (**a**,**b**) are annotated data.

3. Results and Discussion

To identify the optimal model for pig body segmentation in complex environments, we trained and tested two models: PointNet and the enhanced version, PointNet++. The system environment for training the models in this paper is Intel Xeon Platinum 8375C CPU from Intel Corporation, Santa Clara, USA, NVIDIA RTX A6000 GPU from NVIDIA Corporation, Santa Clara, USA, 48GB graphics memory, deep learning framework Pytorch1.12.0, and Cuda11.3. The learning rate of the model was set to 0.001; the learning rate decay to 0.5, the number of rounds decayed every 20 iterations; the batch size of the model training was set as 2; and each model was trained 251 times to ensure convergence. We divided the 529 sets of data with segmentation labels into three different sets: 423 sets for the training set, 53 sets for the validation set, and 53 sets for the test set. The data from the training set were sent to the neural network for deep learning and the parameters were tuned in the validation set. The model parameters saved in the validation set with the best segmentation results were used to segment all data parts. Select the model that performs best on both the training and test sets as the optimal model. The performance of both models was evaluated based on accuracy and mIoU metrics [28,29]. The calculation equation of accuracy is as follows (4):

$$Accuracy = \frac{\sum_{i=0}^{k} p_{ii}}{\sum_{i=0}^{k} \sum_{j=0}^{k} p_{ij}} \tag{4}$$

The *mIoU* denotes the overlap ratio between the ground truth region and the predicted region after segmentation, which is used to measure the segmentation accuracy. The calculation equation of *mIoU* is as follows (5):

$$mIoU = \frac{1}{k+1} \sum_{i=0}^{k} \frac{p_{ii}}{\sum_{j=0}^{k} p_{ij} + \sum_{j=0}^{k} p_{ji} - p_{ii}} \tag{5}$$

In Equations (4) and (5), k + 1 is the total number of classes (including the empty class); p_{ii} is the number of correctly predicted points; and p_{ij} and p_{ji} represent the number of false negatives and false positives, respectively. In general, *mIoU* is calculated on a class-by-class basis; after calculations are performed for each class of debits, they are accumulated and averaged to obtain an overall evaluation.

For the ablation experiments, Figures 10 and 11 compare the mIoU versus accuracy changes of the PointNet++ network using different modules with the PointNet++ base network during the training process. As shown in Figures 10 and 11, the improved PointNet++ obtains a better learning ability and has almost converged after about 40 rounds of training, with an optimal mIoU of 96.52% and accuracy of 98.33% in 251 rounds of training. In contrast, the original PointNet++ network only gradually converged after 50 rounds of training, with an optimal mIoU of 92.98% and accuracy of 96.37% in 251 rounds of training. The experimental results demonstrate that the improved PointNet++ network has a stronger learning ability on the pig body point cloud dataset.

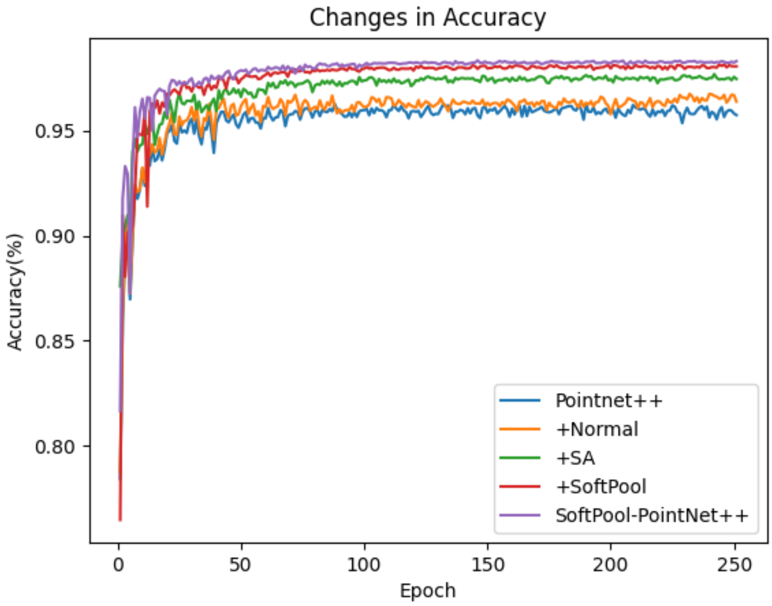


Figure 10. Changes in accuracy.

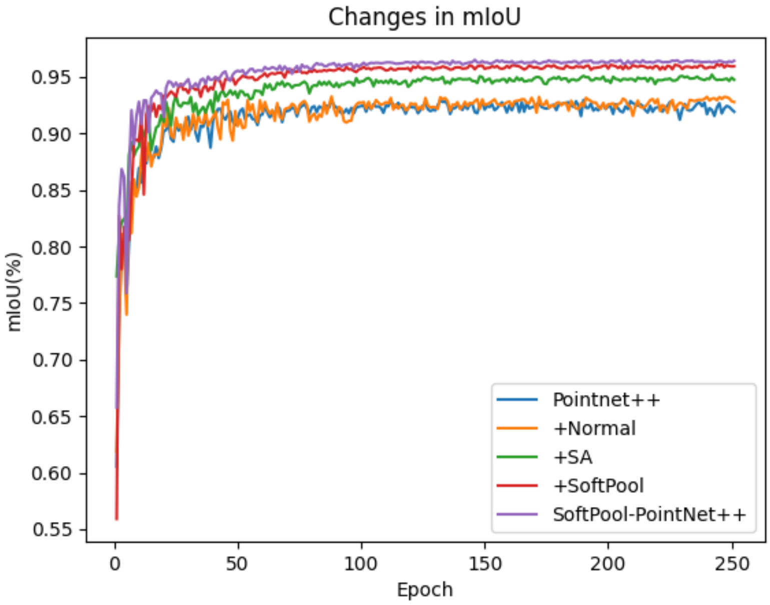


Figure 11. Changes in mIoU.

To further validate the effectiveness of the improvements made to the PointNet++ network in this study, we conducted ablation experiments with different module combinations, including using plane fitting to compute normal vectors (+Normal) on top of the basic PointNet++ model, modifying the network structure and parameters in the set abstraction module (+SA), and employing SoftPool pooling in the feature extraction module (+SoftPool). Table 2 summarizes the results of these experiments. Table 3 presents the segmentation results under different module combinations. From Table 3, it is observed that compared to the segmentation results of the basic PointNet++ model, incorporating normal vector information through plane fitting (+Normal) leads to improvements in both mIoU and accuracy. The normal vector, as an attribute of each point in the point cloud, contains important spatial information and the geometry of the surface can be reconstructed more accurately. Modifying the network structure and parameters in the set abstraction module (+SA) enhances the segmentation accuracy to some extent; it is significantly less effective than the improvement in the pooling method using SoftPool in the feature extraction module (+SoftPool). However, the SA layer continuously adjusts the feeler fields for feature extraction by iteratively sampling the local structure of the point cloud. It may not be as acceptable to capture details as SoftPool assignments of weights for pooling. However, after adopting SoftPool pooling in the feature extraction module (+SoftPool), the mIoU and accuracy of the network are further improved by 3.16% and 1.78%, respectively, compared to the original PointNet++ model. SoftPool facilitates down-sampling and retains more feature information, which is more conducive to subsequent feature extraction. Moreover, when modifying the network structure in the set abstraction module and using SoftPool pooling along with normal vectors (SoftPool–PointNet++), the overall mIoU and accuracy are increased by 3.54% and 1.96%, respectively, compared to the original PointNet++ model. During the process of ablation experiments, it is evident that each module contributes to the performance improvement of the network model. However, this improvement effect is not directly cumulative but rather involves minor enhancements built upon an improved foundation.

Table 2. Description of the different modules used for ablation experiments.

Module (in Software)	**Clarification**
PointNet++	Basic model
+Normal	Calculate the normal vector using plane fitting
+SA	Deepening the feature extraction network in PointNet++ by one layer
+SoftPool	Adoption of SoftPool Pooling
SoftPool-PointNet++	Improved PointNet++ network based on SoftPool pooling approach

Table 3. Results of ablation experiments.

Model	**Testing Sets**	
	Avg Accuracy(%)	**Avg mIoU (%)**
PointNet++	96.37	92.98
+Normal	96.79	93.27
+SA	97.68	95.21
+SoftPool	98.15	96.14
SoftPool-PointNet++	98.33	96.52

The information presented in Table 4 compares the model size and inference time for various modules. The inference time has been increased because of the utilization of the plane fitting method for computing the normal vector information (+Normal). Additionally, +Normal has observed an improvement based on the set abstraction module (+SA). However, due to the increase in the depth of the model, both the model size and inference time have also increased. As the depth and size of a model increase, the inference time also increases. Additionally, the reference time of the model also increases. Moreover, the feature extraction module's change to SoftPool pooling, as implemented by +SoftPool, could potentially lead to longer processing times per point. This is because it involves an exponentially weighted cumulative activation and the retention of more information in the down-sampled activation mapping. SoftPool increases the inference time; however, it also enhances the network's performance by retaining more information. When using SoftPool pooling in combination with a deep network model, like SoftPool–PointNet+, the accuracy and mIoU improve by 0.18% and 0.38% respectively. In contrast, the inference time increases by 53.97 ms compared to the SoftPool-only pooling approach, which may be attributed to the fact that each additional SA layer, additional feature extractions, and aggregation operations are required for the point cloud data. These operations include calculating the distance between points, searching for nearest neighbor points, and performing feature transformations, which are relatively time-consuming computational steps. Meanwhile, the FPS of the sampling layer will select the surrounding points to form a region around each center point when selecting the center point and then process them as input samples. This process requires multiple iterations, each of which may involve complex computations, thus increasing the inference time. It is worth noting that improving either the SA layer or the SoftPool pooling approach results in improved performance based on the original PointNet++ model. There is a need to optimize the model for better performance without compromising the segmentation effect.

Table 4. Model size and inference time for different modules.

	PointNet++	**+Normal**	**+SA**	**+SoftPool**	**SoftPool–PointNet++**
Model size (MB)	5.36	5.36	5.96	5.36	5.96
Inference time (ms)	165.58	177.69	217.39	189.58	243.55

Figure 12 demonstrates the comparison of the visualization effect of the pig point cloud using the improved PointNet++ network model after segmentation and the traditional PointNet++ network model with manually labeled data. The segmentation results show that the improved PointNet++ network model effectively segments the point clouds of the pig and the railing, even for the data with overlapping point clouds of the pig and the railing, and can complete the segmentation task well for the different postures of the pig while the traditional PointNet++ network model for the overlapping problem will appear to divide a part of the pig's body into the railing. From the point cloud of the pig in Column 3, the basic PointNet++ model is not good enough to segment the part of the pig's body that probes the railings, such as the pig's snout shown in the figure; although the improved PointNet++ network model splits the point cloud of the pig that has too much contact with the railings, there is still a part of the pig's snout that is not recognized and the segmentation of some details is still insufficient. SoftPool–PointNet++ has a more accurate segmentation effect and the removal of the segmentation boundary is comparable to manual labeling.

To evaluate the effectiveness and advantages of the method presented in this paper, SoftPool–PointNet++ segmentation results were compared with various point cloud segmentation algorithms on our collected dataset. The comparison results are shown in Table 5. Even though PointNet has the ability to capture the overall information of the point cloud, it struggles to extract detailed local features. This implies that it may not be able to provide enough information when dealing with complex tasks that require a finer-grained analysis of local features, such as object detection or complex scene understanding. PointCNN [30] is an extension and generalization of the classical convolutional neural network and its core operation is X-Conv. X-Conv first weights and displaces the input points to capture the spatial-local associations in the point cloud. Finally, the high-level abstract features are transformed into the final output by one or more fully connected layers. Kernel point convolution (KPConv) [31] is a convolutional neural network for processing point cloud data whose core idea is to perform a weighted summation operation similar to planar convolution through a set of predefined kernel points. The core idea is to perform a weighted summation operation similar to planar convolution through a set of predefined kernel points. KPConv can efficiently capture the spatial structure and local relationships of point cloud data by defining kernel points and correlating relationships. SoftPool–PointNet++, proposed in this paper, has introduced a new methodology to enhance the feature extraction process and improve the segmentation ability. In this approach, the SA layer has been increased to refine the feature information and improve the depth of the model. Then, SoftPool has been utilized to reduce the computation and prevent feature redundancy while retaining more features. Figures 13 and 14 show our method's mIoU vs. accuracy variation with different point cloud segmentation algorithms for the training process on our collected dataset. In the pig body point cloud segmentation experiments, the SoftPool–PointNet++ network algorithm outperforms the other methods. Figures 13 and 14 illustrate the mIoU versus accuracy variation with different point cloud segmentation.

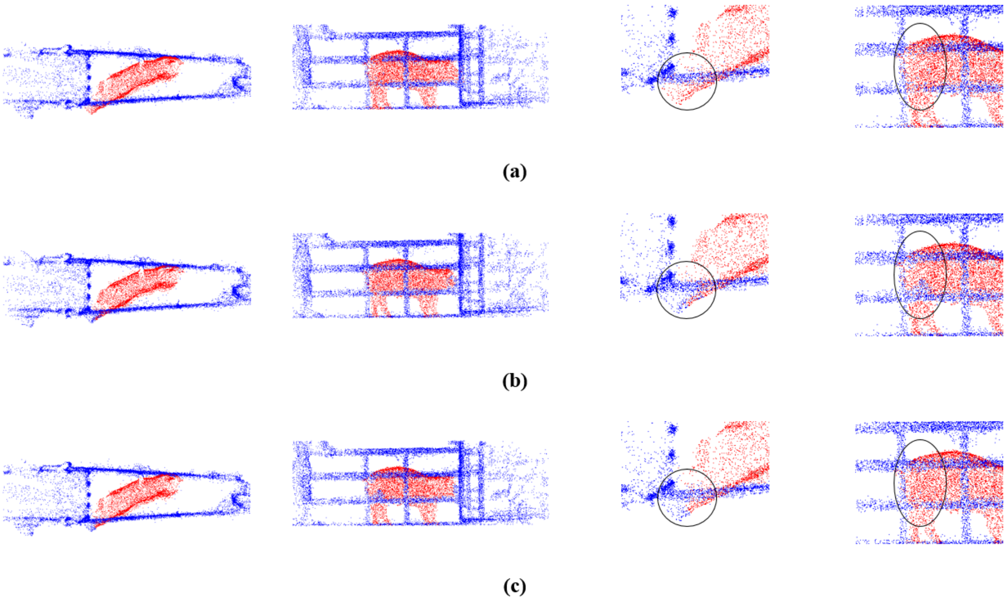


Figure 12. Comparison of point cloud segmentation visualization effect of a pig body in some postures in a complex environment. The first and second columns are top and flat views of the point cloud segmentation visualization of the pig body, the third column is an enlarged view of the point cloud segmentation visualization of the part of the pig body exploring the railing, and the fourth column is a visualization of the segmentation effect of the part of the pig body overlapping with the railing. Circles show the differences between the different visualizations. (**a**) Manually labeled data of pig point cloud; (**b**) PointNet++ segmentation results of pig point cloud; (**c**) Modified PointNet++ segmentation results of pig point cloud.

Table 5. Performance comparison of each algorithm on our dataset.

Model	Testing Sets	
	Avg Accuracy(%)	Avg mIoU (%)
PointNet	88.48	79.88
PointCNN	94.41	88.97
PointNet++	96.37	92.98
KPConv	97.52	94.91
SoftPool–PointNet++	98.33	96.52

SoftPool–PointNet++ has achieved good results in pig point cloud segmentation in complex environments. However, any method that relies on point cloud data may face challenges in highly complex or heavily occluded scenarios where the data may be incomplete or noisy. Additionally, unknown factors occurring during point cloud acquisition will affect the quality of collected pig point cloud data, which increases the difficulty of point cloud acquisition due to unpredictable pig actions during the process. To address this problem, it is necessary to optimize the acquisition scheme. Furthermore, the introduction of SoftPool will increase the computation and complexity of the network, which may pose challenges in applications requiring real-time processing. Future work should also consider the effect of different pig poses on segmentation results as the pig's point cloud will be collected in various poses after entering the acquisition environment.

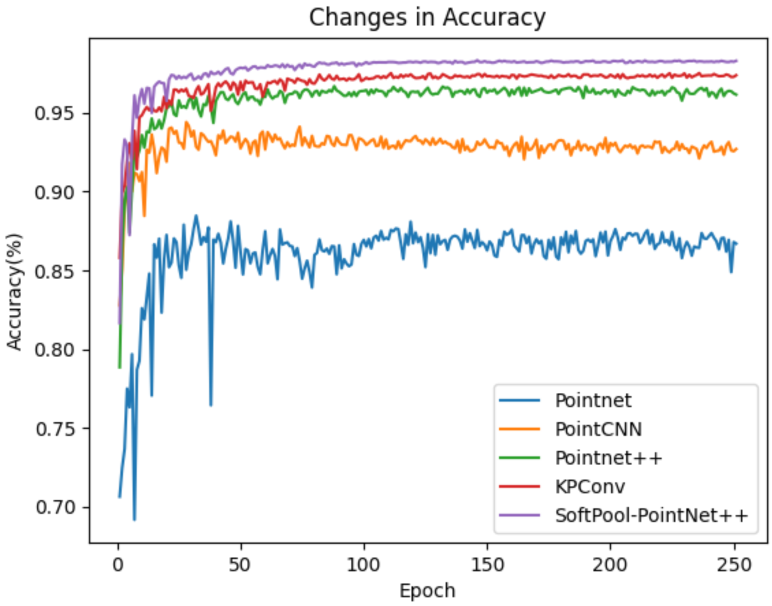


Figure 13. Changes in the accuracy of different algorithms.

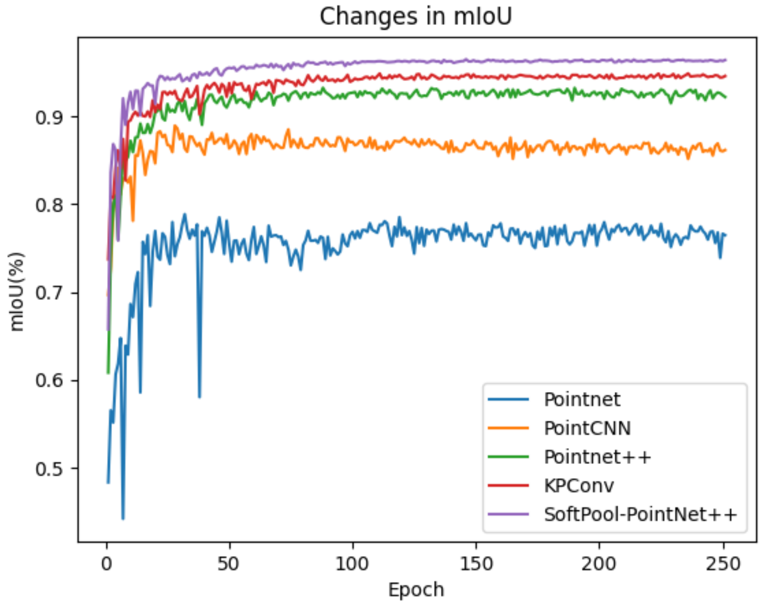


Figure 14. Changes in the mIoU of different algorithms.

4. Conclusions

There are certain parts in the pig and railing point cloud that overlap excessively, causing PointNet++ to divide the overlapping part into the railing point cloud. This could be due to dust present during the data collection process, slight camera occlusion while taking pictures, or the pig's violent reaction due to fear during data collection. As a result, the detailed features of the railing and the pig's body might have been inaccurately extracted. To overcome this challenge, we propose a novel point cloud segmentation method for pigs in complex environments. The proposed method improves the capability of capturing local features of large livestock, such as pigs, in a free-motion state. It also achieves automatic segmentation of two types of point clouds, namely, the overlapping pig point cloud and the fence point cloud, with an accuracy of 98.33% and mIoU of 96.52%. The results of the experiment demonstrate that the SoftPool–PointNet++ network can accurately segment the point cloud data of pigs in various postures, with visualization effects almost identical to those of manual labeling. However, the network's ability to segment the part of the pig that pokes out of the pen is still inadequate. This may be due to the fact that this part of the pig's body is usually the edge part, making it difficult to extract features. The acquisition environment will be better optimized in the subsequent work. The research contributes to the advancement of 3D point cloud segmentation technology in animal husbandry, offering new perspectives and methods for subsequent research. This is particularly useful for segmentation tasks involving highly dynamic and irregularly shaped objects. In this study, we examine the potential of PointNet++, a traditional point cloud processing model, to handle more intricate and dynamic real-world settings. Our research offers an effective point cloud processing approach for precision livestock management in agriculture.

Author Contributions: Conceptualization, W.M. and K.C.; methodology, X.Q.; software, X.X. (Xianglong Xue); validation, Z.X.; formal analysis, M.L.; investigation, Y.G.; resources, W.M.; data curation, K.C.; writing—original draft preparation, K.C.; writing—review and editing, W.M.; visualization, K.C.; supervision, X.X. (Xingmei Xu); project administration, R.M.; funding acquisition, Q.L. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Beijing Academy of Agriculture and Forestry Sciences (JKZX202214), the National Key R&D Program of China (2022ZD0115702), the Sichuan Science and Technology Program (2021ZDZX0011), and the Technological Innovation Capacity Construction of Beijing Academy of Agricultural and Forestry Sciences (KJCX20230204).

Institutional Review Board Statement: Ethical review and approval were waived for this study due to the fact that our study did not cause any stress to the pigs and there was no contact with the pigs during the collection process, which would not affect the normal activities of the pigs or the normal work of the pen.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

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

References

1. Guo, Y.; Wang, H.; Hu, Q.; Liu, H.; Liu, L.; Bennamoun, M. Deep Learning for 3D Point Clouds: A Survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2021**, *43*, 4338–4364. [CrossRef] [PubMed]
2. Vo, A.V.; Truong-Hong, L.; Laefer, D.F.; Bertolotto, M. Octree-based region growing for point cloud segmentation. *ISPRS J. Photogramm. Remote Sens.* **2015**, *104*, 88–100. [CrossRef]
3. Vosselman, G. Point cloud segmentation for urban scene classification. *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.* **2013**, *40*, 257–262. [CrossRef]
4. Zhang, L.; Wang, H. A novel segmentation method for cervical vertebrae based on PointNet++ and converge segmentation. *Comput. Meth. Prog. Biomed.* **2021**, *200*, 105798. [CrossRef]
5. Koo, B.; Jung, R.; Yu, Y. Automatic classification of wall and door BIM element subtypes using 3D geometric deep neural networks. *Adv. Eng. Inform.* **2021**, *47*, 101200. [CrossRef]
6. Kowalczuk, Z.; Szymański, K. Classification of objects in the LIDAR point clouds using Deep Neural Networks based on the PointNet model. *IFAC-PapersOnLine* **2019**, *52*, 416–421. [CrossRef]
7. Chen, X.; Ma, H.; Wan, J.; Li, B.; Xia, T. Multi-view 3d object detection network for autonomous driving. In Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 1907–1915.
8. Elnashef, B.; Filin, S.; Lati, R.N. Tensor-based classification and segmentation of three-dimensional point clouds for organ-level plant phenotyping and growth analysis. *Comput. Electron. Agric.* **2019**, *156*, 51–61. [CrossRef]
9. Wu, G.; Li, B.; Zhu, Q.B.; Huang, M.; Guo, Y. Using color and 3D geometry features to segment fruit point cloud and improve fruit recognition accuracy. *Comput. Electron. Agric.* **2020**, *174*, 105475. [CrossRef]

10. Li, J.W.; Ma, W.H.; Li, Q.F.; Zhao, C.J.; Tulpan, D.; Yang, S.; Ding, L.Y.; Gao, R.H.; Yu, L.G.; Wang, Z.Q. Multi-view real-time acquisition and 3D reconstruction of point clouds for beef cattle. *Comput. Electron. Agric.* **2022**, *197*, 106987. [CrossRef]
11. Shi, S.; Yin, L.; Liang, S.H.; Zhong, H.J.; Tian, X.H.; Liu, C.X.; Sun, A.D.; Liu, H.X. Research on 3D surface reconstruction and body size measurement of pigs based on multi-view RGB-D cameras. *Comput. Electron. Agric.* **2020**, *175*, 105543. [CrossRef]
12. He, H.X.; Qiao, Y.L.; Li, X.M.; Chen, C.Y.; Zhang, X.F. Automatic weight measurement of pigs based on 3D images and regression network. *Comput. Electron. Agric.* **2021**, *187*, 106299. [CrossRef]
13. Zhang, X.J.; Liu, J.Y.; Zhang, B. Research on Object Panoramic 3D Point Cloud Reconstruction System Based on Structure from Motion. *IEEE Access* **2022**, *10*, 110064–110075. [CrossRef]
14. Su, H.; Maji, S.; Kalogerakis, E.; Learned-Miller, E. Multi-view Convolutional Neural Networks for 3D Shape Recognition. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 945–953.
15. Feng, Y.F.; Zhang, Z.Z.; Zhao, X.B.; Ji, R.R.; Gao, Y. Gvcnn: Group-View Convolutional Neural Networks for 3D Shape Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 264–272.
16. Lawin, F.; Danelljan, M.; Tosteberg, P.; Bhat, G.; Khan, F.; Felsberg, M. Deep projective 3D semantic segmentation. In Proceedings of the International Conference on Computer Analysis of Images & Patterns, Ystad, Sweden, 22–24 August 2017; pp. 95–107.
17. Boulch, A.; Guerry, J.; Le Saux, B.; Audebert, N. SnapNet: 3D point cloud semantic labeling with 2D deep segmentation networks. *Comput. Graph.* **2018**, *71*, 189–198. [CrossRef]
18. Maturana, D.; Scherer, S. VoxNet: A 3D Convolutional Neural Network for real-time object recognition. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 922–928.
19. Huang, J.; You, S. Point cloud labeling using 3D Convolutional Neural Network. In Proceedings of the 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, Mexico, 4–8 December 2016; pp. 2670–2675.
20. Wang, L.; Huang, Y.C.; Shan, J.; He, L. MSNet: Multi-Scale Convolutional Network for Point Cloud Classification. *Remote Sens.* **2018**, *10*, 612. [CrossRef]
21. Hu, H.; Yu, J.C.; Yin, L.; Cai, G.Y.; Zhang, S.M.; Zhang, H. An improved PointNet++ point cloud segmentation model applied to automatic measurement method of pig body size. *Comput. Electron. Agric.* **2023**, *205*, 107560. [CrossRef]
22. Liu, L.; Zhang, A.; Xiao, S.; Hu, S.; He, N.; Pang, H.; Yang, S. Single Tree Segmentation and Diameter at Breast Height Estimation With Mobile LiDAR. *IEEE Access* **2021**, *9*, 24314–24325. [CrossRef]
23. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018.
24. Chang, A.X.; Funkhouser, T.; Guibas, L.J.; Hanrahan, P.; Huang, Q.X.; Li, Z.M.; Savarese, S.; Savva, M.; Song, S.; Su, H.; et al. Shapenet: An information-rich 3d model repository. *arXiv* **2015**, arXiv:1512.03012v1.
25. 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 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 652–660.
26. Stergiou, A.; Poppe, R.; Kalliatakis, G. Refining activation downsampling with SoftPool. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 10357–10366.
27. Fu, R.; Zhou, G. Automatic Evaluation of Facial Paralysis with Transfer Learning and Improved ResNet34 Neural Network. In Proceedings of the 2023 15th International Conference on Intelligent Human-Machine Systems and Cybernetics, Hangzhou, China, 26–27 August 2023; pp. 218–222.
28. Armeni, I.; Sener, O.; Zamir, A.R.; Jiang, H.; Brilakis, I.; Fischer, M.; Savarese, S. 3D semantic parsing of large-scale indoor spaces. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1534–1543.
29. Hackel, T.; Savinov, N.; Ladicky, L.; Wegner, J.; Schindler, K.; Pollefeys, M. Semantic3D.net: A new large-scale point cloud classification benchmark. *arXiv* **2017**, arXiv:1704.03847v1. [CrossRef]
30. Li, Y.Y.; Bu, R.; Sun, M.C.; Wu, W.; Di, X.H.; Chen, B.Q. PointCNN: Convolution on X-transformed points. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Montreal, QC, Canada, 3–8 December 2018.
31. Thomas, H.; Qi, C.R.; Deschaud, J.; Marcotegui, B.; Goulette, F.; Guibas, L. KPConv: Flexible and Deformable Convolution for Point Clouds. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision (ICCV), Seoul, Republic of Korea, 27 October–2 November 2019.

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