

# Multiscale Segmentation Extraction Method for Mars Craters Based on High-Resolution DEM Data

Team Members:

Tang Sui(id: tsui5), Chen Wei(id: cwei68), Qianheng Zhang(id: qzhang533)

## ABSTRACT

This study investigates the method of extracting Martian impact crater segmentation based on high-resolution DEM. By examining the structures, segmentation performance, and result analysis of three different classic point cloud deep learning networks—PointNet, PointNet++, and RandLA-Net—this research focuses on the segmentation tasks based on point cloud data features and the performance of multi-scale sampling neural networks. Using HRSC DEM (100m) Martian surface digital elevation model data, a Martian surface impact crater point cloud dataset was created. Classic models were reproduced, and the performance results of the models were analyzed and discussed. The segmentation and extraction of impact craters were analyzed in relation to the models' design to address the unordered nature of point cloud data, the correlation between points, and transformation invariance. By comparing the segmentation and extraction of craters of different sizes within the same sample, this study explores the issue of multi-scale sampling in the models. This lays the foundation for future multi-classification problems of the Martian surface geographical environment and the landing site selection for Mars rovers.

**Key words:** HRSC DEM, martian surface crater, point cloud data, point cloud semantic segmentation, deep learning

# 1 Introduction

## 1.1 Research Background

With the rapid advancement of space technology, Mars exploration has attracted growing attention. Among related research tasks, identifying and cataloging the sizes and spatial distributions of impact craters is essential for understanding Martian topography and geomorphology. Crater studies not only contribute to the characterization of impact features themselves, but also provide crucial foundations for higher-level tasks such as landing-site selection for future Mars landers, investigations of potential Martian life, and studies of Martian geological structures.

In Martian remote sensing, optical imagery has developed relatively maturely; however, its utility is limited by the restricted coverage of high-resolution remote sensing data, which makes detailed and precise global crater investigations difficult. Additionally, spectral imagery is prone to shadow-induced information loss and often cannot directly represent surface geometric structures with high accuracy [1].

To address these limitations, this study adopts point-cloud-based semantic segmentation methods using high-resolution DEM-derived point-cloud data, which better preserve geometric information and possess broader surface coverage. Compared with traditional non-machine learning or classical machine learning approaches, deep learning can extract higher-level abstract features, leverage contextual and global information, and thus provide improved accuracy, robustness, and generalization [2]. Experiments in this study use a point-cloud dataset derived from Martian DEMs for model training and testing, thereby offering a basis for more complex future tasks such as landing-site selection and 3D surface reconstruction of Mars [3].

## 1.2 Development of Mars Exploration Missions

Human exploration of Mars spans more than 60 years, and today Mars hosts more spacecraft than any other planet beyond Earth—including orbiters, landers, and rovers. Over time, mission priorities have shifted from studying Mars' ancient past to assessing the feasibility of future human colonization.

On Earth, all known life requires water. If life once arose and evolved on Mars, sustained water availability would have been indispensable. Thus, the search for past life on Mars focuses on locating regions where liquid water may have once existed stably. Data from multiple NASA Mars missions indicate the presence of liquid water beneath the surface in a few locations, and within water-ice deposits near the Martian poles.

As research on Martian water evolved, scientific attention expanded to studying Mars' climate, which influences both the existence of water and the planet's potential habitability. Current Martian climate dynamics are driven by seasonal variations in CO<sub>2</sub> ice caps, large-scale dust transport, and vapor exchange between the surface and atmosphere—producing Mars' most dynamic weather feature: dust storms. Observing dust storms and other weather patterns throughout a Martian year is essential not only for understanding past habitability but also for supporting future human exploration.

Parallel to climate research, describing Martian geological characteristics is another vital task for understanding the similarities and differences between Earth and Mars. Processes such as aeolian activity, fluvial erosion, volcanism, tectonics, and impact cratering have shaped Mars' surface. Determining the ages and compositions of different rock types—especially those formed in aqueous environments—is central to reconstructing Mars' geologic history.

With ongoing Mars orbiter missions, landers, rovers, and increasingly sophisticated instruments, our understanding of Mars continues to deepen, raising new scientific questions. However, all such research relies on a fundamental understanding of the Martian surface—particularly impact craters, which represent one of the most distinctive geological differences between Earth and Mars. Craters also play a key role in landing-site selection and geological targeting for future robotic or human exploration. Enhancing crater understanding therefore supports a wide range of future endeavors.

## 1.3 Characteristics of Martian Impact Craters

Impact craters are surface geological structures formed when small celestial bodies collide with terrestrial planets or moons at extremely high velocities. As early as 1609, Galileo observed lunar craters using a telescope. In 1964, NASA's Mariner 4 became the first spacecraft to fly by Mars, followed by additional orbiters and landers from the United States and the Soviet Union, providing the first fundamental understanding of Martian geology.

In the 1980s, NASA launched a series of high-precision Mars orbiters that produced high-resolution imagery and global topographic maps. Since then, research on Martian impact craters has become increasingly sophisticated.



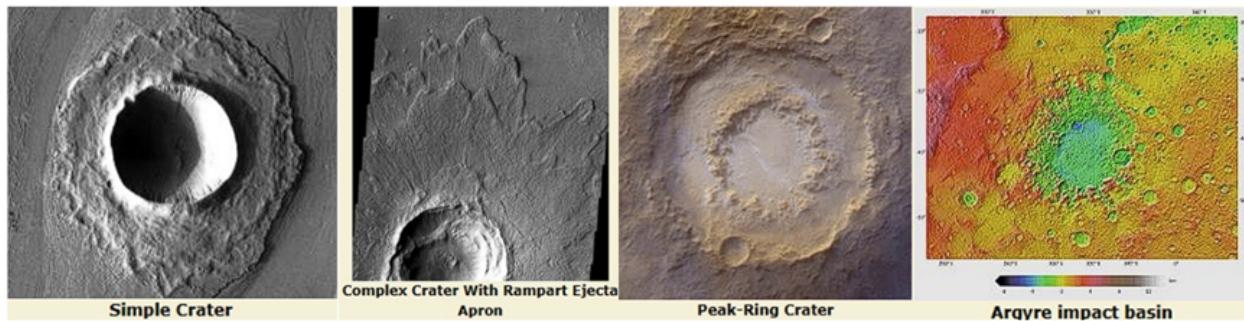
**Figure 1.1** Three stages of Mars missions—from top to bottom: flyby, orbit, landing (<https://marsed.asu.edu>)

Early telescopic observations categorized craters by shape and diameter into four types: micro-craters, simple craters, complex craters, and impact basins [4]. With higher-resolution

crater datasets, more scientific classification schemes were developed, including simple craters, complex craters, multi-ring basins, and buried craters [5].

- **Simple craters** (diameter < 5 km) are bowl-shaped with a depth-to-diameter ratio of ~0.2, and their inner walls often contain layered structures [6].
- **Complex craters** (diameter > 5 km) contain central peaks, terraced walls, and a flatter floor; their depth-to-diameter ratio can be as low as 0.03.
- **Multi-ring basins** (> 130 km) have concentric mountainous rings; the Hellas and Argyre basins are Mars' most famous examples, characterized by thin crusts.
- **Buried craters** (> 200 km) are typically found in quasi-circular depressions across the northern plains [7].

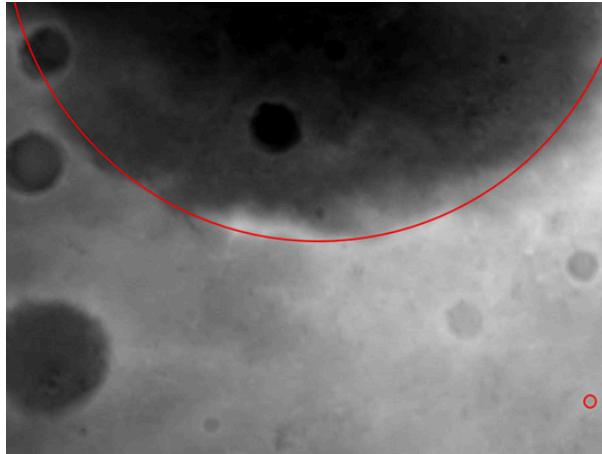
Impact events on Mars often expose subsurface minerals; more than half of the hydrated minerals identified on Mars have been found within craters [8], and pure ice has been discovered in relatively young mid-latitude craters [9].



**Figure 1.2** Categories of Martian impact craters (<https://marsed.asu.edu>)

Craters on Mars vary drastically in size, number, density, and morphology across different regions. The largest confirmed crater exceeds 2,300 km in diameter and lies in the southern hemisphere, while the smallest craters—only tens of centimeters wide—are ubiquitous. This vast scale variation poses significant challenges to crater classification and detection.

Accurately segmenting and extracting craters across multiple scales remains a core scientific challenge.



**Figure 1.3** Vast scale differences between Martian craters

## 1.4 Related Work

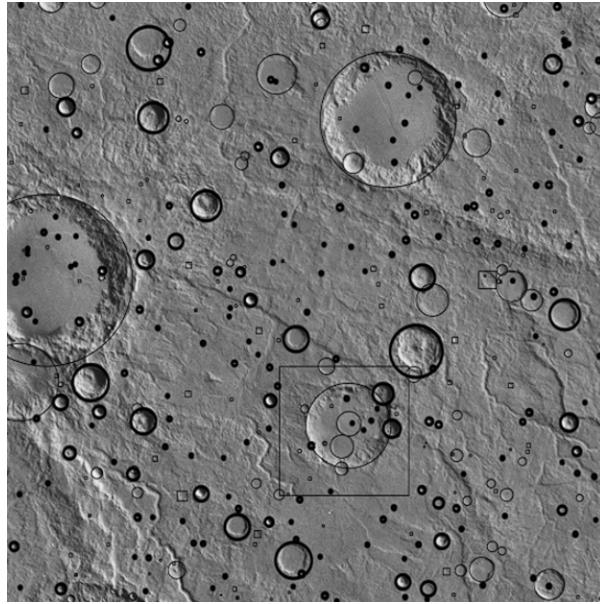
Early crater detection relied on manual visual interpretation of imagery, a process that was time-consuming, labor-intensive, and inconsistent across interpreters [10]. With advances in computing, automated crater detection methods have expanded, primarily falling into two categories: image-based methods and 3D point-cloud–based methods.

### 1.4.1 Image-Based Approaches

Traditional non-machine learning crater detection typically uses pattern-recognition techniques to extract crater edges. Enhancing edge features is therefore crucial, with the most common methods involving Hough transforms or template matching.

- Salamuniccar et al. [11] used fuzzy edge detectors with Radon/Hough transforms and morphological data to improve accuracy.
- Kim et al. [12] applied gray-level co-occurrence matrices to capture texture features such as contrast and homogeneity, designing adaptive sliding windows based on crater size.
- Cheng et al. [13] performed dual Canny edge detection and used topological constraints and convex-edge analysis to refine crater boundaries.
- Barata et al. [14] used texture-based PCA for segmentation and watershed transforms for local boundary fitting.

Among traditional machine learning methods, **support vector machines (SVMs)** were the most widely applied. SVMs have been successful across many vision tasks such as face detection and digit recognition. In crater detection, Wetzler et al. [15] combined SVMs with scale pyramids to achieve strong performance.



**Figure 1.4** SVM + scale pyramid crater detection on Mars [15]

Stepinski et al. [16] developed the first terrain-based detection algorithm using gradient-based decision trees. Di et al. [17] constructed an AdaBoost classifier using scaled Haar and local binary features. Pedrosa et al. [18] performed crater detection in infrared imagery using FFT-based pattern analysis combined with morphological processing and watershed segmentation.

Traditional machine learning performs well on training-region data but generalizes poorly in new settings due to variability in crater shapes, lighting conditions, scale, and overlap. Deep learning, by automatically learning robust features, achieves much stronger generalization. For example:

- Silburt et al. [19] trained a CNN on lunar DEM images and tested generalization on Mercury.
- DeLatte et al. [2] evaluated hyperparameters and trained on infrared imagery.
- Lee [20] used a Martian DTM and a U-Net structure to detect circular features, with tests on lunar and Mercurian data.

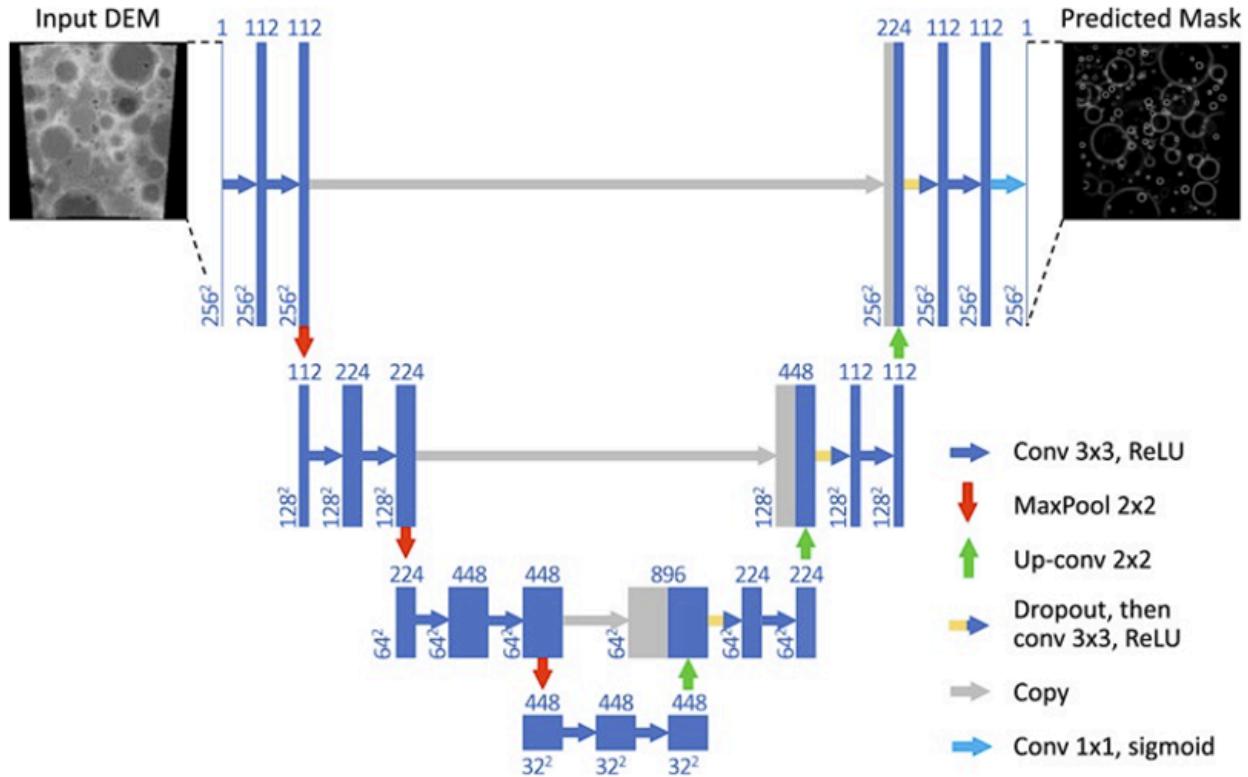


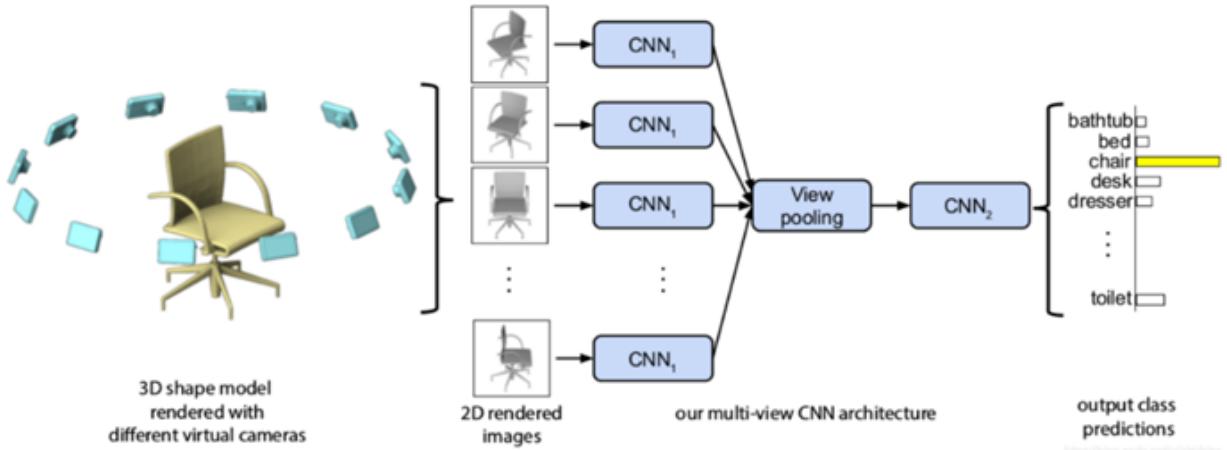
Figure 1.5 CNN-based lunar crater extraction [19]

### 1.4.2 Point-Cloud-Based Approaches

Point-cloud approaches largely rely on deep learning and can be categorized into **projection-based** and **point-based** methods [1].

- Projection-based approaches convert point clouds into structured data such as voxels [21] or multi-view 2D projections [22].
- Point-based approaches operate directly on raw point coordinates.

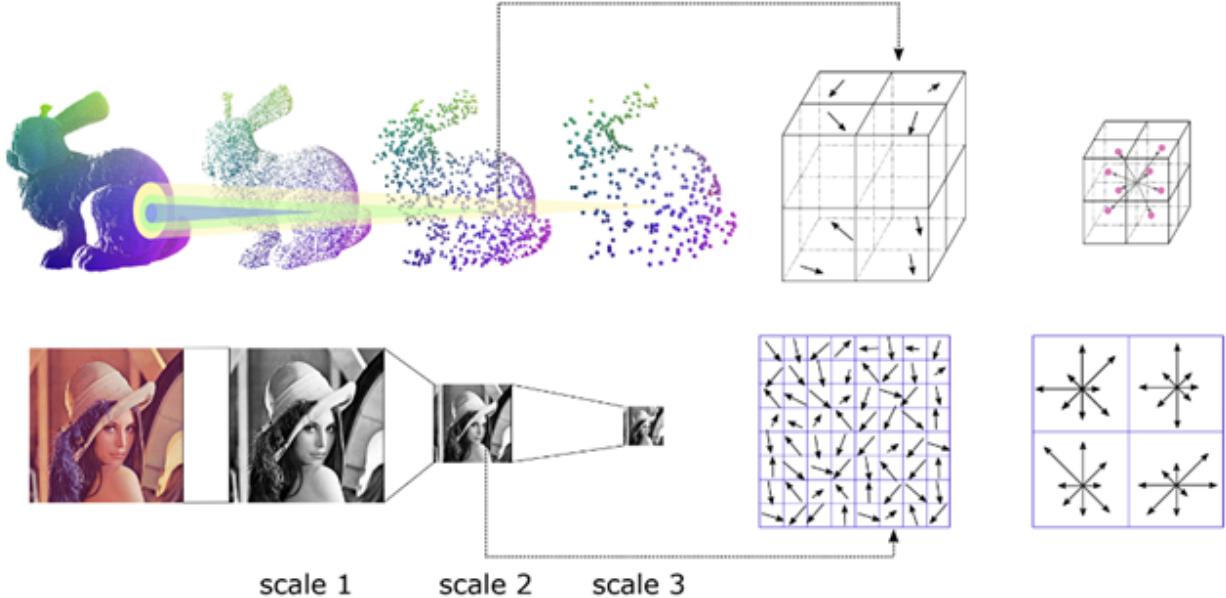
Projection-based methods transform point clouds into multi-view 2D images and apply 2D semantic segmentation networks. Although these methods leverage mature image-processing techniques and are relatively easy to implement, their segmentation accuracy is limited, and determining optimal viewpoint numbers and distributions is challenging.



**Figure 1.6** Converting point clouds into multi-view projections [23]

Point-based methods face challenges due to the unordered, irregular, and transformation-invariant nature of point clouds. Conventional convolutions cannot be directly applied.

- Qi et al. [25] introduced **PointNet**, which uses symmetric max-pooling functions to achieve order invariance and aggregate local features into global representations. A **T-Net** module learns transformation-invariant features.
- Qi et al. [26] extended this into **PointNet++**, introducing hierarchical feature learning for multi-scale feature extraction, significantly improving performance on non-uniform point clouds.
- Hu et al. [27] proposed **RandLA-Net**, balancing efficiency and accuracy through random sampling, local feature aggregation, hierarchical receptive-field expansion, and attention-based feature selection.
- Jiang et al. [28] introduced **PointSIFT**, inspired by the scale- and rotation-invariant SIFT operator. The method aggregates features from eight directions to encode shape patterns and learns adaptive scales.



**Figure 1.7** Comparison of PointSIFT and SIFT [28]

## 1.5 Comparison of Semantic Segmentation and Object Detection

Although deep-learning-based semantic segmentation is less computationally efficient than traditional object detection (due to encoder–decoder architectures and higher complexity), segmentation provides more accurate boundary extraction. For example, U-Net-based segmentation can closely match the elliptical shape of a crater, whereas YOLO-based detection yields only rectangular bounding boxes.

In navigation applications, object detection is preferred for real-time performance. However, in geological analysis—such as distinguishing primary and secondary craters to infer formation age—precise boundary modeling is essential, making semantic segmentation superior [29].

## 1.6 Summary

In this study, a Martian crater point-cloud dataset was constructed using HRSC DEM data, and multiple deep-learning point-cloud networks were trained and compared. The work focuses on analyzing how multi-scale crater variations affect model performance and how architectural differences influence segmentation results. The main contributions are as follows:

1. Construction of a Martian surface crater point-cloud dataset based on HRSC DEM.
2. Comparative evaluation of PointNet, PointNet++, and RandLA-Net to analyze their performance on multi-scale crater extraction.

3. Analysis of how different network designs address unorderedness, local–global feature learning, and transformation invariance in point-cloud data, and how these factors shape segmentation performance.

## 2 Model's Architecture

This chapter analyzes the structures and mechanisms of three classical point-cloud segmentation models—PointNet, PointNet++, and RandLA-Net—highlighting how each addresses the inherent characteristics of point-cloud data and how they handle multi-scale challenges.

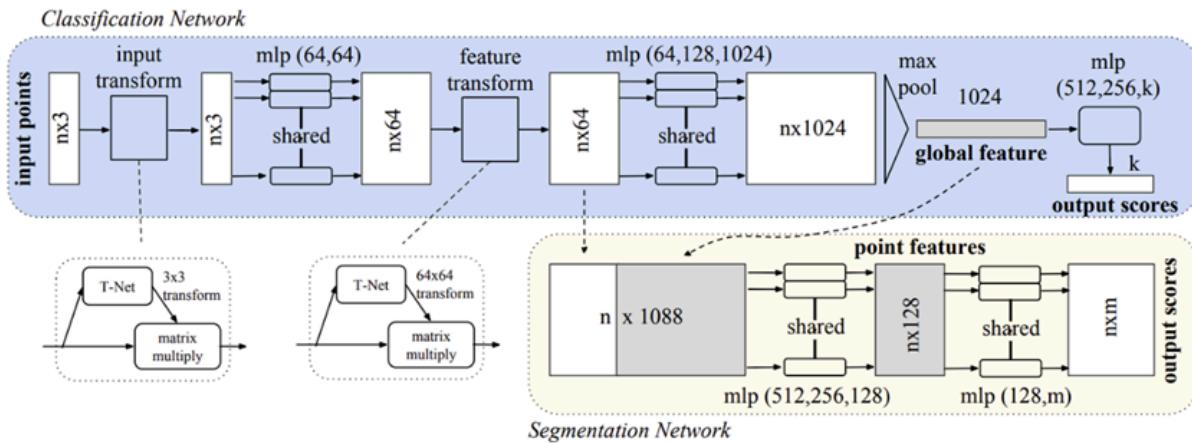
PointNet is designed to address unordered input, point-to-point interactions, and transformation invariance through symmetric functions and T-Net modules. PointNet++, as an extension of PointNet, introduces hierarchical structures to capture local features through multi-level grouping, sampling, and PointNet-based feature extraction. RandLA-Net employs random sampling and a novel local feature aggregation module to efficiently process large-scale point clouds while preserving essential geometric details.

### 2.1 PointNet Model

Point-cloud data exhibit three fundamental properties in deep learning:

(1) **Unorderedness**, (2) **Point interactions**, and (3) **Transformation invariance**.

PointNet incorporates three corresponding modules to address these properties while maintaining computational efficiency.



**Figure 2.1** Structure of the PointNet model [25]

The basic architecture of PointNet is shown in Fig. 2.1. The classification network (blue) and the segmentation network (yellow) share most layers. The classification network takes  $n$  points as input, applies feature transformations, aggregates per-point features via a max pooling layer, and outputs class scores for the entire point cloud. The segmentation network extends this by concatenating global and local features for each point and predicting a class score per point. The “MLP” layers use ReLU activation and batch normalization; the final MLP in the classification branch includes a dropout layer.

## Unorderedness and Symmetric Functions

To address unordered input, PointNet approximates a general function over a set of points using symmetric functions:

$$f(\{x_1, \dots, x_n\}) \approx g(h(x_1), \dots, h(x_n)) \quad (2.1)$$

Here  $f$  is the target function,  $h(\cdot)$  is a point-wise transform (MLP), and  $g(\cdot)$  is a symmetric function implemented via max pooling.

Other approaches—such as enforcing an input order or applying RNNs to all permutations—either fail to preserve spatial consistency or are computationally infeasible. Qi et al. [25] show that merely sorting the input provides negligible improvement, and RNNs cannot scale to large point sets.

## Point Interactions via Global–Local Fusion

PointNet models point interactions by concatenating the global feature vector with each point’s local feature. This fusion enables the network to jointly leverage local geometric cues and global semantic context. Experiments show that this design successfully captures surface normals and supports state-of-the-art performance in shape and scene segmentation.

## Transformation Invariance via T-Net

To achieve geometric transformation invariance, PointNet introduces the **T-Net**, inspired by the spatial transformer network of Jaderberg et al. [31]. T-Net predicts an affine transformation matrix that aligns input points to a canonical space. A second T-Net can be inserted in feature space to align learned features. Since the feature transformation matrix is high-dimensional, a regularization term is added:

$$L_{reg} = \| I - AA^T \|_F^2 \quad (2.2)$$

where  $A$  is the predicted feature-alignment matrix. This term encourages orthogonality, preserving information.

## 2.2 PointNet++ Model

The PointNet++ model is an extension of the PointNet model. On the basis of PointNet, it adds a hierarchical structure, enabling the network to robustly learn features from point-cloud data with non-uniform sampling densities.

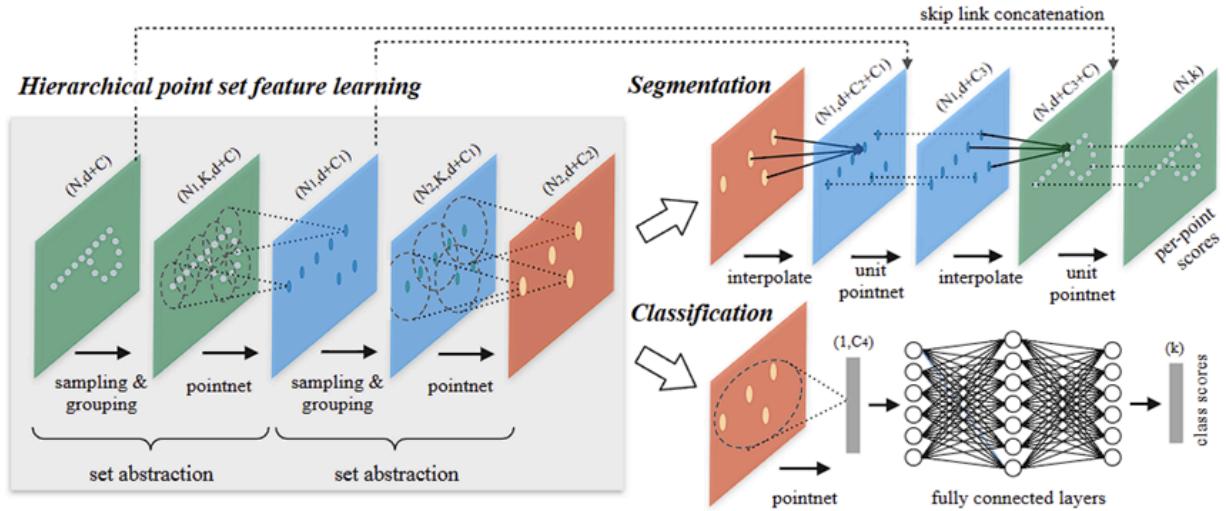


Figure 2.2 Structure of the PointNet++ model [26]

### 2.2.1 Hierarchical Learning of Point-Cloud Features

The original PointNet model aggregates features of the entire point cloud using a single max pooling layer. In contrast, the architecture of PointNet++ introduces hierarchical grouping of points and gradually abstracts larger local regions along the hierarchy. The hierarchical structure of PointNet++ is composed of several **set abstraction** modules. In each set abstraction module, some points are discarded, and the remaining points form the input of the next-level abstraction.

Each set abstraction module consists of three layers: (1) a **sampling layer**, (2) a **grouping layer**, and (3) a **PointNet layer**.

The sampling layer selects a set of points from the input point cloud and defines them as local centroids. The grouping layer constructs local regions around these centroids by searching for neighboring points. The PointNet layer applies a small PointNet network to each local region and outputs a feature vector for that local region.

A set abstraction module takes an input matrix of size  $N \times (d + C)$ , where  $N$  is the number of points,  $d$  is the dimensionality of the coordinates, and  $C$  is the dimensionality of the point feature vector. The output is a matrix of size  $N' \times (d + C')$ , where  $N'$  is the number of downsampled points, and  $C'$  is the dimensionality of the new feature vector.

### 2.2.2 Sampling Layer

For a given point set  $x_1, x_2, \dots, x_n$ , the sampling layer selects a subset  $x_{i_1}, x_{i_2}, \dots, x_{i_m}$  using **iterative farthest point sampling**. Here,  $x_{i_j}$  is the point that is farthest from the already selected set  $\{x_{i_1}, x_{i_2}, \dots, x_{i_{j-1}}\}$  among all unselected points. Compared with random sampling, iterative farthest point sampling provides better coverage of the entire point set for the same number of centroids. Unlike conventional CNNs that scan the vector space without considering the distribution of point clouds, iterative farthest point sampling allows the receptive field to be determined by the data itself.

### 2.2.3 Grouping Layer

The grouping layer takes as input a point set of size  $N \times (d + C)$ , where  $N$  is the number of points,  $d$  is the dimensionality of the coordinates, and  $C$  is the dimensionality of the feature vector. In addition, the centroid coordinates of size  $N' \times d$  are provided as input. The grouping layer outputs a tensor of size  $N' \times K \times (d + C')$ , where each of the  $N'$  groups corresponds to a local region, and  $K$  is the number of neighboring points associated with each centroid. Although the value of  $K$  may vary for different groups, the subsequent PointNet layer converts each group with a variable number of points into a fixed-size local feature vector.

In conventional CNNs, the pixels in a local neighborhood are determined by the convolution kernel size (Manhattan distance). In a metric space for point clouds, the points in a local neighborhood are determined by a defined distance metric.

A **ball query** finds all points within a sphere of a given radius around the query point (up to a maximum number). The **k-nearest neighbors (kNN)** method finds a fixed number of nearest points to the query point. Compared with kNN, ball query ensures a fixed spatial extent of the local neighborhood, resulting in better spatial consistency of local features, which is more suitable for local pattern recognition tasks.

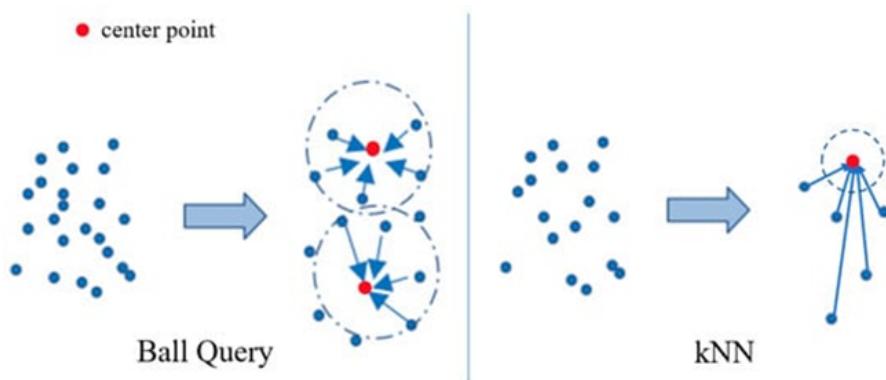


Figure 2.3 Ball query vs. kNN [32]

### 2.2.4 PointNet Layer

The PointNet layer takes  $N'$  local point sets as input, each of size  $K \times (d + C')$ , so the overall input is of size  $N' \times K \times (d + C')$ . The output of each local point set is abstracted into a local feature associated with its centroid and neighboring points, giving an output of size  $N' \times (d + C')$ .

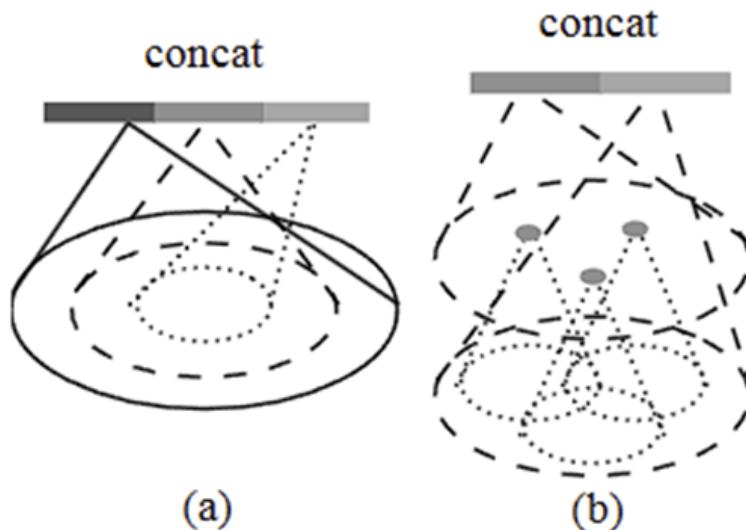
Within each local region, the coordinates of points are first transformed into a coordinate frame relative to the centroid. Then a PointNet network is used as the basic architecture for local pattern learning. By using relative coordinates along with point features, the model can capture relationships between points within each local region.

### 2.2.5 Robust Feature Learning under Non-Uniform Sampling Density

In point-cloud processing, it is very common that points in different regions exhibit different sampling densities, and such non-uniform distributions pose challenges for feature learning on point sets. Features learned from dense regions may not generalize well to sparse regions; thus, models trained mainly on sparse point clouds may struggle to recognize fine-grained local structures.

Ideally, the model should capture as many fine details as possible in densely sampled regions. However, in sparse regions, local patterns may suffer from undersampling, making it difficult for the model to learn detailed features. To compensate for this, it becomes necessary to look for larger-scale patterns over a wider area.

The PointNet++ model proposes a **density-adaptive PointNet layer**, which learns how to combine features from regions of different scales under varying sampling densities. In PointNet++, each set abstraction module extracts local patterns at multiple scales and adaptively combines them based on local point density. According to how local regions are grouped and how features at different scales are combined, two types of density-adaptive layers are defined.



*Figure 2.4 (a) MSG and (b) MRG: two types of density-adaptive layers [26]*

### (1) Multi-Scale Grouping (MSG)

As shown in Fig. 2.4(a), a simple and effective way to capture patterns at different scales is to apply multiple grouping operations with different radii and then apply a PointNet to each scale to extract features. The resulting multi-scale features are concatenated to form a multi-scale representation. During training, randomly dropping some input points is used as a regularization strategy to optimize the way the network combines multi-scale features.

### (2) Multi-Resolution Grouping (MRG)

In MSG, a large local PointNet must be run for each centroid at every scale, which leads to high computational cost, especially at the lowest level where the number of centroids is very large. MRG provides an alternative approach that avoids excessive computation while maintaining the ability to adaptively aggregate information based on the distribution of points.

As shown in Fig. 2.4(b), the spatial feature at a given level is formed by concatenating two vectors. One vector (left in the figure) summarizes features of subregions from the lower level via the abstraction hierarchy. The other vector (right in the figure) is obtained by applying a single PointNet directly to all original points in the local region.

In sparse regions, the second vector may be more reliable, because the subregions used for computing the first vector are more sparsely sampled and more affected by undersampling. In this case, the second vector should be given higher weight. Conversely, in dense regions, the first vector provides richer detail, as it recursively perceives higher-resolution patterns at lower levels. Compared with MSG, MRG avoids computing features over large neighborhoods at the lowest level, thus achieving higher computational efficiency.

## 2.2.6 Point Feature Propagation in Set Segmentation

In set abstraction modules, the original point set is progressively downsampled. However, in semantic segmentation tasks, point-wise features are required for all original points. One way to achieve this is to sample all original points as centroids in each set abstraction layer, but this would result in extremely high computational cost. Another approach is to **propagate features** from the downsampled points back to the original points.

PointNet++ adopts a hierarchical feature propagation strategy based on distance interpolation and skip connections (see Fig. 2.5). During feature propagation, PointNet++ propagates point features of size  $N_l \times (d + C)$  to a set of size  $N_{l-1}$ , where  $N_{l-1}$  and  $N_l$  are the input and output point set sizes of the  $l$ -th set abstraction layer, respectively. The propagation is achieved by interpolating the feature values of the  $N_l$  points at the coordinates of the  $N_{l-1}$  points.

PointNet++ uses inverse distance weighted averaging based on kNN for interpolation. The interpolated features at the  $N_{l-1}$  points are then concatenated with the skip-connected features

from the corresponding set abstraction layer. Finally, the concatenated features are passed through a single PointNet (with shared fully connected layers and ReLU) to update each point's feature vector, which is analogous to point-wise convolution in CNNs. This process is repeated until features are propagated back to the original point set. The entire propagation can be written as:

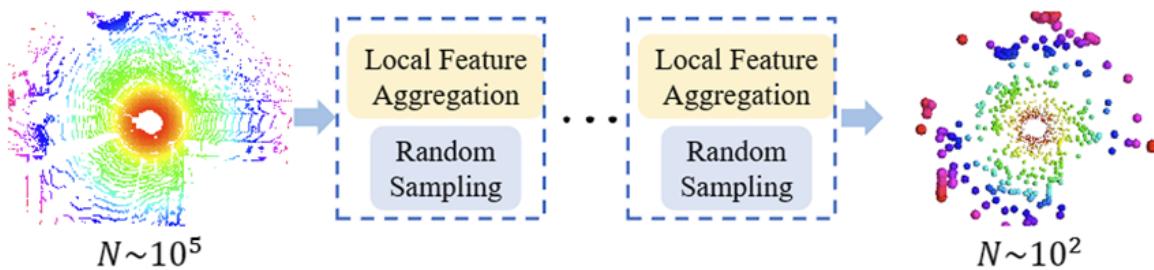
$$f^{(j)}(x) = \sum_{i=1}^k w_i(x)f_i^{(j)}/\sum_{i=1}^k w_i(x) \quad (2.3)$$

## 2.3 RandLA-Net Model

Although the PointNet model uses multilayer perceptrons to efficiently learn features for each point, it is difficult for PointNet to capture broader global information for each point. While it can obtain rich local information, it is essentially limited to relatively small point sets and cannot be directly scaled to very large point clouds. The main reasons are:

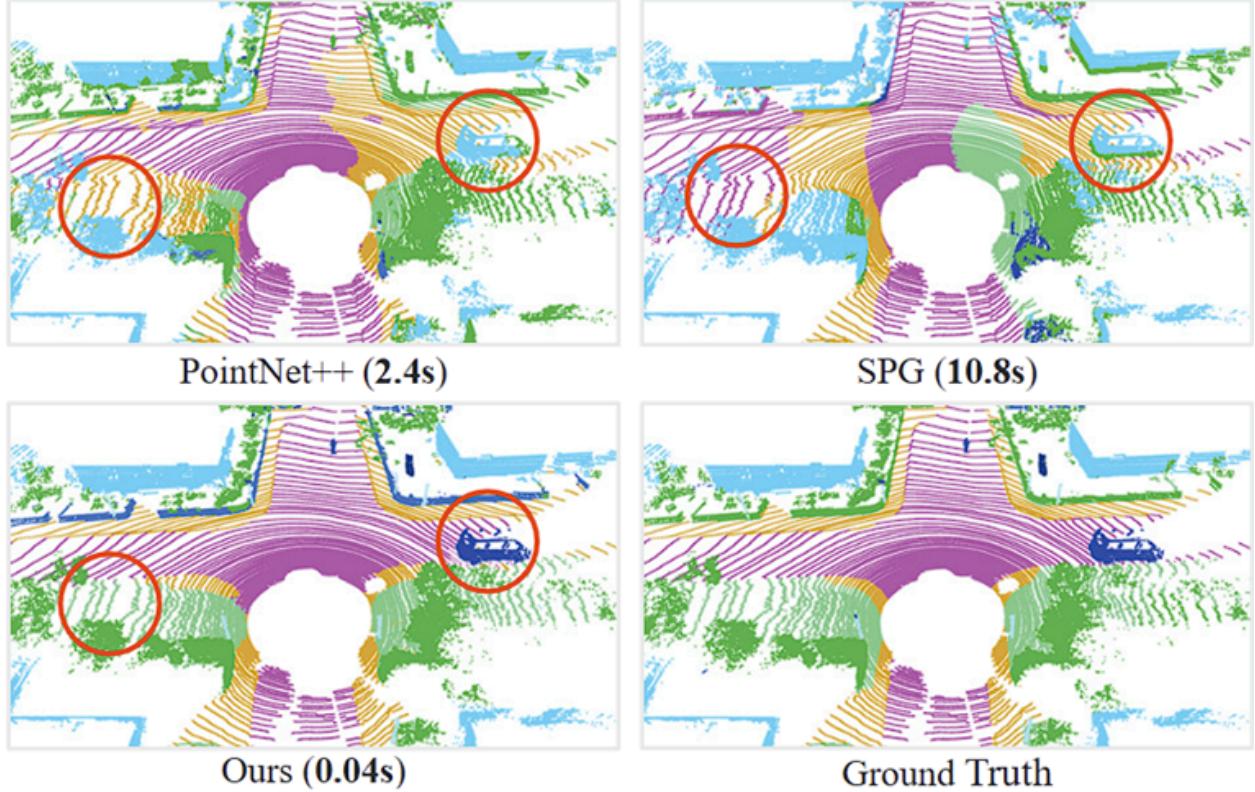
- (1) Point sampling can be computationally expensive or memory-inefficient;
- (2) Local feature learners often rely on expensive kernelization or graph construction, making them unsuitable for large numbers of points;
- (3) Local feature learners may fail to capture complex structures in large-scale point clouds due to limited receptive fields.

Hu et al. [27] proposed the RandLA-Net model, which can directly process large-scale point-cloud data without any pre-processing or post-processing steps. RandLA-Net is based on random sampling and a novel local feature aggregation module. By progressively enlarging the receptive field, it preserves the complex geometric structures in point clouds.



*Figure 2.5 Large-scale point-cloud data are downsampled at every layer of RandLA-Net, yet sufficient features are preserved [27]*

For large-scale point clouds, it is crucial in deep neural networks to preserve sufficient information while downsampling. RandLA-Net uses a simple and fast random sampling strategy to greatly reduce point density and, at the same time, designs a local feature aggregator to retain salient features, thus achieving a balance between efficiency and effectiveness.



*Figure 2.6 Performance of RandLA-Net (bottom left) on public datasets, achieving state-of-the-art trade-off between accuracy and efficiency [27]*

### 2.3.1 Random Sampling

Random sampling is a heuristic sampling strategy that uniformly selects points from the original  $N$  points. Its computational complexity is  $O(1)$ , independent of the total number of input points, i.e., constant time. Therefore, it is highly scalable and can achieve extremely high computational efficiency regardless of the size of the input point cloud. However, random sampling may cause many useful point features to be discarded. To address this issue, RandLA-Net designs a special local feature aggregation module to mitigate feature loss.

### 2.3.2 Local Feature Aggregation

As shown in Fig. 2.7, the local feature aggregation module of RandLA-Net is applied in parallel to each point and consists of three neural components:

- (1) **Local Spatial Encoding (LocSE) block,**
- (2) **Attentive Pooling block, and**
- (3) **Dilated Residual block.**

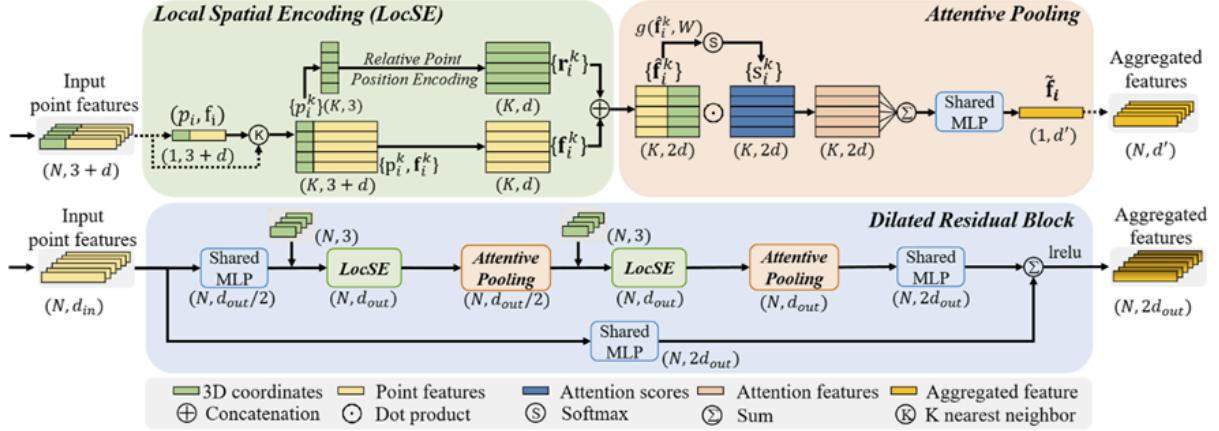


Figure 2.7 Structure of the RandLA-Net local feature aggregation module [27]

### (1) Local Spatial Encoding Block

Given a point cloud  $P$  and a feature vector for each point (e.g., original RGB or intermediate learned features), the local spatial encoding block explicitly embeds the  $X$ ,  $Y$ ,  $Z$  coordinates of all neighboring points so that point features are always aware of their relative spatial locations. This allows the block to effectively model local geometric information and helps the network learn complex local structures in large-scale point clouds.

Concretely, for each point, its neighboring points and corresponding Euclidean distances are determined via kNN. The coordinates of the center point, the neighboring points, and the Euclidean distances are encoded, and the encoded features of all neighbors are concatenated to obtain an enhanced local spatial feature. This feature explicitly encodes the local geometric structure around the center point.

### (2) Attentive Pooling Block

The attentive pooling block processes the enhanced local spatial features output by the local spatial encoding block. Unlike PointNet, which uses max pooling to rigidly aggregate neighboring features (resulting in substantial information loss), the attentive pooling block uses an attention mechanism to automatically learn important local features, thus avoiding excessive information loss during feature aggregation.

For a given local feature set  $F_i = \{f_{1,i}, f_{2,i}, \dots, f_{3,i}\}$ , a function  $g(\cdot)$  is designed to learn a unique attention score for each feature. The function  $g(\cdot)$  is implemented as a shared MLP followed by a softmax function. The learned attention scores can be regarded as a soft mask that automatically selects important features.

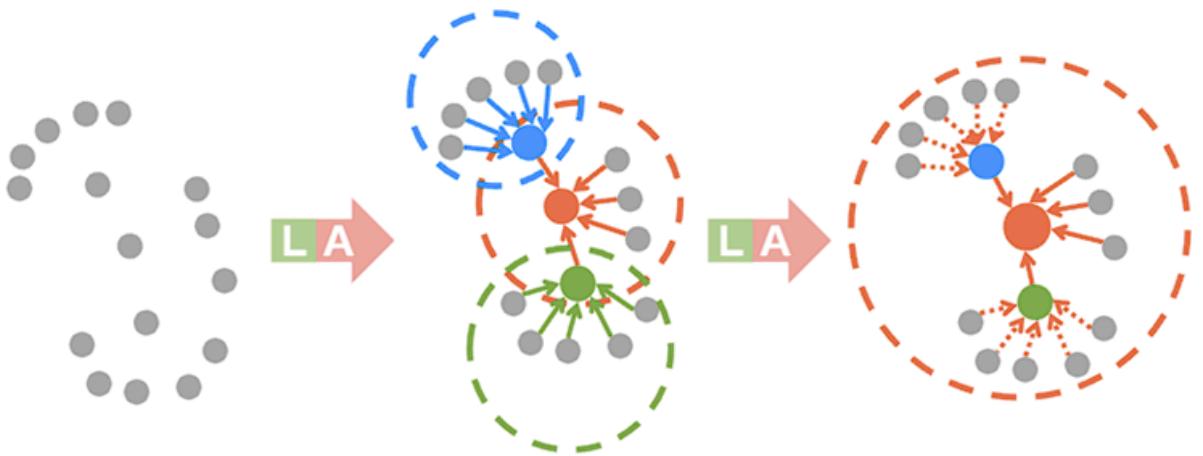
In summary, for a given input point cloud  $P$ , the local spatial encoding and attentive pooling blocks in RandLA-Net learn to aggregate geometric patterns and features from each point's  $k$  nearest neighbors, ultimately generating an informative feature vector.

### (3) Dilated Residual Block

Since large-scale point clouds are heavily downsampled, it is necessary to significantly enlarge each point's receptive field in order to preserve fine geometric details. As shown in Fig. 2.7, multiple local spatial encoding blocks and attentive pooling blocks are connected with skip connections to form a **dilated residual block**.

In Fig. 2.8, the red point selects  $k$  neighbors after the first pass through the local spatial encoding and attentive pooling blocks, and can receive information from up to  $k^2$  neighbors after the second pass. This provides a very simple way to expand the receptive field and propagate features over a larger effective region.

The more such units are stacked, the larger the receptive field becomes; however, this also reduces overall computational efficiency and increases the risk of overfitting. Therefore, in RandLA-Net, two sets of local spatial encoding and attentive pooling blocks are stacked to achieve a balance between efficiency and performance.



*Figure 2.8 How the dilated residual block increases each point's receptive field (dashed circles); colored points denote aggregated features; L denotes the local spatial encoding block; A denotes the attentive pooling block [27]*

Overall, the local feature aggregation module in RandLA-Net effectively preserves the complex local spatial structures of large-scale point clouds by explicitly modeling neighboring geometric structures and significantly enlarging the receptive field. At the same time, it is composed only of feed-forward MLPs, thus maintaining good computational efficiency.

## 2.4 Comparison Between RandLA-Net and the PointNet Family

The RandLA-Net model and the PointNet family differ mainly in two aspects of structural design: **multi-scale handling** and **sampling strategy**.

For multi-scale problems, the PointNet model does not have an explicit multi-scale processing mechanism; it applies the same operations to all points without distinguishing between global and local features. As a result, it cannot fully capture both large-scale global geometric structures and small-scale geometric details, leading to limited performance in multi-scale crater extraction on the Martian surface.

In contrast, the PointNet++ model addresses craters of different scales through multi-level farthest point sampling and grouping. In each layer, a small PointNet network is used to extract geometric features, and local geometric features at different scales are gradually aggregated to form global features that can adapt to different scales.

RandLA-Net achieves multi-scale processing through random sampling and local feature aggregation. By multi-level feature aggregation, it captures features at different scales; through neighborhood search and local feature encoding, it ensures that each sampled point can capture sufficient local spatial information. Furthermore, the introduction of an attention mechanism allows the feature aggregation process to be more flexible and dynamically adjusts weights according to point features. Finally, through hierarchical aggregation, RandLA-Net progressively extracts features at different scales to form a complete multi-scale representation.

Regarding sampling, the PointNet model, due to its relatively small computational cost, does not adopt any special sampling algorithm and applies the same computations to all points. The PointNet++ model uses farthest point sampling combined with kNN or ball query for grouping, progressively reducing the number of points to decrease the computational load and to handle large-scale point clouds; its drawback is the relatively high computational cost of the sampling process.

The RandLA-Net model uses random sampling, which has the advantage of low computational cost. However, random sampling may lead to insufficient sampling in some important regions, causing the model to miss critical geometric features. To address this, RandLA-Net designs the local feature aggregation module to compensate for the potential undersampling problem introduced by random sampling, enabling RandLA-Net to effectively perform multi-scale feature extraction and high-precision segmentation on large-scale point-cloud data, while achieving a balance between accuracy and efficiency.

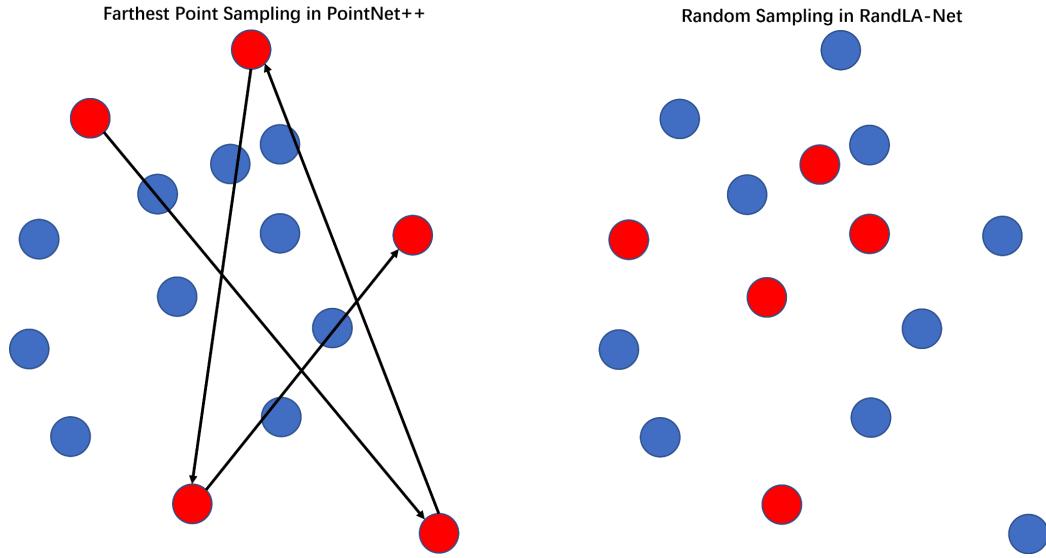


Figure 2.9 Farthest point sampling in PointNet++ vs. random sampling in RandLA-Net

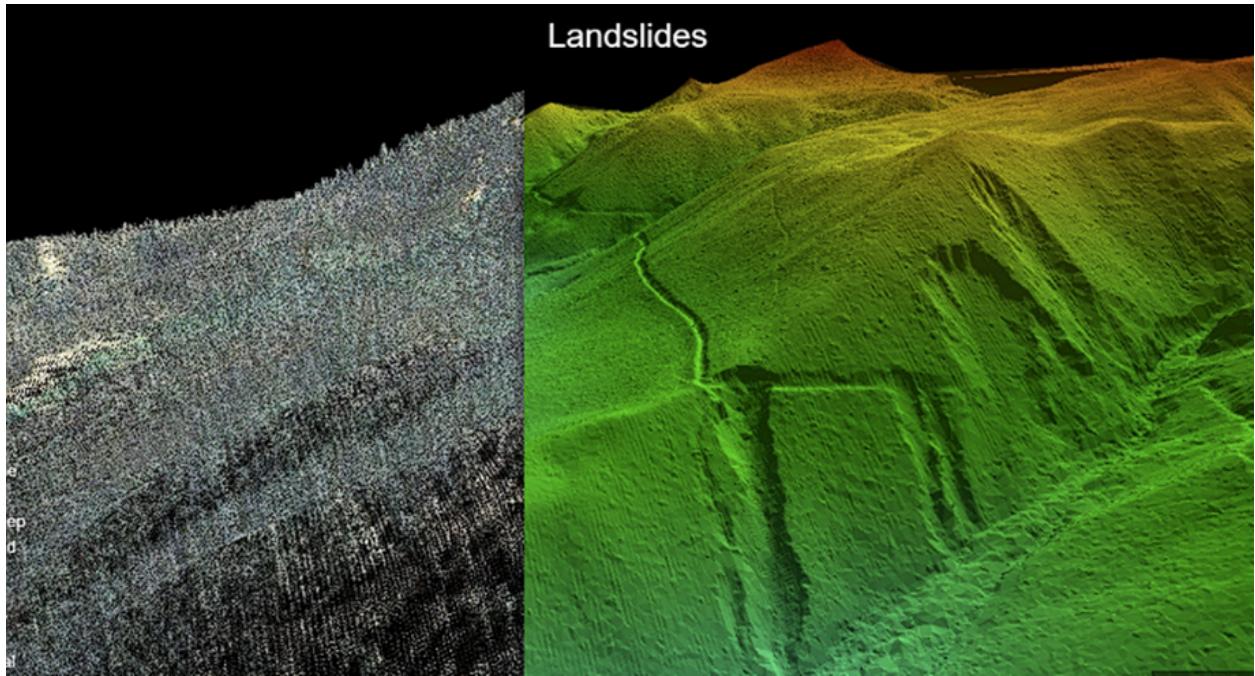
## 3 Research Methods

### 3.1 Dataset

The dataset used in this study is the High Resolution Stereo Camera Digital Elevation Model (HRSC DEM) released by the European Space Agency (ESA). The data were acquired by the Mars Express mission, which obtained Martian surface topography and elevation using stereo-photogrammetric techniques (<https://ode.rsl.wustl.edu/mars/productsearch>).

This study selects six regions as the research areas: **Mawrth Vallis, Kasei Valles, Nili Fossae, McLaughlin, Oyama Crater, and Tyrras Vallis**. The downloaded data are 2.5D DEM data, which must be converted into 3D point-cloud data for subsequent processing.

In 2.5D data, the height  $z$  is a function of the planar coordinates  $(x, y)$ , i.e., each  $(x, y)$  corresponds to a single  $z$ . In contrast, in 3D point-cloud data, the same  $(x, y)$  can correspond to multiple elevation values  $z$ . Therefore, when DEM data are converted to point clouds, the resolution in the  $z$ -direction is lost. However, this loss is acceptable for the Martian crater extraction problem, because there are almost no vertical surfaces with a slope of 90 on the Martian surface and within craters, meaning that each  $(x, y)$  can be reasonably assumed to correspond to only one elevation  $z$ .

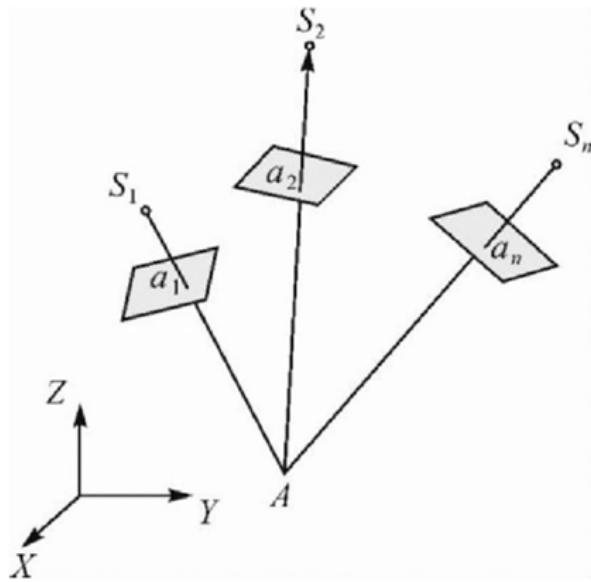


*Figure 3.1 3D point-cloud data (left) vs. 2.5D DEM data (right)*  
[\(<https://www.usgs.gov/media/images/lidar-point-cloud-vs-bare-earth-dem>\)](https://www.usgs.gov/media/images/lidar-point-cloud-vs-bare-earth-dem)

The original dataset includes 8-bit orthorectified images for the nadir channel and four color channels, as well as 16-bit DTMs (with 1 m vertical resolution). Since the orthorectified images are generated from the DTMs, they are only valid within the coverage of the DTMs.

The DTMs are generated based on multi-image matching using a pyramidal least-squares correlation algorithm. Before that, stereo images are pre-processed with adaptive Gaussian low-pass filtering to reduce the impact of image compression. Three-dimensional points are determined using least-squares forward intersection, followed by DTM grid interpolation. Using the adjusted orbits and pointing data, the DTMs are then used to generate orthorectified images. The only additional pre-processing step is a histogram-based linear contrast stretch, which does not alter the linear radiometric calibration of the images.

Based on the high-resolution DTM, quality assessment is performed against the Mars Orbiter Laser Altimeter (MOLA) DTM, and the final exterior orientation is refined to improve the dataset.



*Figure 3.2 Determination of 3D points using forward intersection [33]*

The selected study areas are mainly regions with abundant data and dense cratered terrain. The relevant parameters are listed in Table 3.1.

**Table 3.1 Parameters of the study areas**

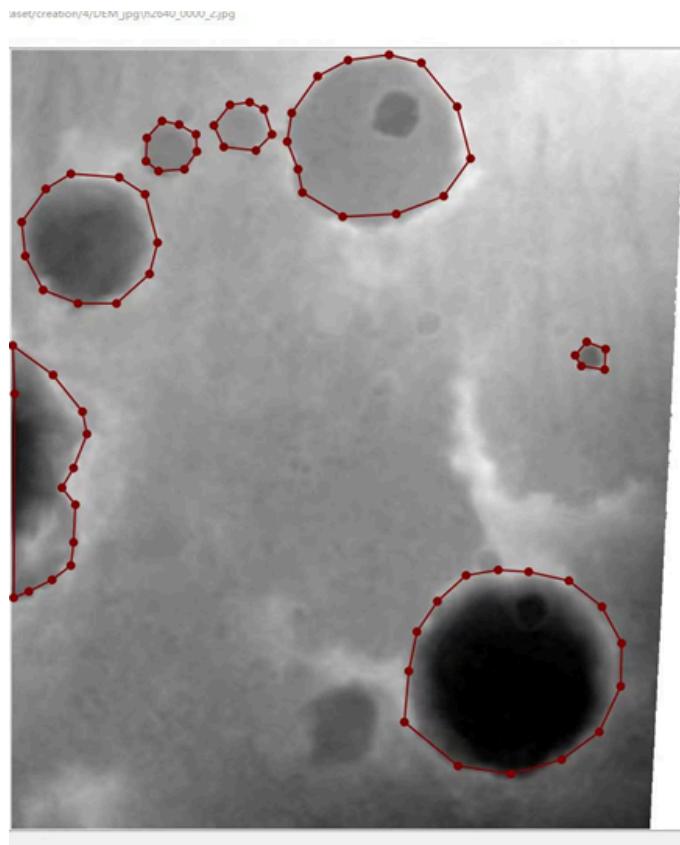
ID	Name	Latitude	Longitude	Length (radius) / km
1	Mawrth Vallis	22.3°N	343.5°E	636
2	Kasei Valles	24.6°N	65.0°E	1600
3	Nili Fossae	22.6°N	76.8°E	667
4	McLaughlin	21.9°N	337.6°E	91
5	Oyama Crater	20.5°N	31.5°E	79
6	Tyrras Vallis	8.4°N	50.2°E	150

## 3.2 Methodology

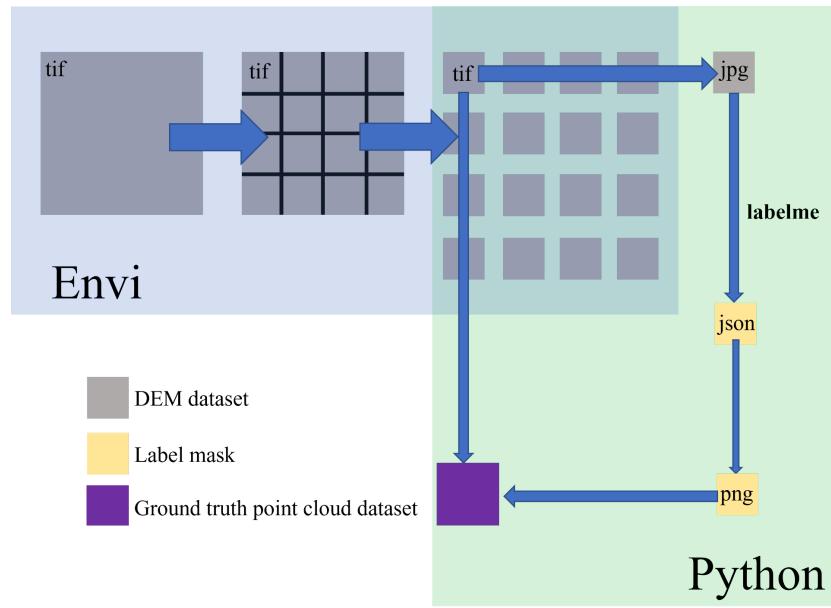
### 3.2.1 Data Preprocessing

The original DEM data downloaded from ESA are in TIFF format and are first opened in ENVI. All TIFF images are then converted to JPG format. Using the **Labelme** library in Python, visual

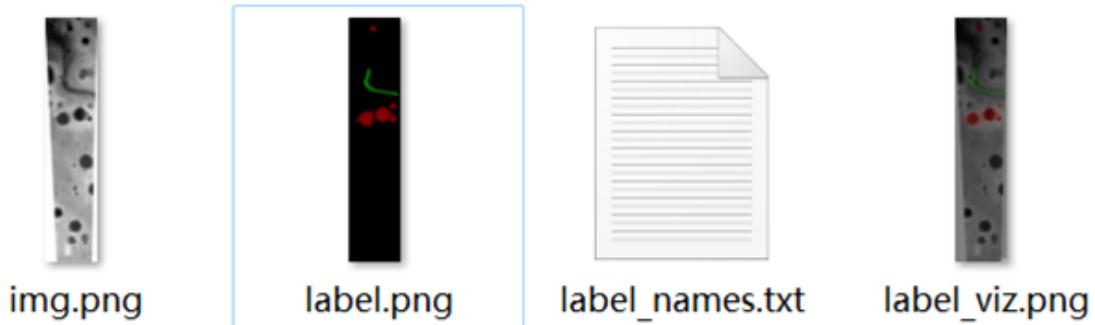
annotation is performed on the images to create masks for all craters, and the corresponding JSON files are generated. The JSON mask files are then converted to PNG format. Finally, the PNG masks and DEM images are combined to generate labeled point-cloud files.



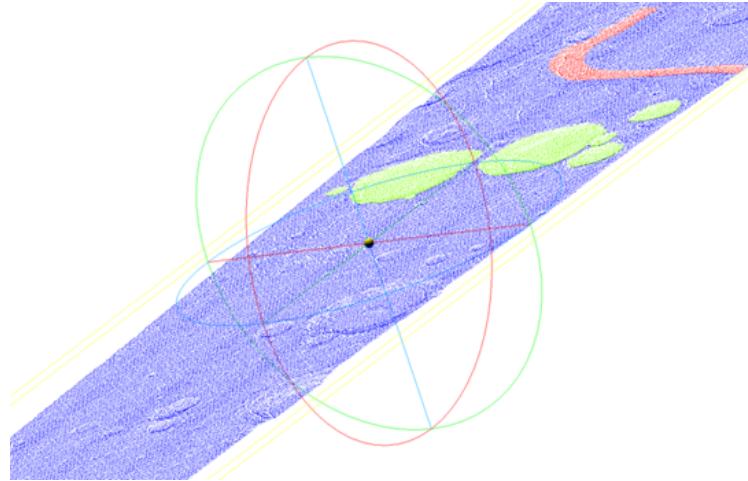
*Figure 3.3 Visual annotation of impact craters on images*



*Figure 3.4 Data preprocessing workflow*



*Figure 3.5 Each sample consists of DEM data, mask, and class labels*



*Figure 3.6 Labeled point-cloud data*

### 3.2.2 Model Training

The preprocessed point-cloud data are fed into the three models: **PointNet**, **PointNet++**, and **RandLA-Net**. For each sample, 80,000 points are uniformly sampled.

All three models use the **Adam optimizer** with an initial learning rate of 0.001, where  $\beta_1 = 0.9$  and  $\beta_2 = 0.999$ . The **cross-entropy loss** is used as the loss function, defined as:

$$L_{ce} = -\frac{1}{N} \sum_{i=1}^N y_i \log(p_i) \quad (3.1)$$

where  $N$  is the number of samples,  $y_{ii}$  is the true label, and  $p_i$  is the predicted probability.

The batch size for all models is set to 8. PointNet and PointNet++ are trained for 250 epochs, and RandLA-Net is trained for 150 epochs.

## 3.3 Experimental Design

*Figure 3.7 Overall experimental workflow*

### 3.3.1 Training Procedure

The training procedure is illustrated in Fig. 3.7. First, the total of 235 samples in the dataset are split into **training, test, and validation sets** with a ratio of **170 / 55 / 10**. The training samples are then separately fed into the PointNet, PointNet++, and RandLA-Net networks. After adjusting hyperparameters, the final results and confusion matrices are obtained, visualized, and evaluated in terms of accuracy metrics.

### 3.3.2 Confusion Matrix and Accuracy Evaluation

The confusion matrix is a commonly used evaluation tool in classification tasks. It provides an intuitive way to compare model predictions with ground truth. In this study, the confusion matrix is used not only to compute evaluation metrics, but also to visualize results in the point-cloud samples and compare them against ground truth and predictions from different models. The definitions are given in Table 3.2.

**Table 3.2 Definition of the confusion matrix**

Actual \ Predicted	Positive (positive)	Negative (negative)
Positive (positive)	True Positive (TP)	False Negative (FN)
Negative (negative)	False Positive (FP)	True Negative (TN)

The **Intersection over Union (IoU)** is a key metric in image segmentation and object detection. It measures the degree of overlap between the prediction and the ground truth, i.e., the similarity between the predicted region and the true region. A higher IoU indicates a higher overlap and better segmentation performance. It is defined as:

$$\text{IoU} = \frac{TP}{TP+FP+FN} \quad (3.2)$$

**Accuracy** is an important metric for evaluating classification models. It represents the proportion of correctly predicted samples among all samples in the test set. Accuracy is simple to compute and is effective for datasets with relatively balanced class distributions. A higher accuracy indicates better overall performance. It is defined as:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN} \quad (3.3)$$

**Precision** is another important evaluation metric. It represents the proportion of correctly predicted positive samples among all samples predicted as positive. Precision reflects the reliability of positive predictions. A higher precision means fewer false positives, which is particularly important in tasks that emphasize avoiding false alarms. It is defined as:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (3.4)$$

**Recall**, also known as sensitivity, is an important metric that represents the proportion of actual positive samples that are correctly predicted as positive. A higher recall indicates that the model can detect more positive samples and miss fewer true positives. This is especially suitable for scenarios where it is critical to find as many positive samples as possible. It is defined as:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3.5)$$

The **F1-Score** is an important metric that is the harmonic mean of precision and recall. It provides a balanced evaluation when dealing with imbalanced datasets. A higher F1-Score indicates that the model achieves a better trade-off between precision and recall and thus has better overall performance. It is defined as:

$$\text{F1-Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (3.6)$$

## 4 Results and Discussion

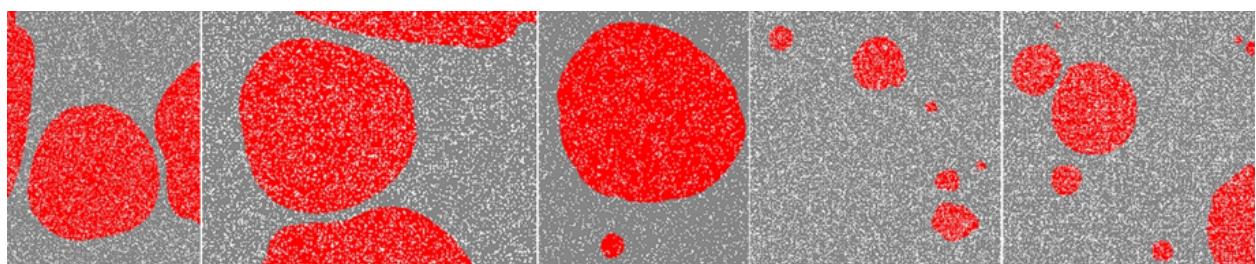
### 4.1 Prediction Results

The accuracy evaluation of the model prediction results is shown in Table 4.1.

**Table 4.1 Prediction results of the three models**

Metric	PointNet (epoch = 250)	PointNet++ (epoch = 250)	RandLA-Net (epoch = 150)
Accuracy	0.8869	0.9195	0.9283
Precision	0.7892	0.9042	0.8510
Recall	0.9455	0.8739	0.9247
F1-Score	0.8603	0.8888	0.8828
mIoU	0.7713	0.8147	0.8995

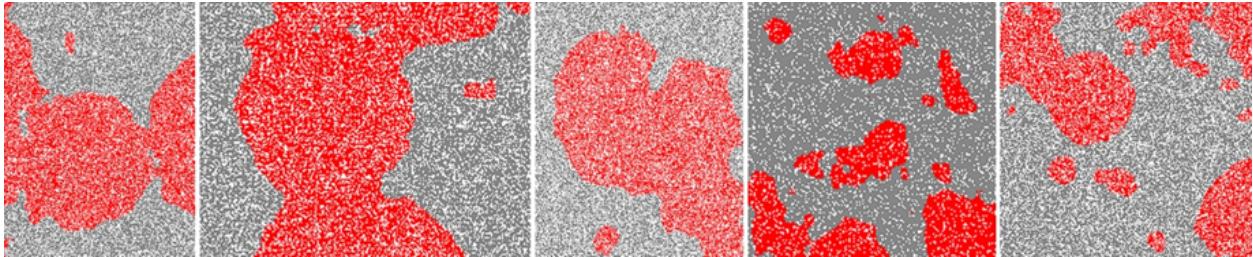
To compare model performance, especially their ability to extract Martian impact craters at different scales, five test samples are selected as shown in Fig. 4.1, where the second image is a rotated and zoomed version of the first image.



*Figure 4.1 Ground truth of test samples (red: craters; gray: background)*

#### 4.1.1 PointNet Model Prediction Results

The prediction results of the PointNet model are shown in Fig. 4.2.



*Figure 4.2 Prediction results of the PointNet model*

Comparing the first two predicted images with the ground truth, the PointNet network can roughly segment the complete crater in the center and the two half craters on both sides. After rotation and scaling, there is no obvious difference, indicating that the PointNet model effectively handles transformation invariance for point-cloud data. With relatively low computational cost, the small T-Net subnetwork can predict an affine transformation matrix and thus realize geometric transformation invariance for the point cloud.

In the third image, when facing two craters with a noticeable size difference, PointNet successfully extracts the smaller crater. However, for the larger crater, PointNet not only fails to fully extract the upper-right portion of the crater, but also incorrectly classifies the background in the lower-right corner as crater. This shows that the PointNet network performs poorly when dealing with large-scale craters.

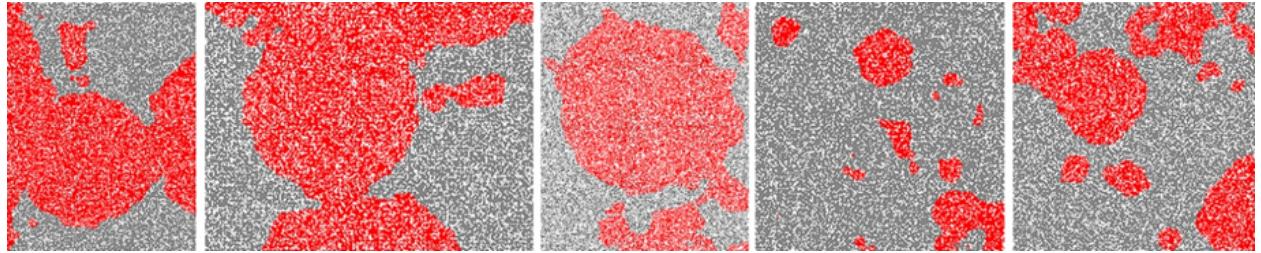
In the fourth and fifth images, PointNet can basically extract all craters, but apart from the smaller ones, the edges of larger craters are often misclassified, with background around the crater edges mistakenly labeled as crater, and this misclassification is quite serious.

Compared with PointNet++ and RandLA-Net, the PointNet model has a relatively high **Recall** and a relatively low **Precision**. This indicates that PointNet has a high detection rate for craters but also produces many false positives (FP). Possible reasons include: the model tends to overfit and exhibits high variance, or the training data are highly imbalanced, causing PointNet to bias toward predicting background as crater. Most false positives occur around the edges of large craters, which is closely related to the fact that PointNet does not have an explicit design for handling objects at different scales.

The relatively low **Accuracy** and **F1-Score** indicate that, overall, the PointNet model has poorer performance than the other two models. Although the model performs well in extracting craters themselves, its overall performance on the entire test set is unsatisfactory. Similarly, the low **mIoU** combined with high **Recall** suggests that while the model successfully detects many crater regions, the predicted regions are not accurate enough, leading to suboptimal overall performance.

#### 4.1.2 PointNet++ Model Prediction Results

The prediction results of the PointNet++ model are shown in Fig. 4.3.



*Figure 4.3 Prediction results of the PointNet++ model*

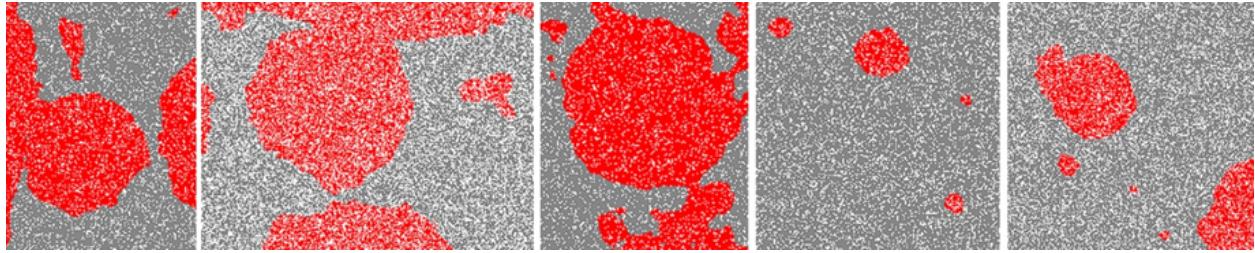
PointNet++ is an extension of PointNet with added hierarchical structure, but it does not introduce major changes in handling geometric invariance. Therefore, when processing rotated or scaled data, its performance is similar to that of PointNet. In the first and second images, the overall behavior is similar to PointNet, and the model shows good performance on data-augmented samples.

In the third image, when facing a large crater and a small crater on the Martian surface, the PointNet++ model accurately segments the larger crater. However, for the smaller crater, although it correctly classifies the crater itself, it also incorrectly labels the background around the small crater as part of the crater. A similar problem occurs around the larger crater, though less prominently than around the smaller crater. In the upper-right portion of the image, some background is also mistakenly classified as a new crater. This tendency to misclassify background as crater near crater boundaries is also observed in the fourth and fifth images. For larger-scale craters, the boundary extraction is acceptable, and the circular outline can be roughly recognized. However, for smaller-scale craters, the boundaries become indistinguishable.

Compared with PointNet and RandLA-Net, the PointNet++ model has higher **Precision** and **F1-Score** but much lower **Recall**. This suggests that the model is conservative: the craters predicted by PointNet++ are highly reliable, but it only detects a subset of all craters, preferring to miss some craters rather than reduce prediction accuracy. This behavior may be related to class imbalance in the dataset, where large amounts of background and relatively few crater pixels lead the model to favor background. A new loss function or strategy should be designed such that missed craters (FN) are penalized more heavily, while false positives (FP) are penalized less, to achieve a better balance between Precision and Recall.

#### 4.1.3 RandLA-Net Model Prediction Results

The prediction results of the RandLA-Net model are shown in Fig. 4.4.



*Figure 4.4 Prediction results of the RandLA-Net model*

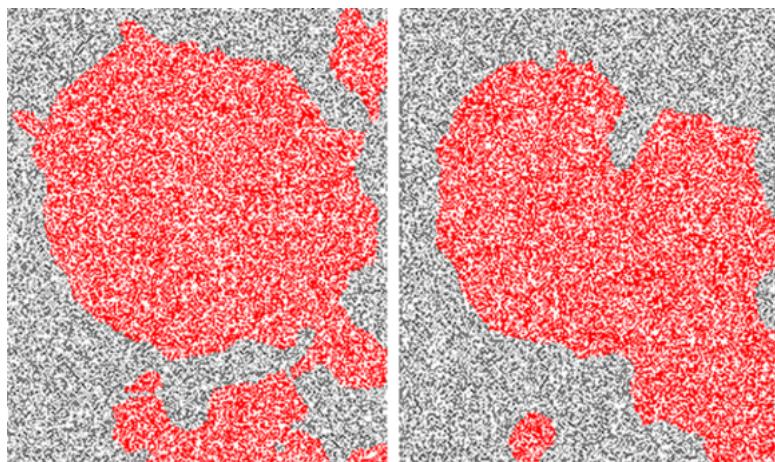
The RandLA-Net model benefits from its unique local feature aggregation blocks, feature enhancement, and attention mechanisms, which help it learn geometric information from the input point cloud. In the first and second images, RandLA-Net exhibits transformation invariance comparable to that of PointNet and PointNet++. Furthermore, it separates the central crater from part of the crater on one side and successfully identifies the background between the two craters.

In the fourth and fifth images, RandLA-Net performs better in distinguishing background around craters. Although it fails to correctly detect two small craters, the background around each crater is seldom misclassified as crater.

Compared with the other two models, RandLA-Net achieves the highest **Accuracy** and **mIoU**, and ranks second in the other three metrics. This indicates that RandLA-Net achieves the best overall performance and precision among the three models. Although it does not rank first in Precision, F1-Score, or Recall individually, it is the most balanced model and has the best overall classification accuracy. This advantage can be attributed to the design of its local feature aggregation module, which enables RandLA-Net to extract local features that better reflect geometric information and thus perform particularly well in detailed regions.

## 4.2 Comparison Between Different Models

### 4.2.1 PointNet vs. PointNet++

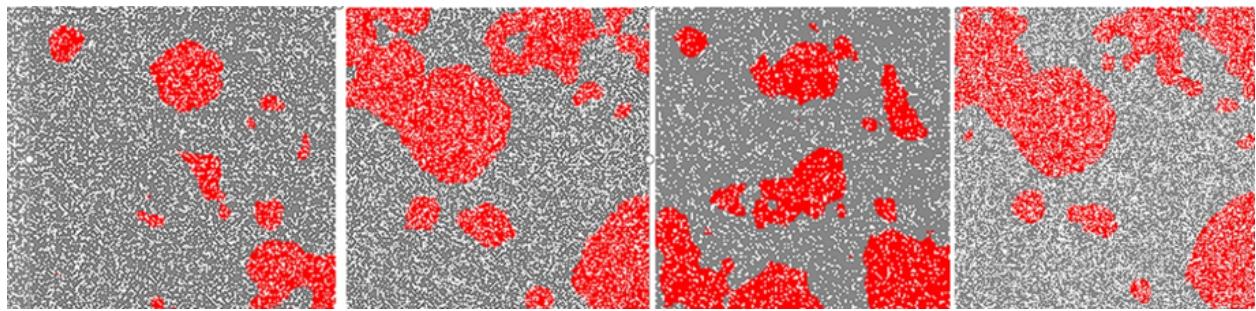


*Figure 4.5 Comparison of prediction results: PointNet++ (left) vs. PointNet (right)*

Compared with PointNet, the main improvement in PointNet++ lies in its hierarchical structure. By iteratively extracting features from local regions, local point sets can learn large-scale features. With density-adaptive feature extraction, PointNet++ can learn geometric features more efficiently and robustly.

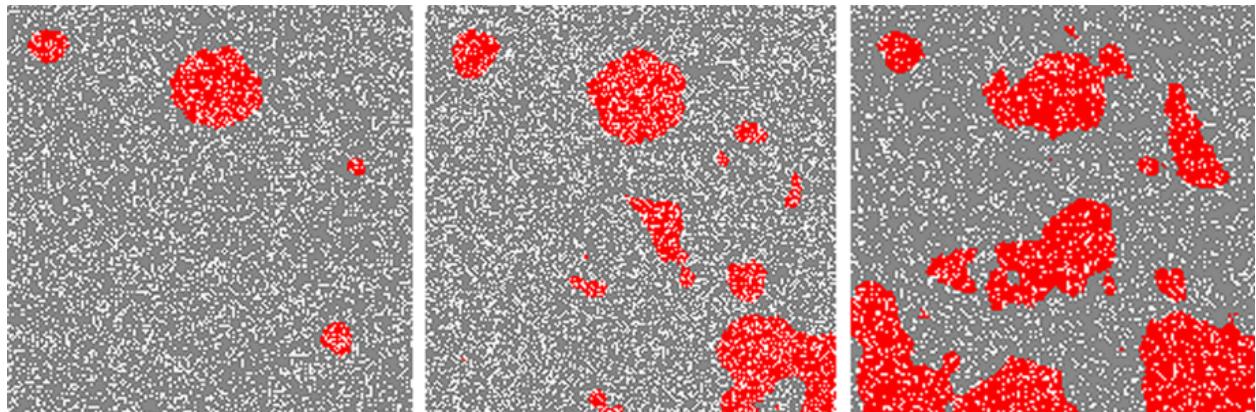
In Fig. 4.5, the upper-right portion of the large crater in the middle clearly shows this advantage: the PointNet++ model correctly classifies the crater region, whereas PointNet incorrectly classifies it as background. However, PointNet++ exhibits more serious over-classification of background around small craters, incorrectly labeling background as crater edges.

In Fig. 4.6, however, we can see that in other samples, the phenomenon of over-expansion of crater edges is similar between PointNet and PointNet++. This suggests that such errors are not caused solely by the hierarchical structure and are influenced by other factors as well.



*Figure 4.6 Comparison of prediction results: PointNet++ (left two) vs. PointNet (right two)*

#### 4.2.2 RandLA-Net vs. the PointNet Family



*Figure 4.7 Comparison of prediction results: RandLA-Net (left), PointNet++ (middle), PointNet (right)*

As shown in Fig. 4.7, RandLA-Net performs significantly better than PointNet++ and PointNet when dealing with multi-scale craters. Although it sacrifices correct detection of a few very small

craters, it preserves accurate boundaries for craters at various scales. By contrast, PointNet++ and PointNet detect almost all craters, but most crater boundaries are poorly extracted; the background near crater edges is often over-classified as crater or as a separate crater.

Overall, compared with the PointNet family, RandLA-Net shows better performance when a single sample contains craters of multiple scales, and its extraction of local features around crater boundaries is more detailed, leading to better performance in fine-scale regions.

In addition, under the same hardware configuration, number of sampled points (80,000), and batch size (batch size = 8), the average training time per epoch for RandLA-Net is **1 minute 29 seconds**, while PointNet++ and PointNet require **2 minutes 20 seconds** and **1 minute 57 seconds**, respectively. This indicates that the random sampling strategy in RandLA-Net provides better computational efficiency.

## 5.1 Attention Mechanism in the RandLA-Net Model

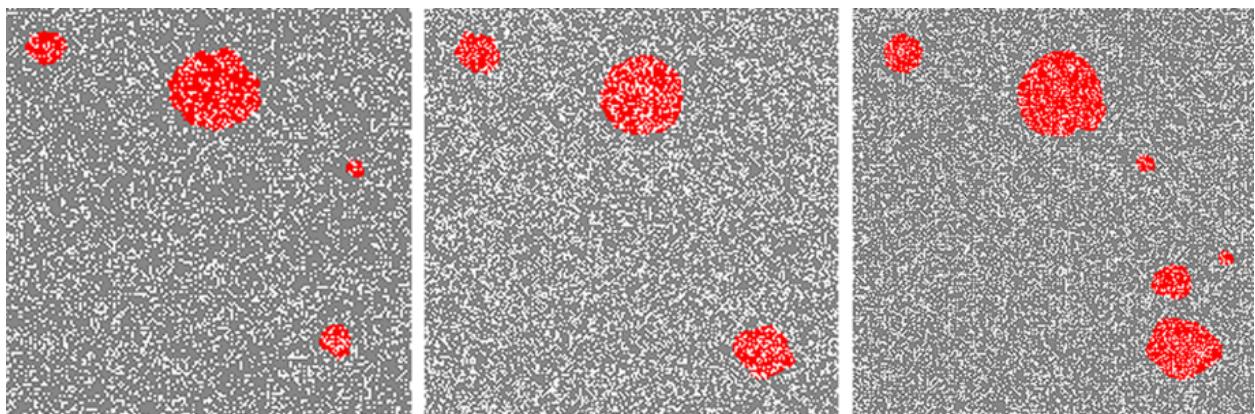


Figure 4.8 Attentive Pooling (left), Max Pooling (middle), ground truth (right)

Figure 4.8 compares the results obtained when the attentive pooling block in the local feature aggregation module of RandLA-Net is replaced by the max pooling layer used in PointNet.

In the original RandLA-Net with the attentive pooling block, craters at three different scales are successfully detected. However, when the attentive pooling block is replaced with max pooling, the smallest crater on the right side of the figure is not correctly extracted. In this case, too much information about the point features of the smallest crater is lost during max pooling.

# 5 Conclusions and Outlook

## 5.1 Conclusions

Beginning with the construction of a point-cloud dataset, this study completes a full pipeline of model training, testing, and result analysis. By analyzing the architectures of different models

and the reasons for their varying performance, it focuses on how multi-scale effects influence model behavior.

HRSC DEM elevation data of the Martian surface are chosen as the original data source. The 2.5D DEM data are converted into 3D point-cloud data, and a crater point-cloud dataset is created through data preprocessing and data augmentation. The dataset is divided into training, validation, and test sets.

The PointNet, PointNet++, and RandLA-Net models are configured and trained on the same dataset. For five representative test samples containing craters at multiple scales, the prediction results are analyzed in detail, with particular attention to the classification of craters and crater edges at different scales within the same image. The structural characteristics of each model are then examined to explain the differences in their prediction results.

In terms of multi-scale crater extraction from the same sample—especially the extraction performance at crater edges—RandLA-Net performs best due to its specially designed local feature aggregation module. PointNet++, which employs hierarchical sampling and grouping strategies, clearly outperforms PointNet, which lacks explicit multi-scale design.

In terms of computational complexity, the random sampling strategy of RandLA-Net is more efficient than the farthest point sampling of PointNet++, which in turn is more efficient than PointNet with no explicit sampling strategy.

## 5.2 Outlook

The three models used in this study were not originally designed for recognizing any specific object class, and no explicit constraints were imposed on the target shape. However, most Martian impact craters are circular or elliptical. Therefore, if morphological constraints emphasizing circular boundaries could be incorporated into the neural network design, it would better adapt the models to crater extraction tasks.

From the analysis of the three models, we observe that PointNet++ and PointNet tend to misclassify background around craters as part of the crater or as new craters, whereas RandLA-Net tends to ignore smaller craters. Different strategies can be adopted depending on the specific application scenario. For example, in lander touchdown site safety analysis, some false positives for craters may be acceptable. In contrast, for geological sampling on the Martian surface, small craters may be less important, and missing some small-scale craters may be acceptable.

In this study, the main evaluation metric is the mean Intersection over Union (mIoU). For future multi-class problems or specific task scenarios, IoU for individual classes can be emphasized, and additional task-specific evaluation metrics should be designed accordingly.

Although the dataset used in this study ensures that almost every sample contains craters, the crater density is imbalanced across samples. In some samples, crater area accounts for

two-thirds of the total area, while in others, crater area is less than one-tenth. The number of craters also varies significantly across samples. To better handle such imbalance in both crater area and crater count, more targeted loss functions—such as **Focal Loss**—could be designed.

The attention mechanism introduced in RandLA-Net has already demonstrated strong performance in CNN-based models. In the future, attention mechanisms could be integrated into the PointNet and PointNet++ architectures to help the models better capture fine-grained local point features and improve classification performance around crater edges.

## 6 References

- [1] HU Y., XIAO J., LIU L., et al. Detection of small impact craters via semantic segmenting lunar point clouds using deep learning network [J]. *Remote Sensing*, 2021, 13(9): 1826.
- [2] DELATTE D. M., CRITES S. T., GUTTENBERG N., et al. Segmentation convolutional neural networks for automatic crater detection on Mars [J]. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 2019, 12(8): 2944–2957.
- [3] ZHA K., YUAN J. Landing Site Selection of Detector Based on 3D Point Cloud Segmentation [C]. *Proceedings of the 2021 International Conference on Mechanical, Aerospace and Automotive Engineering*, 2021.
- [4] MELOSH H. J. Impact Cratering: A Geologic Process [J]. New York: Oxford University Press; Oxford: Clarendon Press, 1989.
- [5] LIU YANG, WU XING, LIU ZHENGHAO, et al. Advances in research on the geological evolution and habitable environment of Mars [J]. *Reviews in Earth and Planetary Physics*, 2021, 52(4): 416–436.
- [6] WATTERS W. A., GEIGER L. M., FENDROCK M., et al. Morphometry of small recent impact craters on Mars: Size and terrain dependence, short-term modification [J]. *Journal of Geophysical Research: Planets*, 2015, 120(2): 226–254.
- [7] FREY H. V., ROARK J. H., SHOCKEY K. M., et al. Ancient lowlands on Mars [J]. *Geophysical Research Letters*, 2002, 29(10): 22-1–22-4.
- [8] SINGH P., SARKAR R., PORWAL A. Orbital remote sensing of impact-induced hydrothermal systems on Mars [J]. *Ore Geology Reviews*, 2019, 108: 101–111.
- [9] BYRNE S., DUNDAS C. M., KENNEDY M. R., et al. Distribution of mid-latitude ground ice on Mars from new impact craters [J]. *Science*, 2009, 325(5948): 1674–1676.
- [10] BARLOW N. G. Crater size-frequency distributions and a revised Martian relative chronology [J]. *Icarus*, 1988, 75(2): 285–305.
- [11] SALAMUNICCAR G., LONCARIC S. Method for crater detection from Martian digital topography data using gradient value/orientation, morphometry, vote analysis, slip tuning, and calibration [J]. *IEEE Transactions on Geoscience and Remote Sensing*, 2010, 48(5): 2317–2329.

- [12] KIM J. R., MULLER J.-P., VAN GASSELT S., et al. Automated crater detection: A new tool for Mars cartography and chronology [J]. *Photogrammetric Engineering & Remote Sensing*, 2005, 71(10): 1205–1217.
- [13] CHENG Y., JOHNSON A. E., MATTHIES L. H., et al. Optical landmark detection for spacecraft navigation [C]. 2003.
- [14] BARATA T., ALVES E. I., SARAIVA J., et al. Automatic recognition of impact craters on the surface of Mars [C]. *Proceedings of the International Conference on Image Analysis and Recognition (ICIAR)*, 2004.
- [15] WETZLER P. G., HONDA R., ENKE B., et al. Learning to detect small impact craters [C]. *Proceedings of the 2005 IEEE Workshops on Applications of Computer Vision (WACV/MOTION'05)*, 2005.
- [16] STEPINSKI T. F., MENDENHALL M. P., BUE B. D. Machine cataloging of impact craters on Mars [J]. *Icarus*, 2009, 203(1): 77–87.
- [17] DI K., LI W., YUE Z., et al. A machine learning approach to crater detection from topographic data [J]. *Advances in Space Research*, 2014, 54(11): 2419–2429.
- [18] PEDROSA M. M., DE AZEVEDO S. C., DA SILVA E. A., et al. Improved automatic impact crater detection on Mars based on morphological image processing and template matching [J]. *Geomatics, Natural Hazards and Risk*, 2017, 8(2): 1306–1319.
- [19] SILBURT A., ALI-DIB M., ZHU C., et al. Lunar crater identification via deep learning [J]. *Icarus*, 2019, 317: 27–38.
- [20] LEE C. Automated crater detection on Mars using deep learning [J]. *Planetary and Space Science*, 2019, 170: 16–28.
- [21] LAWIN F. J., DANELLJAN M., TOSTEBERG P., et al. Deep projective 3D semantic segmentation [C]. *Computer Analysis of Images and Patterns (CAIP)*, 2017.
- [22] BOULCH A., LE SAUX B., AUDEBERT N. Unstructured point cloud semantic labeling using deep segmentation networks [J]. *3DOR@Eurographics*, 2017, 3: 1–8.
- [23] SU H., MAJI S., KALOGERAKIS E., et al. Multi-view convolutional neural networks for 3D shape recognition [C]. *2015 IEEE International Conference on Computer Vision*, 2015.
- [24] RIEGLER G., OSMAN ULUSOY A., GEIGER A. OctNet: Learning deep 3D representations at high resolutions [C]. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [25] QI C. R., SU H., MO K., et al. PointNet: Deep learning on point sets for 3D classification and segmentation [C]. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [26] QI C. R., YI L., SU H., et al. PointNet++: Deep hierarchical feature learning on point sets in a metric space [J]. *Advances in Neural Information Processing Systems (NeurIPS)*, 2017.

- [27] HU Q., YANG B., XIE L., et al. RandLA-Net: Efficient semantic segmentation of large-scale point clouds [C]. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.
- [28] JIANG M., WU Y., ZHAO T. PointSIFT: A SIFT-like network module for 3D point cloud semantic segmentation [J]. *arXiv preprint arXiv:1807.00652*, 2018.
- [29] CHEN DONG, HU FAN, ZHANG LIQIANG, et al. A review of impact crater recognition methods [J]. *Science China: Earth Sciences*, pp. 1–24.
- [30] VINYALS O., BENGIO S., KUDLUR M. Order matters: Sequence to sequence for sets [J]. *arXiv preprint arXiv:1511.06391*, 2015.
- [31] JADERBERG M., SIMONYAN K., ZISSERMAN A. Spatial transformer networks [J]. *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.
- [32] ZHANG Z., LIN L., ZHI X. R-PointNet: Robust 3D object recognition network for real-world point clouds corruption [J]. *Applied Sciences*, 2024, 14(9): 3649.
- [33] GAO HONGYU. Discussion on oblique UAV image mapping technology based on a multi-strip forward-intersection algorithm [J]. *Coordinate & Space*, 2020(06): 79–82 + 6.