Rapid extraction of pavement aggregate gradation based on point clouds using deep learning networks

Chen, Siyu; Chen, Can; Ma, Tao; Han, Chengjia; Luo, Haoyuan; Wang, Siqi; Gao, Yangming; Yang, Yaowen

DOI
10.1016/j.autcon.2023.105023

Publication date
2023

Document Version
Final published version

Published in
Automation in Construction

Automation in Construction 154 (2023) 105023

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Rapid extraction of pavement aggregate gradation based on point clouds using deep learning networks

Siyu Chen [a], Can Chen [a], Tao Ma [a], Chengjia Han [a,c,*], Haoyuan Luo [a], Siqi Wang [a], Yangming Gao [b], Yaowen Yang [c]

[a] School of Transportation, Southeast University, Nanjing, China
[b] Department of Engineering Structures, Delft University of Technology, Delft, Netherlands
[c] School of Civil and Environmental Engineering, Nanyang Technological University, Singapore

ARTICLE INFO

Keywords:
Asphalt pavement
Aggregate gradation
Point clouds
Artificial neural networks
Multi-feature fusion

ABSTRACT

Usage of asphalt mixture with poor gradation will most likely lead to pavement deficiency. There is a growing need for rapid and non-destructive methods to extract pavement aggregate gradation. In this study, a deep learning-based method that utilizes point clouds data for gradation extraction was proposed. Firstly, a data enhancement algorithm along with three data format conversion methods (aligned point cloud, voxel, and depth image) were proposed to preprocess the original collected point clouds. Subsequently, different neural network models were designed for each data format to extract gradation. Finally, a multi-feature fusion network was developed, which using extraction network as the backbone and additional auxiliary information. In the case study, the MAE loss of multi-feature fusion networks with PointNet, Vox-ResNet34 and GoogLeNet-v4 as the backbone respectively achieved 0.202, 0.142 and 0.046 on the test set, which means an estimation accuracy of more than 95% for the pavement aggregate gradation.

1. Introduction

Aggregates are important ingredients of asphalt mixture, providing microtexture and serving parts of macrotexture for the asphalt pavement. Aggregate gradation refers to the specific size distribution of aggregates, which significantly affects the overall properties of the asphalt mixtures, encompassing high-temperature performance and modulus property [1–6]. Asphalt mixture with inadequate aggregate gradation is susceptible to early moisture damage, compromised skid resistance, and other performance issues [7,8]. Currently, researchers and pavement engineers have developed different types of asphalt mixtures to achieve different application purposes. Dense-graded Asphalt Concrete (AC), gap-graded Stone Matrix Asphalt (SMA), and Open-graded Friction Course (OGFC) are the most widely used mixtures [9]. Different types of gradations represent variations in size distribution. However, there are still a gap in quality control (QC) of aggregate gradation during pavement construction [10]. The actual aggregate distribution may differ from the designed aggregate gradation, due to various reasons during mixing and paving processes. Therefore, it becomes essential to accurately estimate the aggregate gradation on-site to minimize discrepancies between mixture design and pavement construction.

The traditional methods to determine aggregate gradation could be classified as indoor or outdoor test methods. The indoor test methods include asphalt binder extraction method and X-ray Computed Tomography (CT) technique. The binder extraction method is used to separate the asphalt binder from the aggregate in the asphalt mixture [11]. The aggregate gradation can be estimated after the binder extracted samples. However, this method is typically used for sampling in the laboratory, which cannot be used for on-site evaluation. The X-ray CT method captures a series of CT images from the field core, which are then processed through digital image processing techniques to reconstruct a 3D numerical model of the asphalt mixture [12,13]. Therefore, the aggregate gradations can be estimated from the virtual asphalt mixture. However, the CT images were obtained from drilled cores, making the whole process time-consuming.

Outdoor testing methods involve the utilization of cameras, laser sensors, and other tools to capture digital information. The analysis of aggregate of pavement is achieved through image segmentation methods [14]. However, the pavement images obtained from cameras are limited to 2D images, lacking texture depth and the ability to discern

* Corresponding author at: School of Transportation, Southeast University, Nanjing, China.
E-mail address: chengjia.han@ntu.edu.sg (C. Han).

https://doi.org/10.1016/j.autcon.2023.105023
Received 12 December 2022; Received in revised form 5 June 2023; Accepted 11 July 2023
Available online 20 July 2023

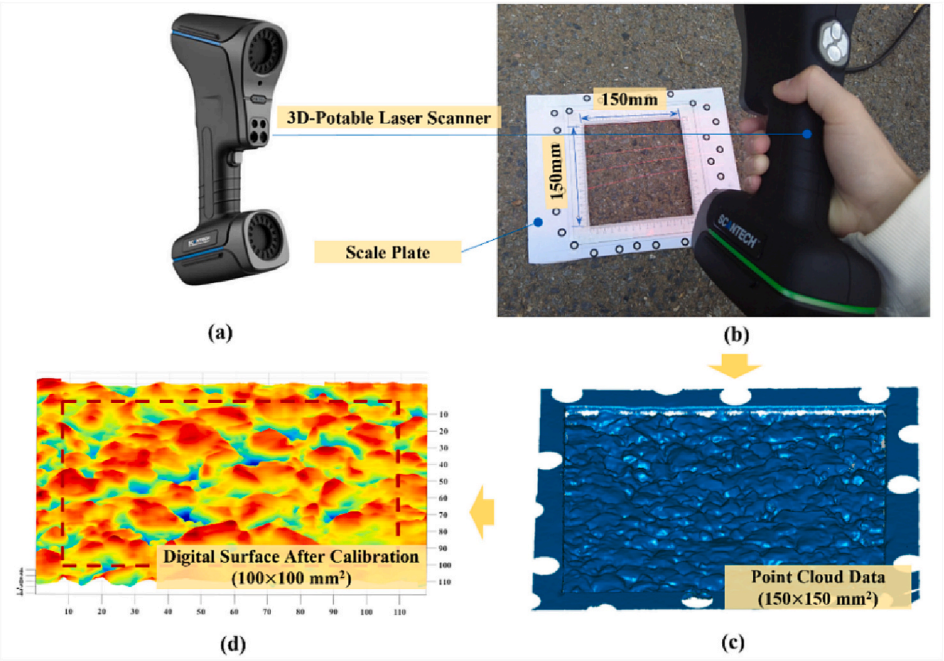


Fig. 1. Data acquisition process of the pavement point cloud.

fine spatial properties of aggregate. With the aid of 3D-laser scanning technology, the pavement surface texture can be captured at a high resolution, reaching a minimum level of mm-level precision [15]. Researchers attempted to investigate the 3D information of the pavement, such as highway obstacles [16], distress detection [17,18], surface texture [19]. However, there have been limited studies conducted on the evaluation of pavement aggregate gradation. Huyan et al. [20] conducted a quantitative analysis on the macrotexture of asphalt pavement based on 3D point cloud data, establishing a mapping between macrotexture and 10 characteristic indicators. However, they did not extract the gradation. On the other hand, Weng et al. [21] proposed a multi-featured fusion network based on residual convolutional neural network to estimate the aggregate gradation using 3D data obtained from laser scanners. However, they directly reduced the dimension of point cloud data into image data for convolution, lacking comprehensive analysis of point cloud data features and corresponding applications. To address these challenges, in this study, a rapid extraction method for pavement aggregate gradation based on point cloud data is proposed. The objective of this study is to rapidly estimate pavement aggregate gradation using point cloud-based deep learning networks. The contributions can be summarized as follows:

(1) An algorithm for point cloud data enhancement based on spatial distribution similarity was proposed, which implemented a small number of original point clouds to obtain a large number of high-quality point cloud samples that are to satisfy neural network training. The neural network trained using the enhanced point clouds achieved more than 0.9 F1-score in the gradation extraction task.

(2) The original point cloud transformation method of pavement was proposed, which converted the original point cloud into three types of data formats that aligned the point cloud, voxel and depth image. Moreover, the corresponding multi-type of neural networks were built to test their performance on the extraction of aggregate gradation. It was found that the neural networks using the voxel format performed more consistently and better in terms of average F1-score, average mean absolute error (MAE) loss and confusion matrix. In addition, the extraction network using GoogLeNet as the backbone and using depth images achieved the best performance with a MAE loss of 0.046.

(3) It was found that the point cloud density affects the performance of the point cloud neural networks. For the task of pavement aggregate gradation extraction, it was recommended to use 6000 points to characterize the asphalt pavement within a 51 × 51 mm^2 area.

(4) A multi-feature fusion network was proposed in this study, which utilizes a point cloud, voxel, or depth image extraction network as the backbone and incorporates auxiliary information. The multi-feature fusion network has a large improvement in extraction accuracy and training stability compared with their backbone network.

This study is organized as follows. The data obtaining process is presented in Section 2. The point cloud data enhancement and the three transformation methods are showed in Section 3. The aggregate gradation extraction methods using various neural networks are showed in Section 4. The case study including the application of the proposed method and the corresponding discussions are presented in Section 5. The conclusion is provided in the final section.

2. Point cloud data acquisition

2.1. Data acquisition

The digital pavement surface was obtained using a 3D-Portable laser scanner, as shown in Fig. 1(a). The 3D laser scanner is KSCAN 20 laser scanner combines infrared and blue lasers into one single device. By sending 14 crossed laser beams and receiving these laser beams from the pre-set reflective marking dots, this laser scanner can export the 3D coordinate information of the scanned surface. The horizontal and vertical resolution of the laser scanner are 0.05 and 0.02 mm, respectively.

For each selected pavement surface, a 150 × 150 mm^2 area was chosen for scanning, ensuring a minimum of 7500 × 7500 scanning points (Fig. 1(b) and (c)). The original point cloud data of each surface contained noise points, discrete points, and points outside the scanning area. To address this, the point cloud processing software, Geomagic Wrap, was employed to edit the scanned point cloud of the pavement surface. Through trial and error, the denoising process of the point clouds was achieved as follows. Outliers were visually detected and subsequently removed. A prismatic-shaped noise filter was selected, and smoothness level one was applied to enhance the scanning surface. The noise points were filtered out, and a reduced area of 100 × 100 mm^2 (Fig. 1(d)) was cropped out for further analysis.

Table 1
Texture parameters for pavement surface texture evaluation.

Parameters	Equation/Comments
Arithmetic mean height, S_a	$S_a = \frac{1}{A}\iint \lvert z(x,y)\rvert dxdy$
Root Mean Square Height, S_q	$S_q = \sqrt{\frac{1}{A}\iint z^2(x,y)dxdy}$
Skewness, S_{sk}	$S_{sk} = \frac{1}{S_q^3}\left[\frac{1}{A}\iint z^3(x,y)dxdy\right]$
Kurtosis, S_{ku}	$S_{ku} = \frac{1}{S_q^4}\left[\frac{1}{A}\iint z^4(x,y)dxdy\right]$
Mean Texture Depth, MTD	Mean texture depth using sand patch method
Estimated Mean Texture Depth, EMTD	Ratio between enclosed volume to the surface area
Zp0	Height distribution passes 0%, the lowest height
Zp20	Height distribution passes 20%
Zp40	Height distribution passes 40%
Zp60	Height distribution passes 60%
Zp80	Height distribution passes 80%
Zp100	Height distribution passes 100%, the top height

Note: A is the area of the footprint surface which is usually a rectangular area, and z(x,y) is the height of a point located at given (x, y) coordinates.

2.2. Indicators

In this study, the distribution of surface heights was plotted by utilizing the probability density function of the height values. The surface characterization of the four mixtures was investigated. To understand the amplitude and wavelength of the surface texture, three parallel slicers were cut on each selected surface with a gap of 25 mm. Therefore, profiles with the length of 100 mm were extracted from the surface textures.

Unlike 2D texture analysis according to the surface profiles, the purpose of texture analysis is to provide parameters that can be more representative for evaluating specific areas of the pavement. Such analysis becomes more meaningful when considering the interaction between vehicle tires and the pavement or the related behavior under rainfall conditions [22]. In addition, many studies have shown that regional statistical parameters are closely related to skid resistance [23–25]. To analyze the surface characterization, two height parameters (arithmetic mean height, root mean square height), two shape parameters (skewness, kurtosis), and two volume parameters (mean texture depth, estimated mean texture depth) will be considered respectively and calculated [26], as shown in Table 1.

The actual area of contact between the tire and the pavement is regarded as the bearing area, and it can be obtained from a surface map. Therefore, the characterization of surface texture was conducted solely based on the topography of the asphalt pavements, without considering other factors. Mahboob Kanafi and Tuononen concluded that a high correlation of 0.8 was observed between the friction and the top 20% of the surface topography [27]. Du et al. selected the top 30% of the surface topography to analysis the surface texture of the selected asphalt pavements [28]. In this study, top topography area for the four mixtures from 0% to 100% were extracted and evaluated.

The Sand Patch Test [29] was used to assess the pavement surface texture. A specified volume of sand was spread on the pavement in a circular motion using a spreading tool. The sands fill the voids of the pavement surface and form a circle. The diameter of the resulting circle was measured at four different direction and the average value was used. An estimated average depth of the measured surface texture was determined by dividing the volume of used sands to the patched circle area, such depth is regarded as the Mean Texture Depth (MTD). A larger patched circular area on the measured surface indicates a lower estimated texture depth. Typically, a small circular area was desired to obtain a relatively coarse texture, which indicates a good performance for skid resistance.

3. Data enhancement and transformation method for point clouds

To achieve a high accuracy for the surface texture, the point clouds were in a dense form. For example, a scanning area of 100 × 100 mm^2 includes about 300,000 raw point clouds. Direct processing of the raw point clouds leads to insufficient sample size for model training, while the computational cost of a single sample size is too high. Therefore, after the raw point clouds were obtained, the original point clouds were cut into several small sections with appropriate size. In this way, not only did it increase the sample size of the dataset used for machine learning model training, but it also improved the quality of each individual sample data. A novel method for enhancing point clouds was proposed as follows.

3.1. Point clouds enhancement method based on distribution similarity

The key issue for segmentation of the original point cloud is the size of the separated point cloud. Data enhancement must ensure that the

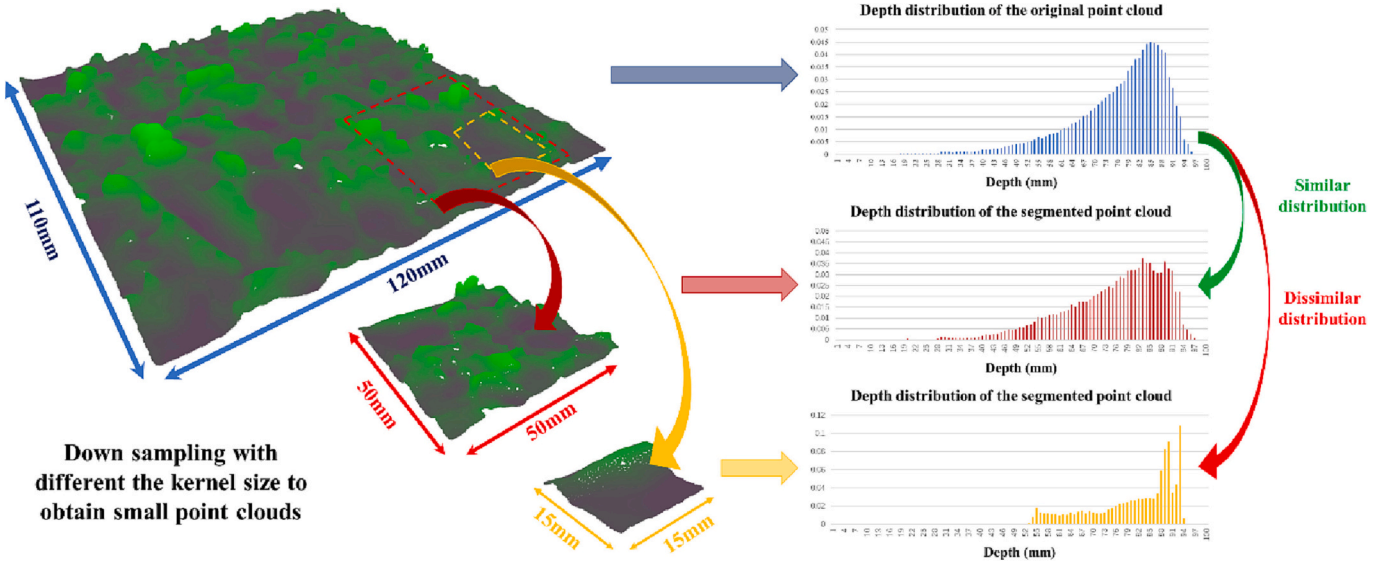


Fig. 2. Distribution difference of point clouds obtained with different segmentation sizes.

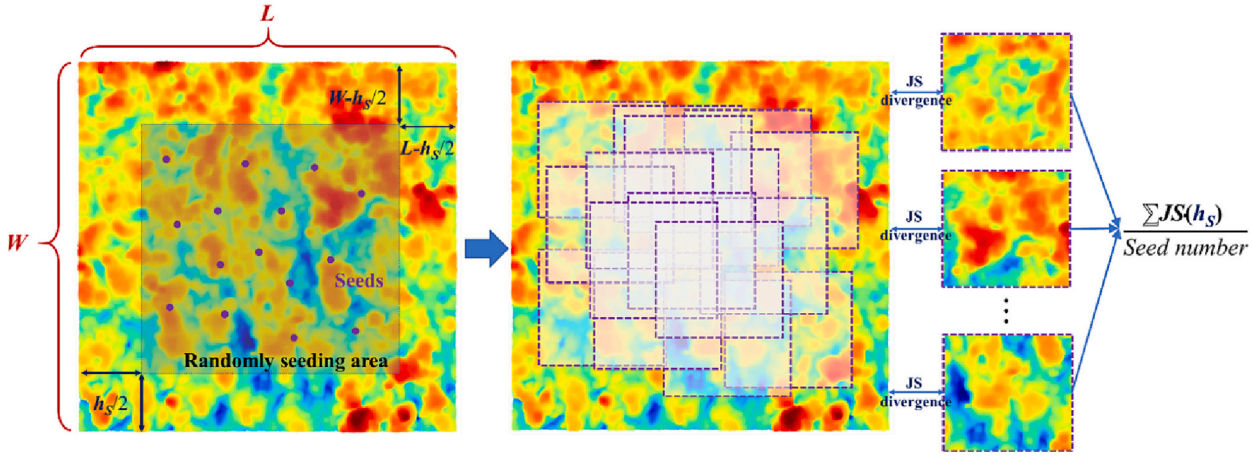


Fig. 3. Estimation of distribution similarity under different segmentation sizes based on random seed distribution.

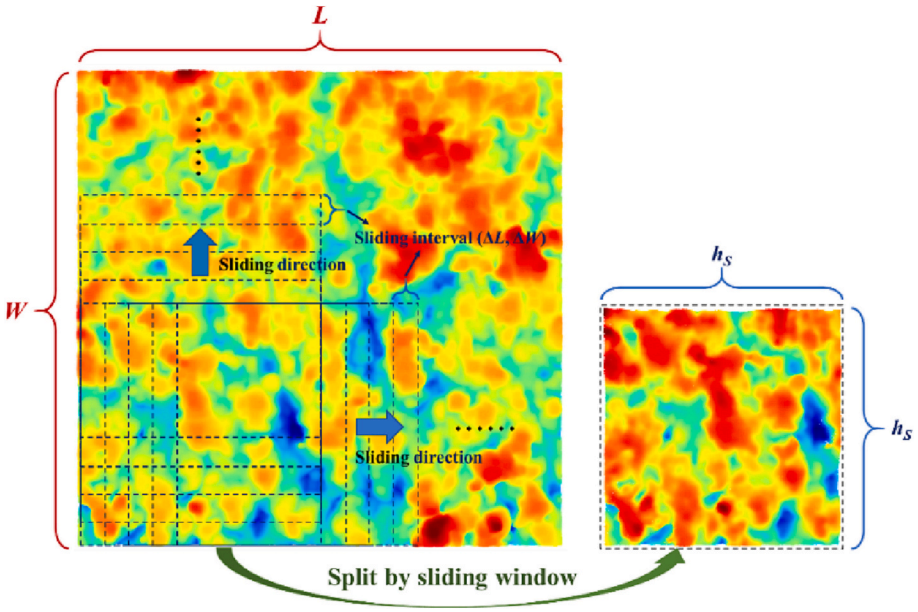


Fig. 4. Original Point cloud segmented by window sliding method.

segmented point cloud is similar to the original point cloud. Asphalt mixture is a collection of aggregates with different particle sizes. Although it is uniform, this uniformity can only be maintained within a certain spatial range. For example, in the extreme case when the segmented point cloud area is smaller than the maximum nominal particle size of the asphalt mixture, the point cloud can no longer represent the spatial shape of asphalt mixture. The Fig. 2 below illustrates the distribution differences between the original point cloud and the segmented point cloud with various sizes. The left figures are the original point cloud (110 mm × 120 mm) and two segmented point clouds with size (50 mm × 50 mm) and (15 mm × 15 mm), meanwhile the right figures are the spatial distribution of corresponding point clouds. It is evident that the distribution of the segmented point cloud with a size of (50 mm × 50 mm) is similar to that of the original point cloud, whereas the distribution of the segmented point cloud with a size of (12 mm × 12 mm) is completely different.

It shows a contradiction between the number of point clouds after division and the quality of the segmented point clouds. When the segmented size is larger, the spatial distribution of each segmented point cloud becomes more similar to that of the corresponding original point cloud, indicating higher data quality. Therefore, it is necessary to find an appropriate segmented size to balance the number and quality of point cloud division. Therefore, a data enhancement method was proposed, including segmented size determination and sliding segmentation. First, the process of determining segmented size was introduced. The segmentation size was continuously reduced from the maximum size until that the segmented point clouds were no longer resemble the original distribution or the predetermined minimum size was reached. The preset minimum size is two times of the maximum nominal size of the mixture gradation, which is an empirical value. This process requires quantitative evaluation of the similarity between the segmented point clouds and the original point cloud, as shown in Fig. 3.

For any split size $h_S \in [\min(L,W), 2S_g]$, where L and W is the length and width of original point cloud (mm), and S_g is the maximum nominal size of the mixture gradation. As shown in Fig. 3, an area in the original point cloud was framed, so that a square with a side length of h_S drawn for the centroid at any point in this area will not exceed the boundary of the original point cloud. This area is called randomly seeding area and the positions of n seeds in this region were determined randomly based on a two-dimensional uniform distribution. After that, each seed was used as the center of the square new point cloud which was cropped out of the original point cloud. Then, the spatial distribution of each new

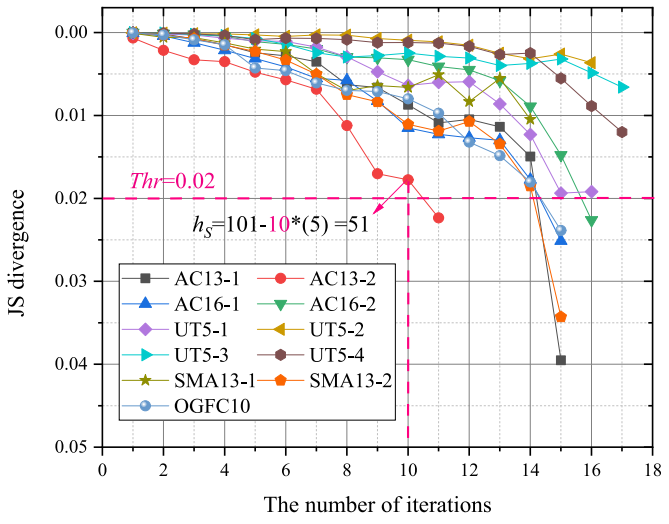


Fig. 5. The JS Divergence changing with the split size decreasing of different original point clouds.

segmented point cloud was calculated, and its distribution difference with the original point cloud through Jensen-Shannon (JS) divergence [30] was estimated as

$$JS(P \parallel Q) = \frac{1}{2}\left[\sum P(x) log \frac{P(x)}{(P(x)+Q(x))/2} + \sum Q(x) log \frac{Q(x)}{(P(x)+Q(x))/2}\right], \quad (1)$$

where P and Q represent the distribution of the new segmented point clouds and the original point clouds, respectively. The JS divergences of the new point cloud and the original point cloud were calculated and averaged. The final result denoted as $JS_f(hs)$, was set as the segmentation similarity estimation corresponding to the current segmentation size h_S. According to the selected threshold $JS(Thr)$, the process was stopped when $JS_f(hs) > JS(Thr)$, reaching the best split size of the current point cloud. When there are K original point clouds, the final split size $h_{S\text{-}f}$ is taken the maximum value as

$$h_{S-f} = max\{h_{S-1}, h_{S-2} \ldots h_{S-K}\}, \quad (2)$$

where $h_{S\text{-}k}$ is the obtained split size h_S of the k-th original point cloud.

When the split size was determined, the next step was to conduct the segmentation of original point clouds. To fully utilize the original point clouds, and partially overlap the new point clouds to improve the information redundancy of training data, the window sliding method was used for segmentation as shown in Fig. 4. The Sliding interval is premarked as (ΔL, ΔW), which is the forward step is in the direction of length and width (mm). The number of new point clouds was obtained by splitting the k-th original point cloud, which is denoted as N_k and calculated as

$$N_k = \frac{(W_k - h_{S-k})(L_k - h_{S-k})}{\Delta W_k \cdot \Delta L_k}. \quad (3)$$

The pseudo code of the whole process of point cloud data enhancement is as follows:

Algorithm:	Data enhancement by splitting the original point clouds into several small point clouds
Inputs:	Raw point clouds dataset O = {o_1, $o_2 \ldots o_K$}, Threshold *Thr*, Minimum split size *min_size*
Outputs:	Split point clouds dataset S = {S_1, $S_2 \ldots S_N$}
Stage 1: Calculate the division dimension h_S	
1:	For k in O = {o_1, $o_2 \ldots o_K$}, h = []:
2:	Calculate the envelope size of imported point clouds ($L \times W \times H$), *max_size* = int(min(L,W))
3:	For i in range (*max_size*, *min_size*, −5):
4:	Calculate the seeding area [*new_L_min*: *new_L_max*, *new_W_min*: *new_W_max*]
5:	Randomly distribute seeds in area, seed number is 20:
6:	*seed_L* = [random.uniform (*new_L_min*, *new_L_max*) for _ in range(20)]
7:	*seed_W* = [random.uniform (*new_W_min*, *new_W_max*) for _ in range (20)]
8:	For j in range (20):
9:	*Split-area*(j) = o_k [(*seed_L*(j)-i/2): (*seed_L*(j) + i/2), (*seed_W*(j)-i/2): (*seed_W*(j) + i/2)]
10:	Calculate JS divergence of split graph and original graph: *JS*(j) = *JS* (*Split-area*(j), o_k)
11:	*JS_max* = np.max (*JS*)
12:	If (*JS_max* > *Thr*):
13:	$h(k) = i + 5$,
14:	Break
15:	h_S = np.max (h)
Stage 2: Sliding segmentation of the original image according to h_S	
16:	For k in O = {o_1, $o_2 \ldots o_K$}, S = []:
17:	$L(k)$ = np.max (o_k[:,0]), $W(k)$ = np.max (o_k[:,1])
18:	For p in range (0, $L(k)$-h_S, 5):
19:	For q in range (0, $W(k)$-h_S, 5):
20:	S.append (o_k[p: $L(k)$, q: $W(k)$])
21:	Return S = {S_1, $S_2 \ldots S_N$}

Based on the above data enhancement process, the data collected in this research that including eleven original point clouds from five types of mixture gradations were operated. The $JS(Thr)$ was selected as 0.02. The changing of JS divergence as the split size decreased of each point clouds, as shown in Fig. 5. It should be noted that 0.02 of $JS(Thr)$ in this study is a subjectively determined. Actually, the threshold is actually a balance between the number and quality of new point clouds. The larger the threshold is, the more tolerant the similarity requirements between the segmented new point clouds are; the smaller the size of the new point clouds are, and the more the number of new point clouds is. In this study, a value of 0.02 is the threshold determined after trial calculation, taking into account the number of original point clouds (11 samples) and the required order of magnitude of training sets (hundreds to thousands of samples) for neural networks. In further research, the optimized threshold selection method needs to be further studied. During this process, the initial maximum size was 101 mm, the size of each iteration was reduced by 5 mm, and the ending minimum size was 30 mm. For each point clouds, when the $JS(hs)$ was greater than 0.02 or the split size was reaching 30 mm, iteration stops.

From the Fig. 5, it can be found that the original point cloud of "AC13–2" is more sensitive to the changing of the split size. When the split size is reduced to the 10-th iterations, the difference between the spatial distribution of segmented point clouds and that of original point cloud was over the threshold. It means that the segmented point clouds may not be similar with the original point cloud if the size was reduced further. Therefore, the data set segmentation size $h_{S\text{-}f}$ was obtained as 51 (mm). The data set originally composed of 11 point clouds was enhanced to obtain 1579 new point clouds with a characterization size of 51 × 51 mm^2 area.

3.2. Point clouds transformation

After the completion of data enhancement, the new point clouds dataset will be used for pavement mixture gradation estimation. To extract effective information, there are three processing methods to process the point cloud data. The first method is to directly conduct data mining on the point clouds, the second method is to convert the point clouds to voxels and select the voxel data as the object of data mining, and the third method is to reduce the dimension of point clouds to 2D

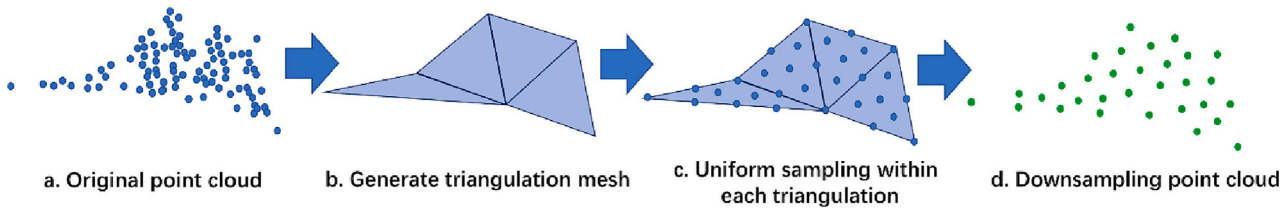


Fig. 6. The process of point cloud data downsampling.

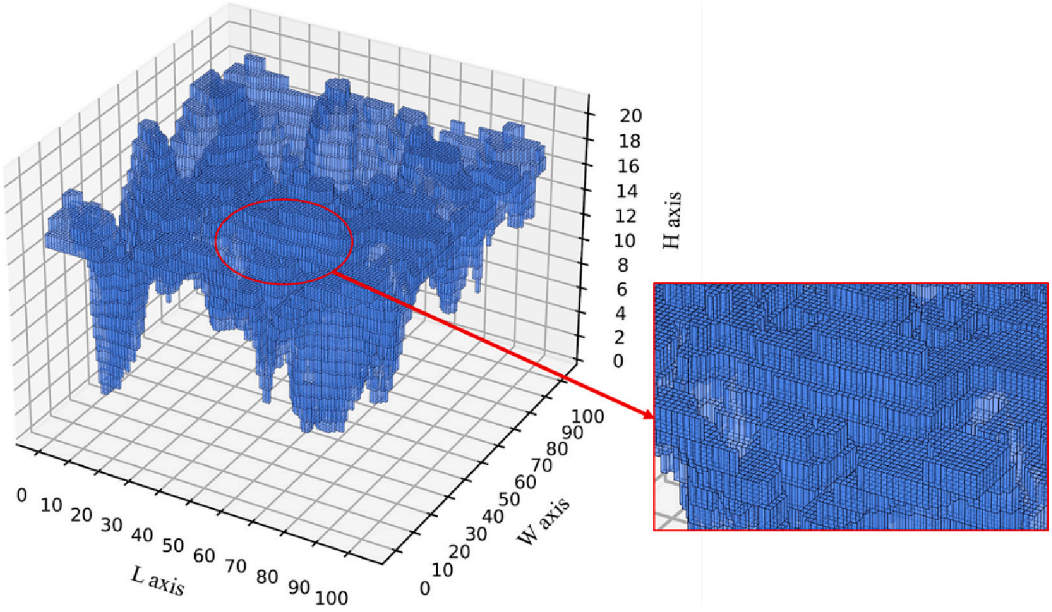


Fig. 7. 3D Voxel data with size (100 × 100 × 20 × 1) converted by point cloud data.

images and use the image data as the object of data mining. In this study, the point clouds datasets were converted into three forms of datasets to explore which data format can obtain the best effect in pavement grading estimation based on point clouds.

3.2.1. Point cloud data downsampling

For the point clouds data format, although each point cloud in the original dataset characterizes the same size pavement area, the number of points between point clouds are different. To process the point clouds in batches, the downsampling based on mesh was used for data alignment of point clouds. The process of point cloud data Downsampling is shown in Fig. 6. Specifically, the original point cloud (Fig. 6(a)) is first gridded to form a mesh (Fig. 6(b)), and then the expected number of points after downsampling is set. After that, the number of points to be sampled in each grid is calculated (Fig. 6(c)) and sampled uniformly to form a new point cloud (Fig. 6(d)). It should be noted that, in this research, the point cloud was meshed using the triangular mesh method and the expected number of point clouds after downsampling was used the number of the smallest point cloud in the original dataset. The point clouds dataset after downsampling is marked as $S_P = \{S_1, S_2 \ldots S_N\}$ with the shape of $(N \times P \times 3)$,where N is the number of samples and P is the point number of each point clouds after downsampling.

3.2.2. Voxel data converted by point cloud

Another processing method of point clouds is voxelization. As two mainstream data forms representing three-dimensional space, the biggest difference between them is whether their points are orderly [31]. The point order exchange of the point cloud does not affect the point cloud shape, but the voxel is a kind of filling after representational space gridding, which means the voxels are ordered. Therefore, there can be an unlimited points of a point cloud in the unit space, and that of a voxel depends on the sparsity of its grid. The voxels can be considered as downsampling results from the point clouds. The resolution of voxels (spatial grid sparsity) is quite important. Given that the observation scale of road surface texture is approximately 0.5 mm, the length and width of the voxels transformed from the point clouds, with a characteristic area of 51 × 51 mm^2, were set to 100 and 100, respectively. Moreover, in the depth direction, the resolution of voxels should be further improved since the pavement surface fluctuation in the depth direction is the focus of this research. Specifically, the observation scale of voxels in depth direction was set as 0.1 mm. Since the variation range of the depth direction of the research dataset is (0, 2 mm), the final voxels dataset is marked as $S_V = \{S_1, S_2 \ldots S_N\}$ with the shape of $(N \times 100 \times 100 \times 20 \times 1)$,where N is the number of samples. Since the point clouds solely contain spatial coordinates without any additional information, the values within each grid of the converted voxels follow a binary distribution, ranging from 0 to 1. One example of the converted voxels is shown in Fig. 7.

3.2.3. Depth image data converted by point cloud

Another method to process point clouds data is to convert them into 2D images, which is also a mainstream practice at present, especially for point clouds data that only contain spatial coordinate information and have simple scenes [32]. In fact, the Figs. 4 and 5 above show the depth image of the point clouds, and the color in the image represents the Z-axis elevation of the position. Specifically, in this study, the point clouds were first meshed by the triangulation mesh method. Then the mesh was uniformly sampled based on the set image size to form a 2D depth image. The depth image dataset after sampling is marked as $S_I = \{S_1, S_2 \ldots S_N\}$ with the shape of $(N \times 750 \times 750 \times 3)$,where N is the number of

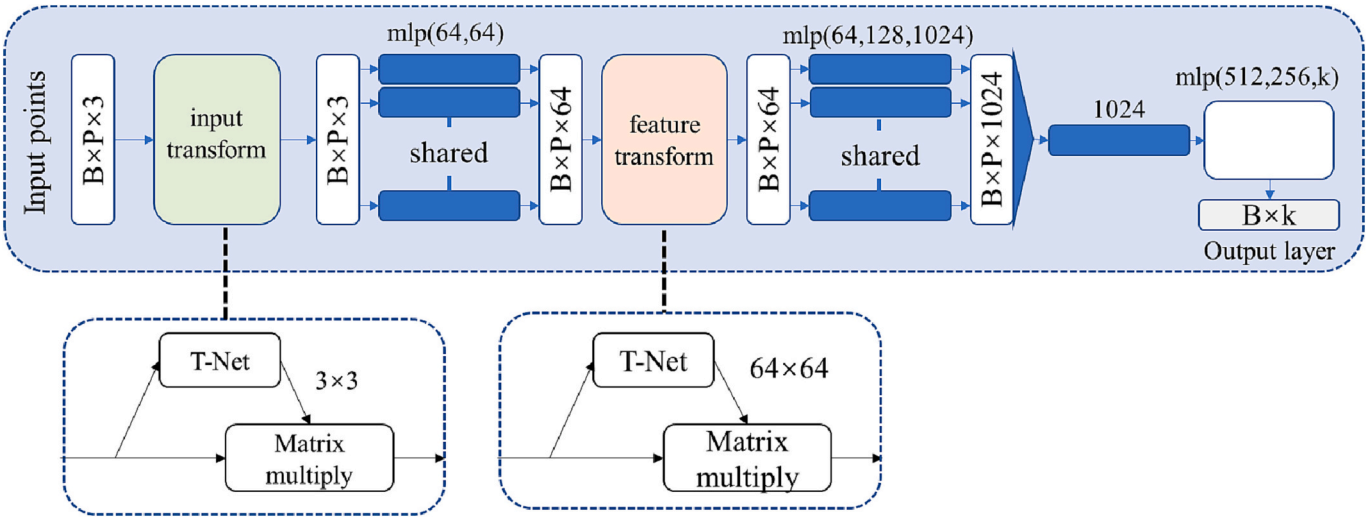


Fig. 8. The structure of used PointNet which input tensor shape is ($B \times P \times 3$) and output tensor shape is ($B \times k$), where the B is the training batch size, the P is the point number, the "3" corresponds to the spatial coordinates of each point and the k corresponds to the gradation of the asphalt mixture [33].

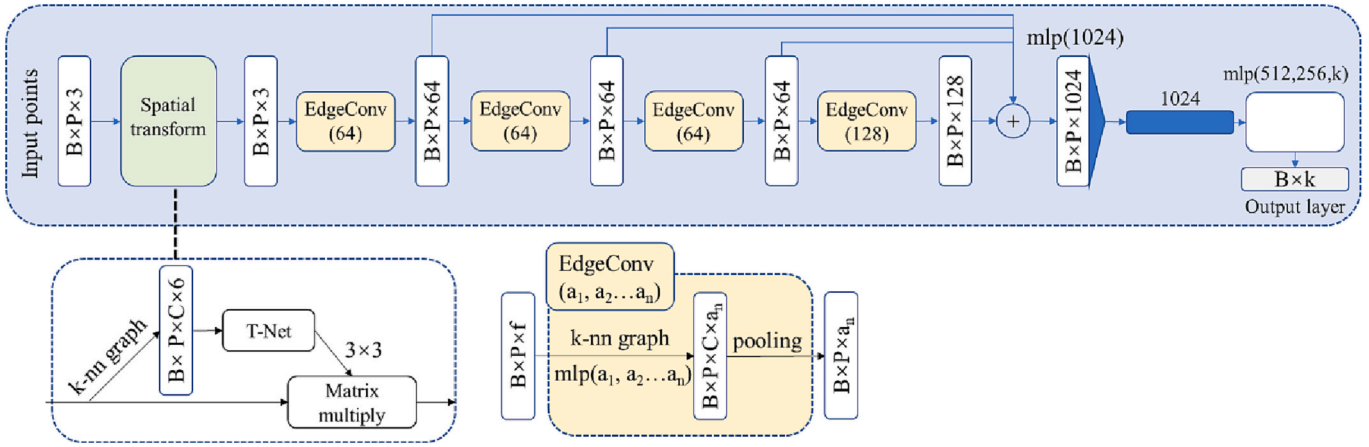


Fig. 9. The structure of used DGCNN which input tensor shape is ($B \times P \times 3$) and output tensor shape is ($B \times k$). The meaning of the input and output of DGCNN is the same as that of PointNet above [36].

samples. It should be noticed that, in this study, RGB tricolor was used to represent the elevation information of each pixel, rather than the gray value of (0, 255).

After completing the data preprocessing, three data sets S_P, S_V and S_I of three data formats were obtained. In the next section, different neural networks were established to realize the deep data mining of different data types.

4. Aggregate gradation extraction using multi-type deep learning networks

For the three data formats of point clouds, three types of neural networks to process the corresponding data formats were established. The purpose of comparing the performance of these networks is to find out which data conversion format is more suitable for the rapid pavement aggregate gradation estimation. Moreover, in this study, multi-feature fusion networks will be constructed by integrating auxiliary information into the aforementioned networks. The aim is to investigate whether the inclusion of auxiliary information can further enhance the performance of grade matching estimation. The structure of the four types of networks were introduced as follows.

4.1. Point cloud processing network

4.1.1. Class I representative network: PointNet

For the processing of the point clouds data, the current methods can be divided into two categories. The first type of processing network is represented by *PointNet* [33], which directly processes the disordered point clouds. It learns the corresponding spatial encoding for each point in the input point cloud, and then utilize the features of all points to obtain a global point cloud feature afterwards. Many new point cloud processing networks have evolved based on *PointNet* to cope with new task requirements, such as *PointNet++* [34], a point cloud segmentation task network in complex environments, and *F-PointNet*, a 3D spatial target detection task network [35], etc. However, for the classification task (regression-classification) in this study, the task itself is not complex (extract the point cloud as a small feature tensor to clarity the gradation category of pavement mixture) and has a single point cloud scene (only pavement texture, no other complex objects), so *PointNet* is more suitable than its update versions which need higher training costs but do not bring corresponding performance improvement. Therefore, the *PointNet* is chosen as the representative point cloud processing network. The structure of *PointNet* established in this study is show in Fig. 8.

The core of *PointNet* lies in two transform modules, which are aligned by multiplying the input features with the transformation matrix learned

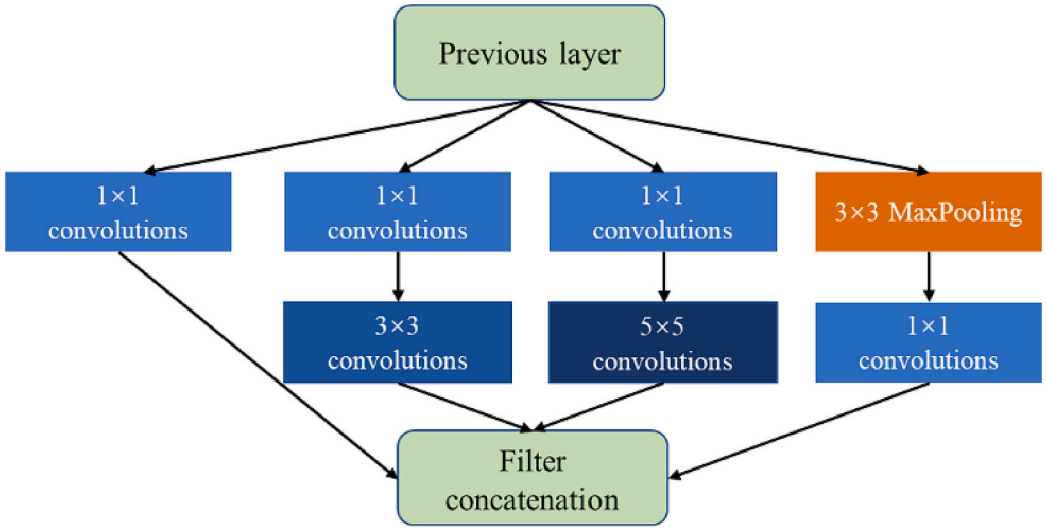


Fig. 10. The structure of Inception module.

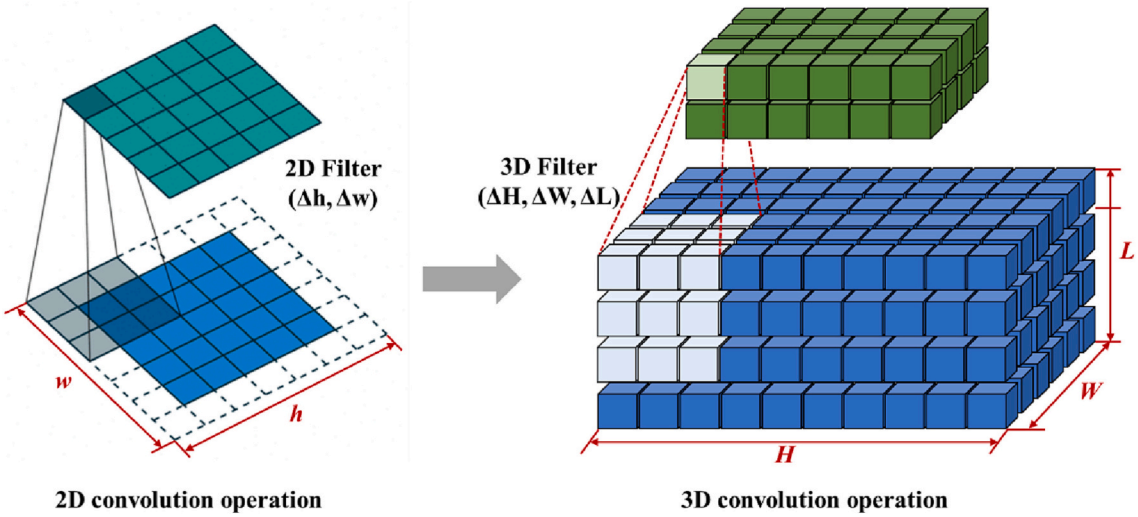


Fig. 11. Process of 2D and 3D convolution operation.

through T-Net, thus ensuring the invariance of the model to a particular spatial transformation. T-Net treated the input point cloud data as single channel images with ($B \times P \times$ dim (3 or 64) $\times$ 1). After three times of convolution and one pooling, the characteristic was reshaped to ($B \times$ 1024) and followed by two layers of full connection. In addition, for the spatial disorder characteristic of point cloud data, *PointNet* uses the Maxpooling method in extracting global features, because the calculation of maximum pooling is order independent. The mlp in *PointNet* stands for multi-layer perceptron, which is used to extract features from point clouds. The convolution with shared weights was used.

4.1.2. Class II representative network: DGCNN

Unlike *PointNet*, the other type of point cloud processing network does not process point clouds as discrete data structures. Instead, it constructs spatial topological information and utilizes graph neural networks to learn the topological relationships among the points. The representative of these networks is *DGCNN* (Dynamic Graph Convolutional Neural Networks) [36]. *DGCNN* is a network structure that combines *PointNet* and *GCNN* (Graph Convolutional Neural Networks). It inherits the permutation invariance of *PointNet* and dynamically adds a graph structure to point clouds during processing to capture local geometric features. Therefore, how to dynamically build the graph structure of the point clouds is the core of *DGCNN*. The *DGCNN* structure used in the study is shown in the Fig. 9.

The core of *DGCNN* is the EdgeConv module. *DGCNN* replaces the original mlp module with the EdgeConv module on the basis of PointNet, and replaces the feature stack of point clouds with the characterization result stack of EdgeConv. Specifically, EdgeConv constructs a dynamic graph structure for the point clouds of each network layer. For the point cloud-based graph structure, the vertices are the individual points of the point cloud, and the relationship of edges is constructed by performing k-nn (k-nearest neighbor) operations on each point to obtain its C nearest neighbors as shown in Fig. 10 above. EdgeConv takes each point of the input point clouds ($B \times P \times f$) as the center point to characterize its edge feature ($B \times P \times P$) with each neighbor, and then aggregates these features to obtain a new characterization of the point ($B \times P \times C$), so the graph structure computed by EdgeConv is dynamically updated instead of a pre-calculated fixed value. The input and output features shape of EdgeConv is sized by mlp. The unit number (a_n) of mlp in the last layer will determine the output dimension($B \times P \times a_n$) of EdgeConv. It should be noted that due to the existence of k-nn operation, the number of input point clouds should be limited. If N is too large, the matrix ($B \times P \times P$) will be huge, which will greatly increase the memory burden of computing.

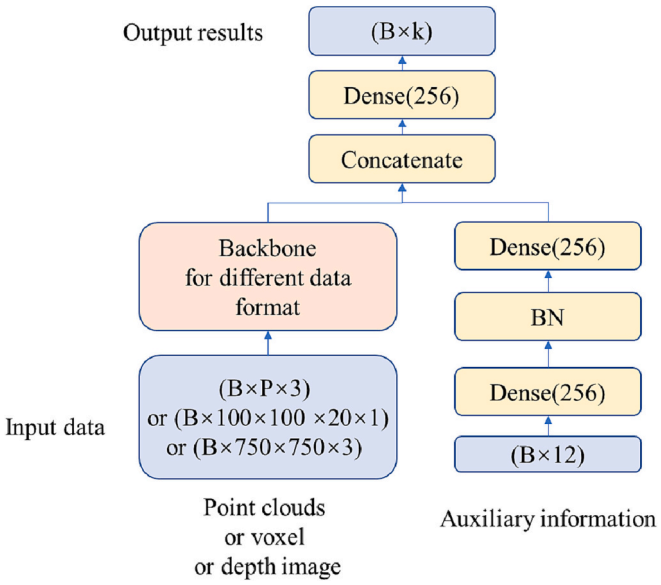


Fig. 12. The structure of multi-feature fusion network.

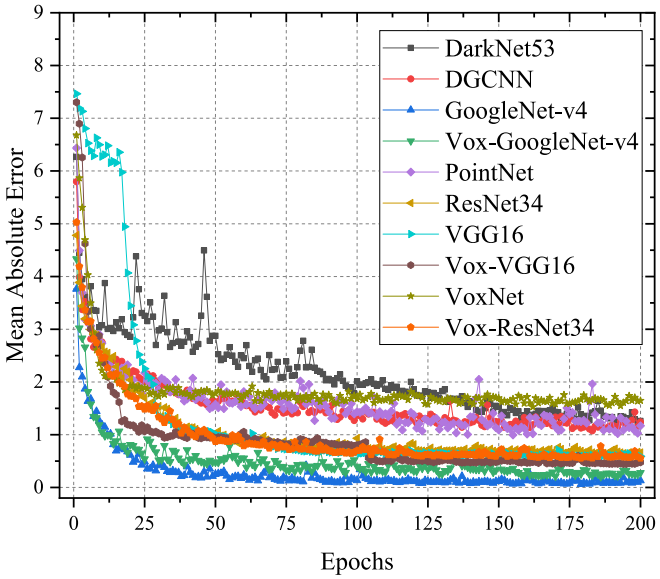


Fig. 13. The loss change of 10 networks on the ***training set*** with the training epochs.

4.2. Depth image processing network

For the processing of the depth image data after the conversion of point clouds data, the following four classes of CNNs were selected, which are *VGG16* [37], *ResNet34* [38], *Darknet53* [39], and *GoogLeNet-v4* [40]. In general, these four classes of networks cover the mainstream networks at different stages of CNN development. Although the four networks have different structures, their inputs and outputs are the same. The input tensor has the shape of ($B \times 750 \times 750 \times 3$) and the output tensor has the shape of ($B \times k$).

VGG16 is the most classic convolutional network backbone with 16 convolution layers which only includes standard convolution and pooling operations. In this study, VGG16 was employed to evaluate the feasibility of using a shallow convolutional network for addressing this research problem. For further details on the VGG16 backbone network, please refer to the research paper [37].

ResNet34 is a representative backbone network for residual block structure applications, which contains 34 convolutional layers and achieves a leap in the number of convolutional network layers. The residual block is designed to solve the degradation problem of deep network. Specifically, the residual block is implemented through shortcut connections, which concatenate the input of the block with the output after multi-layer convolution and then inputs it to the next module. Theoretically, the residual block transforms the fitting of the network to the identity mapping into the fitting of the residual, which allows the model to perform better as the number of layers increases. Similar to ResNet34, DarkNet53 also uses residual structure and includes 53 convolutional layers. DarkNet53 is the backbone used by the YOLO series network in object detection task. Due to its excellent ability to extracting image features, it is also included as one of the experimental networks in this study. For more information about ResNet34 and DarkNet53, please refer to the research paper [38,39].

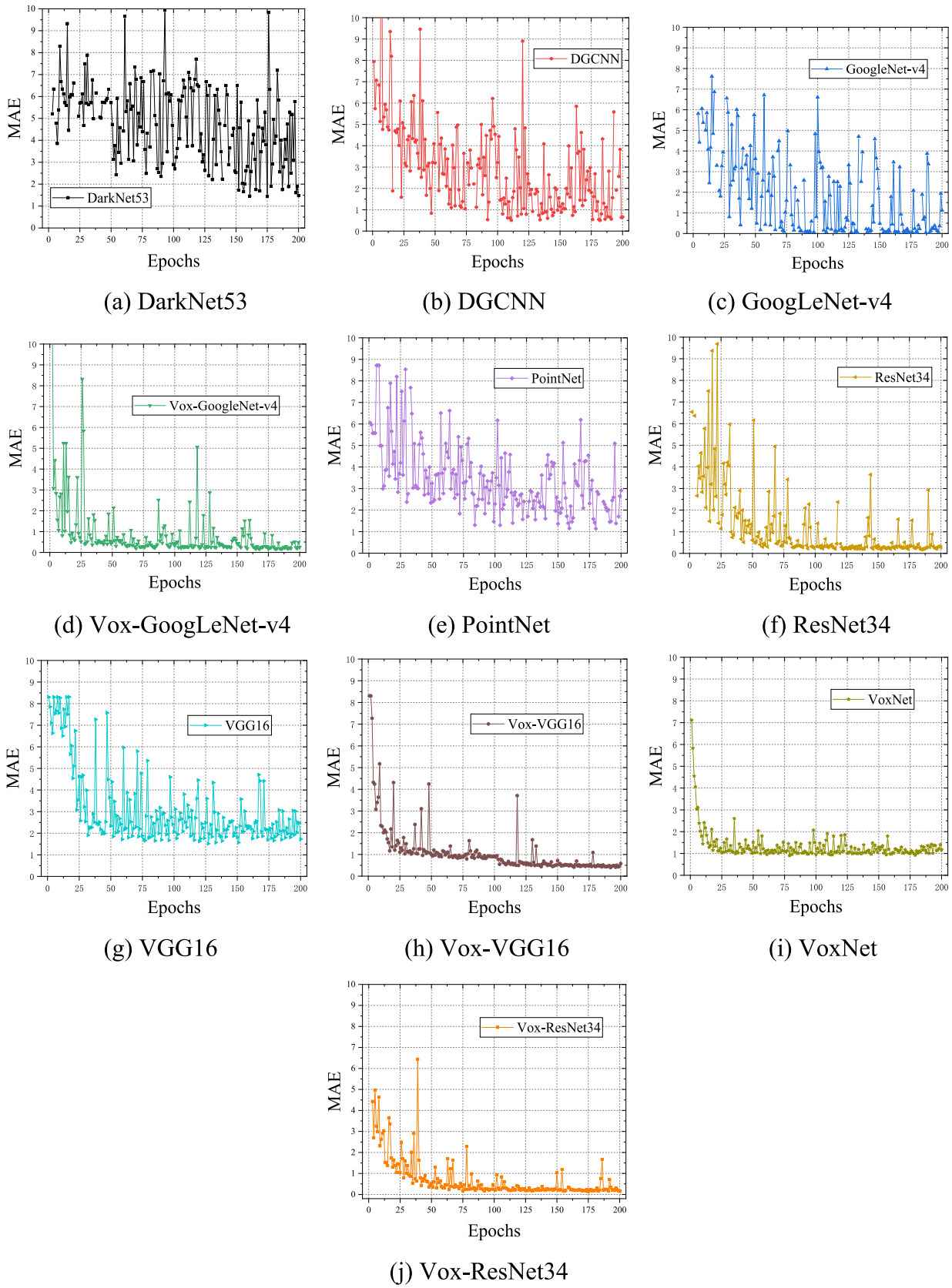
GoogLeNet is a representative backbone network that applies the network lightweight method which called Inception module. Inception is to put multiple convolution or pooling operations together to assemble as one module as shown in Fig. 10. The Inception structure in Fig. 11 uses convolutional kernels at (1×1), (3×3) and (5×5) scales, which allows the network to fuse information at different scales through multiple sensory fields, thus improving the performance of the network. In addition, this approach greatly reduces the number of parameters in the network, thus allowing the network to be used on smaller data sets without overfitting. The Inception module has been updated to the fourth version. In the GoogLeNet-v4 used in this study, the Inception v4 module was used to replace the traditional convolution layer. For more information about GoogLeNet backbone and Inception module, please refer to the research paper [40].

4.3. Voxel processing network

Voxel data can be regarded as the spatial version of image data. When processing voxel data, the traditional CNN model using 2D convolution operation are upgraded by 3D convolution operation. In this study, the following four classes of voxel CNNs were chosen, which

Table 2
The gradation of used five kinds of asphalt mixture. The number of each gradation represents the passing rate of aggregate under the corresponding size sieve.

Maximum size (mm)	AC16 (%)	AC13 (%)	SMA13 (%)	UT5 (%)	OGFC10 (%)
19	100	100	100	100	100
16	94.3	100	98.8	100	100
13.2	77.7	95	90.6	100	100
9.5	57.6	76.5	59.1	99.9	95
4.75	31.3	53	28.8	98.4	60
2.36	25.8	37	19.4	49.6	16
1.18	18	26.5	16.5	36.2	12
0.6	13.9	19	15.3	16.2	9.5
0.3	10.2	13.5	14.2	11.2	7.5
0.15	7.8	10	12.8	8.7	5.5
0.075	5.4	6	10.7	6.7	4
0.0001	0	0	0	0	0



(a) DarkNet53 (b) DGCNN (c) GoogLeNet-v4

(d) Vox-GoogLeNet-v4 (e) PointNet (f) ResNet34

(g) VGG16 (h) Vox-VGG16 (i) VoxNet

(j) Vox-ResNet34

Fig. 14. The loss change of 10 networks ***on validation set*** with the training epochs.

Table 3
The confusion matrix and loss of 10 networks under different thresholds on the test set.

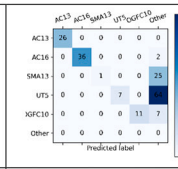
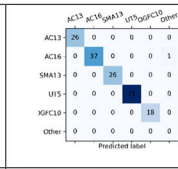
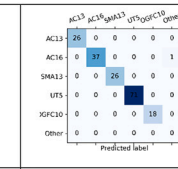
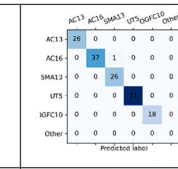
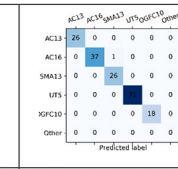
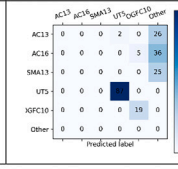
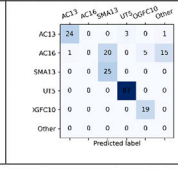
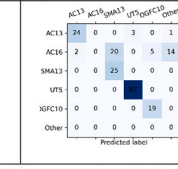
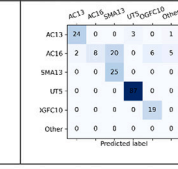
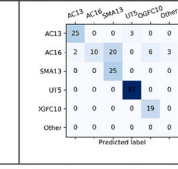
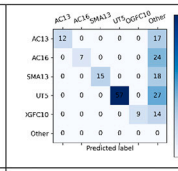
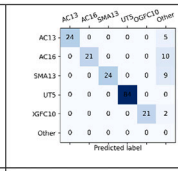
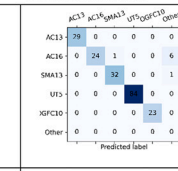
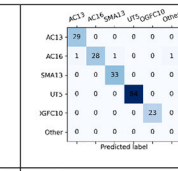
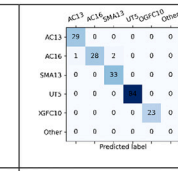
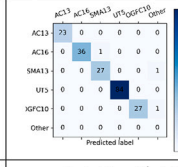
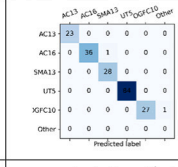
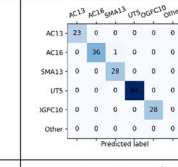
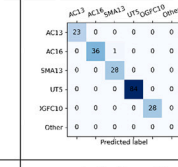
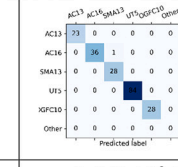
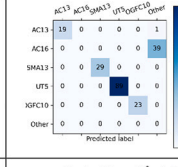
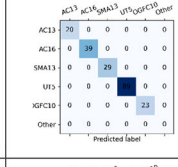
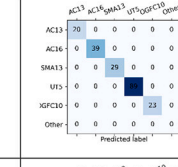
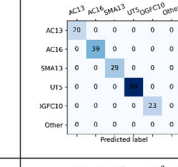
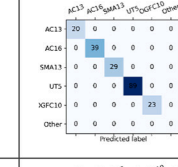
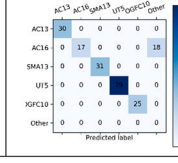
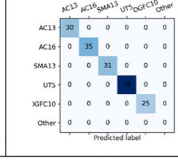
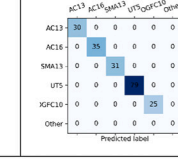
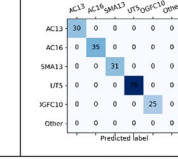
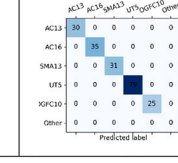
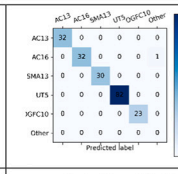
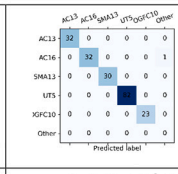
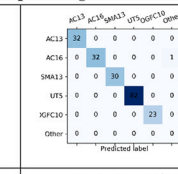
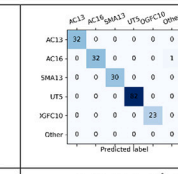
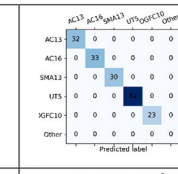
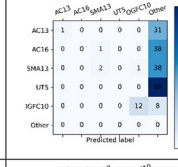
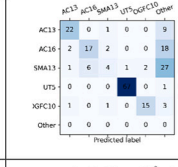
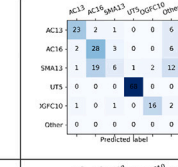
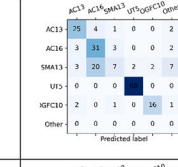
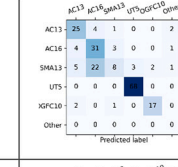
ID.	*Thr*=0.5	*Thr*=1.0	*Thr*=1.5	*Thr*=2.0	*Thr*=2.5	Test loss
Point Clouds networks						
DGCNN (B×6000×3)						0.664
PointNet (B×30000×3)						1.313
Voxels networks						
VoxNet (B×103×103×21×1)						0.630
Vox-ResNet34 (B×103×103×21×1)						0.147
Vox-VGG16 (B×103×103×21×1)						0.376
Vox- GoogLeNet-v4 (B×103×103×21×1)						0.162
Depth image networks						
ResNet34 (B×750×750×3)						0.199
VGG16 (B×750×750×3)						1.554
DarkNet53 (B×750×750×3)						1.315
GoogLeNet-v4 (B×750×750×3)						0.046

Table 4
The performance comparison of 10 Networks.

ID.	(*Thr* = 0.5)			Test loss (MAE)	Running time (ms/step)
	Precision (P)	Recall (R)	F1-score		
DGCNN (B × 6000 × 3)	1	0.539	0.700	0.664	348
PointNet (B × 30,000 × 3)	0.354	0.400	0.376	1.313	410
VoxNet (B × 103 × 103 × 21 × 1)	1	0.433	0.604	0.630	20
Vox-ResNet34 (B × 103 × 103 × 21 × 1)	1	0.980	0.990	0.147	110
Vox-VGG16 (B × 103 × 103 × 21 × 1)	0.80	0.790	0.795	0.376	308
Vox- GoogLeNet-v4 (B × 103 × 103 × 21 × 1)	1	0.897	0.946	0.162	58
ResNet34 (B × 750 × 750 × 3)	1	0.993	0.996	0.199	151
VGG16 (B × 750 × 750 × 3)	0.6	0.136	0.222	1.554	396
DarkNet53 (B × 750 × 750 × 3)	0.4	0.232	0.294	1.315	353
GoogLeNet-v4 (B × 750 × 750 × 3)	1	1	1	0.046	111

are *VoxNet* [41], *Vox-VGG16, Vox-ResNet34,* and *Vox-GoogLeNet-v4*. In the convolution operation, the convolution kernel is a sliding window, in which the weights of data in different positions are shared. In 2D convolution, the convolution kernel is 2D, and it moves in two directions of the original image to generate a new image. Similarly, in 3D convolution, the convolution kernel is a 3D block that slides in three directions on the original voxel to generate a new voxel, as shown in Fig. 11.

VoxNet is the classic 3D CNN for point clouds treatment, which introduces the volumetric occupancy grid of point clouds to represent the estimate of space occupancy and usage of a 3D convolution operation to extract the grid features. In fact, the volumetric occupancy grid of point clouds is the voxelization of point clouds. For more information about VoxNet, please refer to the research paper [40].

In addition, the CNNs mentioned in the previous section were reconstructed using 3D convolutional layers to replace the internal 2D convolutional layers to build corresponding voxelated version of these networks that *Vox-VGG16, Vox-ResNet34,* and *Vox-GoogLeNet-v4*. It should be noted that both DarkNet53 and ResNet34 were used to test the applicability of the residual block structure in this study. Only ResNet34, which has fewer layers and parameters, was selected for the study to reconstruct the voxel network.

4.4. Multi-feature fusion network

In addition to using single point clouds data for mixture gradation estimation, multi feature fusion network was proposed to study whether the auxiliary information can further improve the performance of the network. The structure of the multi feature fusion network is shown in Fig. 12.

In fact, the multi-feature fusion network is the general name of a kind of network. Its input includes two parts: the dataset (point cloud, voxel and depth map) after point cloud data conversion and the auxiliary information. In Fig. 12, the backbone network on the left adopts the network corresponding to different data format mentioned above to extract the features; the auxiliary information on the right through multiple full connection layers are combined with the features of the backbone output, and then through one full connection layer to predict the final mixture gradation. It should be mentioned that as the auxiliary information used in this study is the 12 types of indicators introduced in Section 2, the dimension of the input is (B × 12). The next section will introduce the details and conclusions of the experiment.

5. Case study

5.1. Case I: Comparison of 10 networks on the gradation extraction performance

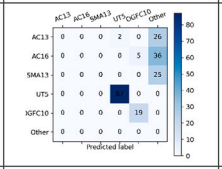
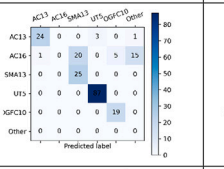
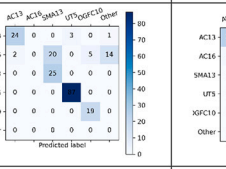
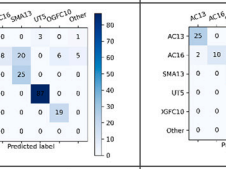
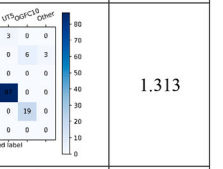
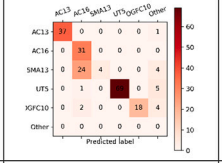
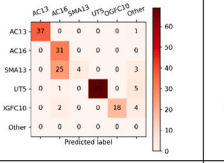
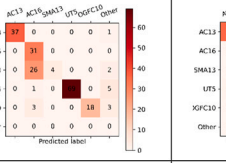
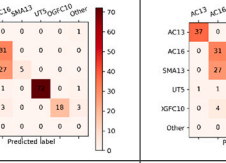
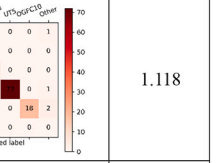
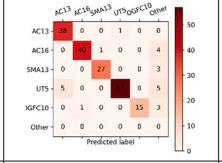
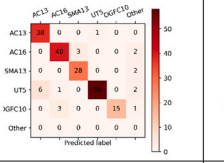
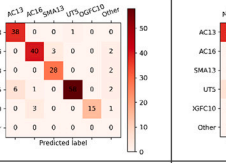
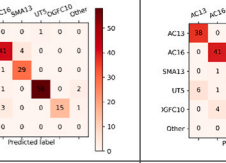
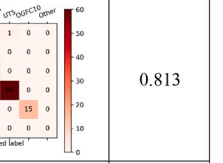
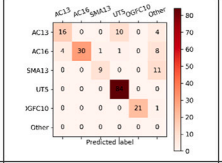
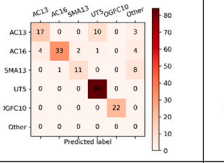
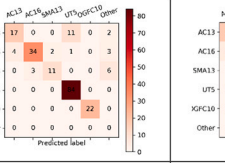
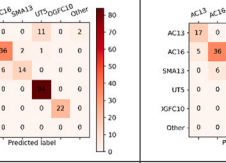
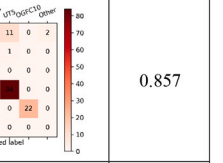
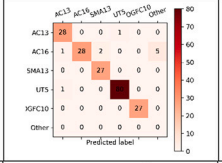
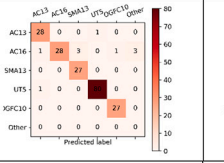
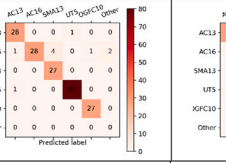
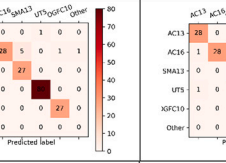
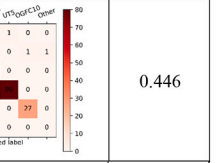
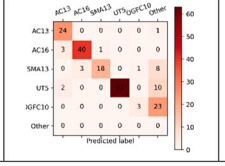
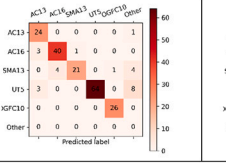
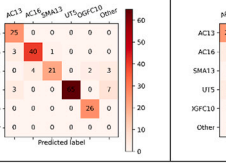
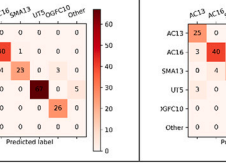
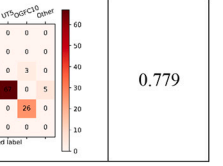
The code of the case study part was transmitted to GitHub at the following address: https://github.com/ChengjiaHanSEU/PAGE.git. The data sets used in this study were the three data sets described previously, and the number of samples in each data set was 1579. Among them, the sample sizes of the training and test sets were 1400 and 179, respectively. The sample size of the point cloud dataset was (30,000 × 3), the voxel dataset was (103 × 103 × 21 × 1) and the depth image dataset was (750 × 750 × 3). The auxiliary information was not used at the first stage. Ten networks were established, namely: PointNet, DGCNN, VGG16, ResNet34, DarkNet53, GoogLeNet-v4, VoxNet, Vox-VGG16, Vox-ResNet34 and Vox-GoogLeNet-v4. It should be noted that due to the existence of k-nn operations in DGCNN, it was impossible to use 30,000 points to construct an adjacent matrix. Therefore, the point clouds used for training DGCNN had been further downsampled with a size of (6000 × 3). These networks were built using the *Tensorflow 2.0* framework. The CPU and GPU for these networks training was the "Intel (R) Core(TM) i9-9900K CPU @ 3.60GHz" with 64GB RAM and the "NVIDIA GeForce RTX 3090". The ground truth of the dataset was the five kinds of mixture gradation corresponding to the point clouds (or voxel, depth image), as shown in Table 2.

Then, the 10 networks were trained, respectively. To maintain the comparability, the training hyperparameters of 10 networks were set as the same. Specifically, the optimizer of the network was Adam with the 1e-3 initial learning rate; the loss function was Mean Absolute Error (MAE); the batch size was 8; the validation set was divided from the training set according to the proportion of 20%, and the training epochs was 200. When training the 10 networks, the loss of the 10 networks on the training set was used to update the parameters of the networks The loss of the 10 networks was calculated on the validation set (not participating in the training) to observe the fitting degree of the networks. The loss on training set during the training process was shown in Fig. 13.

From the Fig. 13, it can be found that the 10 networks reached convergence, and the final accuracy in the training set were obviously stratified. Among them, the loss of GoogLeNet-v4 using depth image data was the smallest, and the loss of Vox-Google Net-v4 using voxel data was the second smallest, which proved that GoogLeNet using Inception module is the most suitable backbone network only from the performance of the training set. Fig. 14 showed the loss change on the validation set during the training process of the 10 networks. In general, the function of the validation set was to judge the network fitting condition. When the loss of validation set fluctuates sharply, it meant that the network fitting was difficult and the generalization was poor. From Fig. 14, it was found that the best performing GoogLeNet in the training set fluctuated sharply. This was not just a problem with GoogLeNet, actually, the DGCNN, PointNet, Darknet53, VGG16 and ResNet34 networks that used point cloud and depth image data were all sharing the same problem. In contrast, the networks Vox-GoogLeNet-v4, VoxNet, Vox-ResNet34 and Vox-VGG16 using voxel data converged relatively smoothly and achieved good accuracy in the validation set.

To intuitively test the real performance of 10 networks in the grading estimation task, we tested 10 networks on the test set of 179 samples were tested. The results were summarized in Table 3. It should be noted that since the research adopts a regression solution, the output of the network was a gradation composed of 12 values, so a threshold denoted

Table 5
The confusion matrix and loss of PointNet under different thresholds on the test set which point cloud has different size.

ID.	*Thr*=0.5	*Thr*=1.0	*Thr*=1.5	*Thr*=2.0	*Thr*=2.5	Test loss
PointNet (B×30000×3)						1.313
PointNet (B×15000×3)						1.118
PointNet (B×12000×3)						0.813
PointNet (B×9000×3)						0.857
PointNet (B×6000×3)						0.446
PointNet (B×3000×3)						0.779

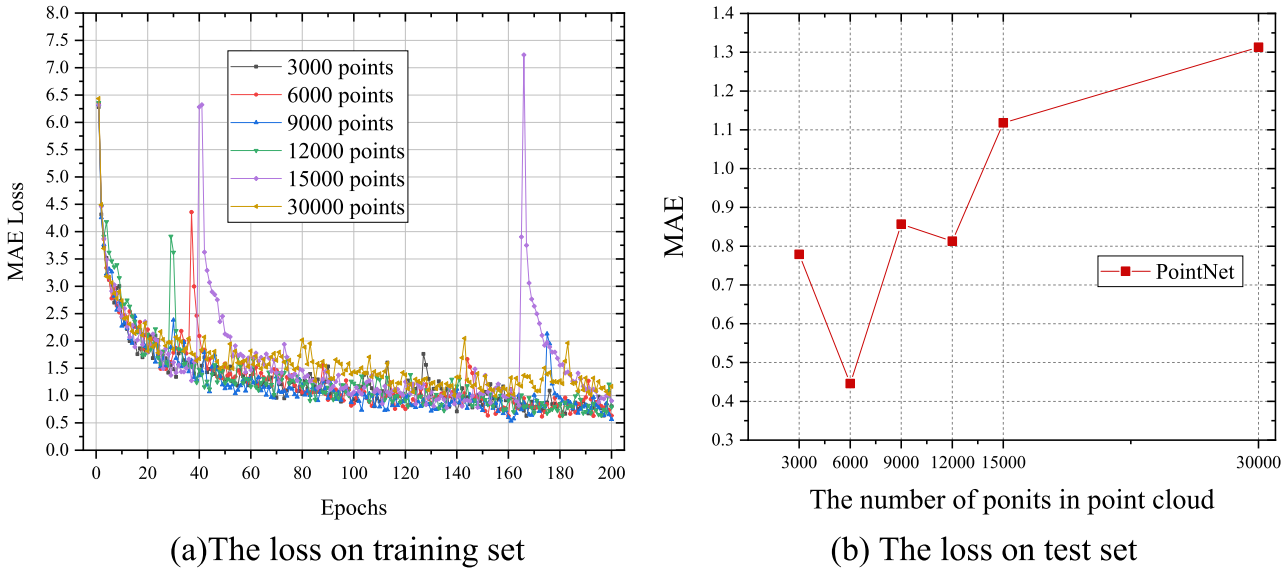
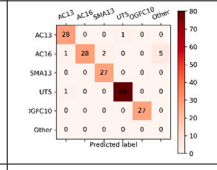
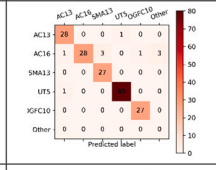
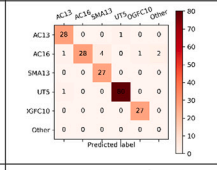
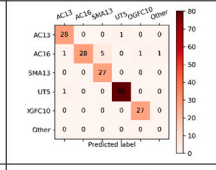
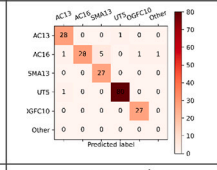
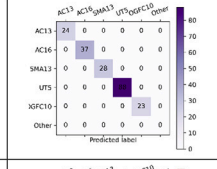
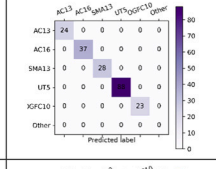
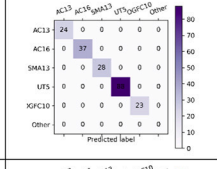
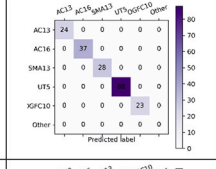
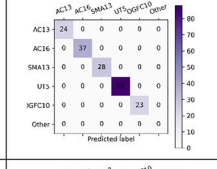
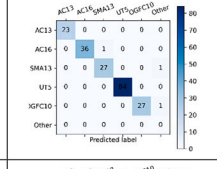
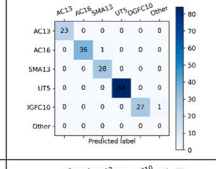
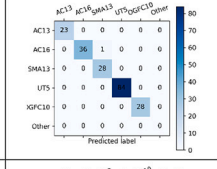
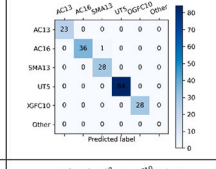
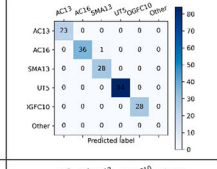
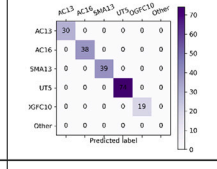
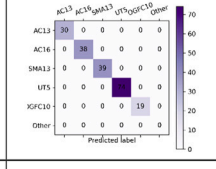
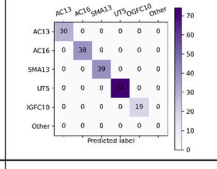
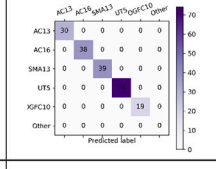
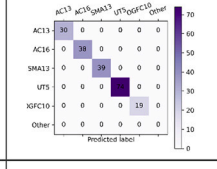
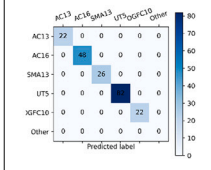
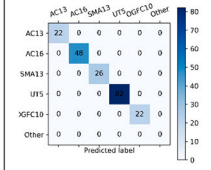
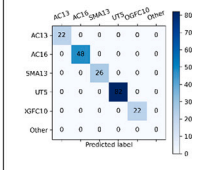
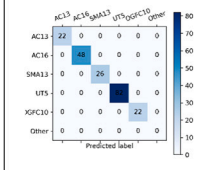
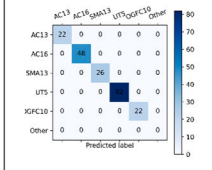
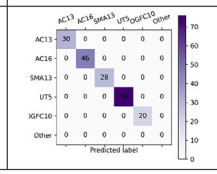
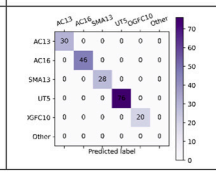
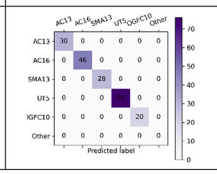
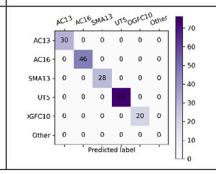
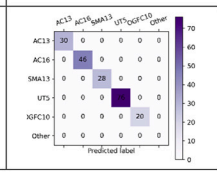


Fig. 15. The loss on training set during the training process and the loss on test set of PointNets trained on the six sizes of the point clouds.

Table 6
Two samples for 12 auxiliary indicators.

ID	S_a	S_q	S_{sk}	S_{ku}	*EMTD*	*MTD*	*Zp*0	*Zp*20	*Zp*40	*Zp*60	*Zp*80	*Zp*100
AC16–1	0.77	0.76	−0.93	4.06	0.92	0.93	−3.38	−0.49	0.02	0.36	0.68	1.28
SMA13–1	1.04	1.35	−0.93	4.07	1.33	0.95	−6.09	−0.63	0.30	0.82	1.28	2.05

Table 7
The confusion matrix and loss of PointNet, Vox-ResNet34, GoogLeNet-v4 and their corresponding multi-feature fusion networks under different thresholds on the test set.

ID.	*Thr*=0.5	*Thr*=1.0	*Thr*=1.5	*Thr*=2.0	*Thr*=2.5	Test loss
PointNet (B×6000×3)						0.446
PointNet-Aided (B×6000×3) & (B×12)						0.202
Vox-ResNet34 (B×103×103×21×1)						0.147
Vox-ResNet34-Aided (B×103×103×21×1) & (B×12)						0.142
GoogLeNet-v4 (B×750×750×3)						0.046
GoogLeNet-v4-Aided (B×750×750×3) & (B×12)						0.046

as *Thr* was needed to measure which asphalt mixture was the output gradation. The MAE was used to measure the distance between the output gradation and the target five asphalt mixture (AC16, AC13, SMA13, OGFC10 and UT5). The target mixture corresponding to the minimum distance was taken as the mixture type of output gradation. However, if the minimum MAE (the value of the minimum distance) was still higher than the predetermined *Thr*, then the output of the network was wrong and belonged to none of the five kinds of asphalt mixture. In the confusion matrix in Table 3, the predicted results that were not belonging to the five asphalt mixtures were recorded as Other. In addition, the last column of Table 3 also records the MAE loss of each network on the test set. In addition, the threshold condition with the highest classification standard were selected. In Table 3, the precision, recall and F1-scores of the 10 networks were calculated according to the confusion matrix. Table 4 summarizes these results with the MAE loss and displays processing speed of 10 networks.

From Table 3 and Table 4, it can be found that GoogLeNet was indeed the most suitable backbone network in this study. The GoogLeNet-v4, which used depth image data, achieved the best performance in 10 networks. However, the ResNet34 was also a good choice. Its performance was only a little inferior to that of GoogLeNet. Considering the limitations of the research sample size, it is difficult for the authors to assert that GoogLeNet is better than ResNet34 in this task. However, the experiment has proved an interesting conclusion that using voxel data in pavement grading estimation is the best choice. Networks using voxel data outperform networks using depth images and networks using point cloud data in terms of average performance and reliability. The reason is likely that the task of this research focuses on pavement gradation prediction, and the pavement gradation is highly sensitive to *Z*-direction data. When converting 3D data to 2D for convolution, the Z-direction

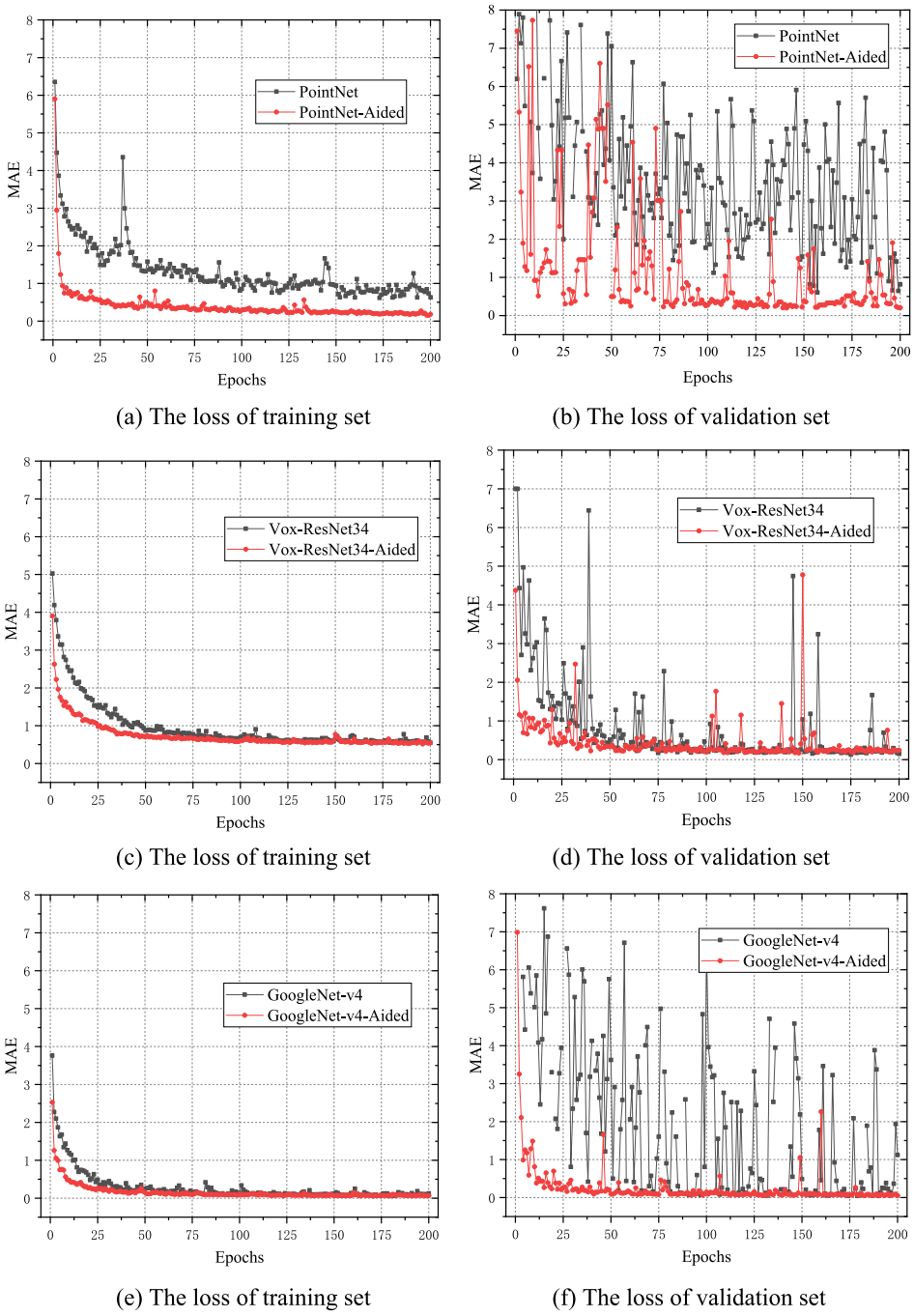


Fig. 16. The loss on training set and validation set of PointNet (a & b), Vox-ResNet34 (c & d), GoogLeNet-v4 (e & f) and their corresponding multi-feature fusion networks.

can only be characterized by 3 values (RGB) per unit pixel, so the network using depth images cannot repeatedly extract Z-direction information features as the network using voxel data, which leads to instability prediction performance. In contrast, the best performing GoogLeNet is able to integrate multi-scale receptive fields due to its Inception module. Although its input is still a two-dimensional depth image, it is able to make up for the lack of information in the Z-direction data by reusing information at multiple scales.

5.2. Case II: Effect of point cloud density on the performance of point cloud networks

An interesting situation was noticed that the DGCNN trained by (6000 × 3) point cloud performed better than the PointNet trained by (30,000 × 3) point cloud. That was against common sense, because it was reasonable that more data input meant more information, and also meant that the network trained with it can have better accuracy and generalization. Therefore, a new experiment was designed that the original point cloud dataset was downsampled according to 30,000, 15,000, 12,000, 9000, 6000 and 3000. Then the PointNet with each downsampled dataset were trained respectively. The training hyperparameters of the network remained the same as in the previous experiment except for the size of the training point clouds. The results were summarized in Table 5. Similarly, the gradation classification confusion matrix for different thresholds and the MAE loss of the networks were in Table 5. Moreover, the MAE loss on training set during

Table 8
Statistical analysis of three types of networks before and after adding auxiliary data.

ID.	(*Thr* = 0.5)			Test loss (MAE)	Running time (ms/step)	Convergence speed (epochs)	Training volatility (times)
	Precision (P)	Recall (R)	F1-score				
PointNet (B × 6000 × 3)	0.989	0.959	0.974	0.446	100	100	71
PointNet-Aided (B × 6000 × 3) & (B × 12)	1	1	1	0.202	104	50	42
Vox-ResNet34 (B × 103 × 103 × 21 × 1)	1	0.980	0.990	0.147	106	50	28
Vox-ResNet34-Aided (B × 103 × 103 × 21 × 1) & (B × 12)	1	1	1	0.142	107	25	11
GoogLeNet-v4 (B × 750 × 750 × 3)	1	1	1	0.046	112	50	69
GoogLeNet-v4-Aided (B × 750 × 750 × 3) & (B × 12)	1	1	1	0.046	115	25	8

training process and the loss on test set of PointNet trained by different size point clouds were shown in Fig. 15.

From the Table 5 and Fig. 15, it can be found that although the accuracy of PointNets trained by different size point clouds on the training set were almost the same. There was an obvious rule that the performance of PointNet was improved as the size of point clouds decreases. As the point cloud density decreased from (30,000 × 3) to (6000 × 3), the performance of PointNet gradually improved to a level similar to that of the voxel network, and its MAE loss on the test set decreased from 1.313 to 0.446. In addition, the confusion matrix at the highest threshold condition (*Thr* = 0.5) showed that the classification ability of PointNet improved from no classification ability to the level that most of the samples were correctly classified except for a few. It was believed that this was due to the disorder feature of point clouds data, which required that the density of the point cloud should not be too high. Due to the point-to-point disorder, the point cloud network cannot directly obtain a fixed spatial arrangement like voxel and image networks. Therefore, too high point cloud density leaded to the complication of local spatial features, which easily caused the point cloud network to overfit the local spatial features. When the point cloud network overly pursued the accurate extraction of spatial structure, the macroscopic feature contours that need to be addressed will be ignored, which finally caused the poor network performance. However, when the point cloud density was too small, the loss of spatial feature details of data also leaded to the performance degradation of the point cloud network. For example, when the point cloud density decreased from (6000 × 3) to (3000 × 3), the loss of PointNet on the test set increased from 0.446 to 0.779. To summarize, the influence of point cloud density on the performance of point cloud network was in an inverted "V" shape which had an extreme point that represented the point cloud density making the network have the best performance. The best point cloud density needs to be determined according to the training set and research questions. The recommended optimal point cloud density in this study was 6000 points to characterize an asphalt pavement area of 51 × 51 mm^2.

5.3. Case III: Effect of auxiliary information fusion on the performance of networks

Next, an experiment based on the first two experiments was used to verify whether the auxiliary information can further improve the performance of pavement grading estimation network. As mentioned in Section 2, the auxiliary information that 12 indicators for each training samples was collected. Two samples for 12 indicators are shown in Table 6. The sample 1 is the additional information of AC16–1 original point cloud, and the sample 2 is the additional information of SMA13–1 original point cloud.

First, the PointNet, Vox-ResNet34 and GoogLeNet-v4 were selected to build three multi-feature fusion networks (mentioned in Sections 4.4), namely PointNet-Aided, Vox-ResNet34-Aided and GoogLeNet-v4-Aided. The same training hyperparameters as the previous experiments were used to train the new three multi-feature fusion networks and compared their results with the networks without the addition of auxiliary information, which was shown in Table 7. In addition, the variation of MAE loss on the training set and the validation set for these 6 networks during the training process were shown in Fig. 16.

From the Table 7, Table 8 and Fig. 16, it can be found that the network performance of the three data formats becomes better after auxiliary information fusion. This change is reflected in three factors. The first factor was the prediction accuracy improvement. With the integration of auxiliary information, on the test set, the loss of PointNet decreased from 0.446 to 0.202, the loss of Vox ResNet34 decreased from 0.147 to 0.142, and that of Google Net-v4 was not further reduced. Moreover, the P, R and F1-score of the three networks have improved after adding auxiliary information. It showed that the fusion of auxiliary information can help the network to improve the prediction accuracy, and the worse the original performance of the network, the more obvious this improvement was, but this improvement cannot break the upper limit of network performance bounded by the quality of the data set.

The second factor was the acceleration of model convergence. From Fig. 16, comparing the change in loss of the three networks with their corresponding multi-feature fusion networks on the training set, it can be found that the networks using the auxiliary information (the red line in Fig. 16(a), (c) and (e)) had an obvious earlier drop in loss than that of the original networks (the black line in Fig. 16(a), (c) and (e)). With the integration of auxiliary information, on the test set, the convergence epoch of PointNet decreased from 100 to 50, that of Vox ResNet34 and Google Net-v4 decreased from 50 to 25.

The third factor was that the stability of the networks fitting process improved greatly. From Fig. 16(b), (d) and (f), it was obvious that the multi-feature fusion networks had a much smoother loss descent process (the red line) on the validation set than that of the original networks (the black line). As stated before, the fluctuation of the validation set was related to the fitting degree, and the addition of auxiliary information made the network obtain more reference data, which help the network less likely to fall into overfitting. In Table 8, an index to measure the volatility of the training process was defined. It counts the number of times that the loss of the validation set has increased by more than 50% compared with the previous epoch. With the integration of auxiliary information, on the validation set, the training volatility of PointNet decreased from 71 to 42, that of Vox ResNet34 decreased from 28 to 11, and that of Google Net-v4 decreased from 69 to 8. Comprehensive experimental results show that adding auxiliary data will consistently improve the performance of three different networks.

For a more visual comparison of the gradation predictions before and after the auxiliary information fusion, the final predicted asphalt mixture gradation of these networks was compared. It should be noted that the direct prediction result of the network is the proportion of aggregate retained in 12 size sieves. Specifically, the output tensor dimension of the last layer of all used networks is ($B \times k$), where B is the batch size and k is taken as 12, indicating the proportion of the aggregate on the 12 size sieves. The SoftMax function is used as the activation

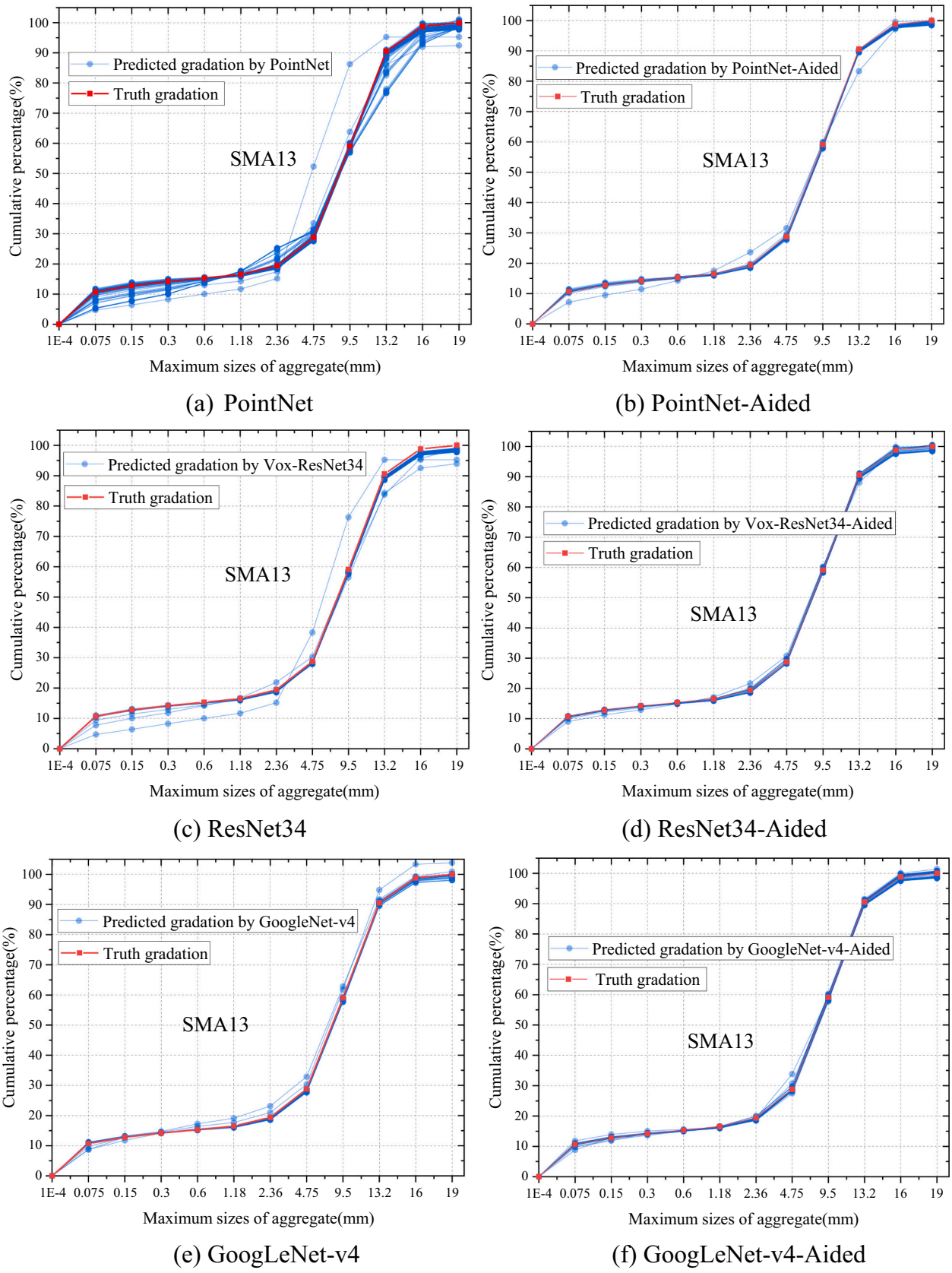


Fig. 17. Comparison of the real gradation and the predicted gradation of PointNet, Vox-ResNet34, GoogLeNet-v4 and their corresponding multi-feature fusion networks.

function of the network output layer, which is calculated as

$$SoftMax(x)_i = \frac{e^{x_i}}{\sum_{k=1}^{12} e^{x_k}}. \quad (4)$$

Therefore, the predicted output of the network is the percentage of retained aggregate of 12 size sieves, and the sum of the 12 values is 100%. After that, we only need to accumulate the predicted aggregate percentages of 12 sizes sieves to get the passing rate of aggregate under the 12 size sieves, which is the final asphalt mixture gradation curve, and its form is the same as that in Table 2. The samples with SMA13 gradation were selected for prediction, and the comparison of the real gradation and the predicted gradation curve of PointNet, Vox-ResNet34, GoogLeNet-v4 and their corresponding multi-feature fusion networks were shown in Fig. 17. In Fig. 17, the blue curve represented the predicted gradation, and the red curve represented the true gradation shown in the fourth column of Table 2. The results visually demonstrated that the multi-feature fusion network had better prediction accuracy than the network using a single point cloud, voxel or depth image data.

6. Conclusions

This study proposed a rapid estimation framework of asphalt pavement gradation based on point cloud data. Considering the lack of comprehensive analysis of point cloud data characterization in the current related research, this study has carried out a comprehensive study from point cloud denoising, data enhancement to point cloud data type conversion, and the fusion application of auxiliary information and point cloud data. In particular, the proposed point cloud data and auxiliary information fusion network has reached the state-of-the-art in the current pavement aggregate gradation extraction method. The conclusions are as follows:

(1) The study explores how to use deep learning methods to process road point cloud data for non-destructive and rapid estimation of asphalt mixture gradation in constructed roads. The case study demonstrates that the proposed neural network model achieves a 95% accuracy in gradation estimation, meeting the requirements for quality control and assurance in asphalt pavement construction and maintenance processes.
(2) Addressing the engineering challenge of high precision requirements for pavement gradation estimation from point cloud data, which leads to slow collection speed and small sample sizes of the original point cloud samples. A data enhancement algorithm for the original acquisition point clouds was proposed, which enhanced the representative raw point clouds into multiple sub-point clouds by optimal segmentation size estimation to improve the number and quality of training samples.
(3) Three rapid estimation methods for pavement gradation based on point clouds were proposed in the study. These methods involve converting the original point cloud data into three different forms: aligned point cloud, voxel, and depth image. Furthermore, ten neural network models were developed for training using these three data forms. The experiment showed that GoogLeNet with inception module using depth images had the best performance with a MAE loss of 0.046. Moreover, neural networks utilizing voxel data were found to be more suitable for the rapid estimation task of pavement gradation compared to those using point cloud and depth image. This is due to their more stable and accurate performance, as demonstrated by the average F1-score, average MAE loss, and confusion matrix.
(4) It has been observed that the density of the point cloud has an impact on the performance of point cloud networks in estimating pavement gradation. The relationship between point cloud density and network performance is inverted V-shaped, i.e., there is an extreme point where the performance of the point cloud network is optimal. According to the experiment results, we recommended using 6000 points to characterize the asphalt pavement for 51 × 51 mm^2.
(5) A construction method for a multi-feature fusion network was proposed for rapid estimation of pavement gradation. This method utilizes the point cloud network, voxel network, or depth image network as the backbone and combines the auxiliary information with the output of these backbones through shallow mapping. The experiment showed that, with the integration of auxiliary information, the loss of PointNet, ResNet34 decreased from 0.446 and 0.147 to 0.202 and 0.142, respectively; the convergence epoch of PointNet, ResNet34 and Google Net-v4 decreased from 100, 50 and 50 to 50, 25 and 25, respectively; and the training volatility of PointNet, ResNet34 and Google Net-v4 decreased from 71, 28 and 69 to 42, 11 and 8, respectively. It demonstrated that the multi-feature fusion network had been greatly improved in prediction accuracy, training process convergence speed and training process fitting stability compared with their backbone networks.

However, there are also some deficiencies which need further research in the future. First of all, the samples in the used dataset are not enough, which requires continuous collection to expand the samples from two aspects of gradation types and quantity. In addition, four of the current 11 original point clouds represent “UT5” gradation, and the dataset has a certain sample imbalance problem. Moreover, there is a certain degree of experience and subjectivity in the determination of the threshold of data enhancement, voxel grid and depth image size. Further experiments need to be carried out to optimize the threshold selection method the determine the impact of grid or size division on network performance. Besides, the research will further explore the implementation of multimodal end-to-end network, so that the network can directly select the most appropriate data format according to the input point cloud obtained from original collection, and complete the aggregate gradation prediction task in an end-to-end structure.

Declaration of Competing Interest

The authors declare that there is no conflict of interests regarding the publication of this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

This paper is part of the research work of National Key Research and Development Project of China (Grant No. 2021YFB2600601, 2021YFB2600600). The authors would like to acknowledge the financial support provided by the National Natural Science Foundation of China (Grant No. 51922030), Natural Science Foundation of Jiangsu (Grant No. BK20220845), “the Fundamental Research Funds for the Central Universities” (Grant No. 2242022R10019). This research is also supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-TC-2021-001).

References

[1] M. Arabani, Effect of glass cutlet on the improvement of the dynamic behaviour of asphalt concrete, Constr. Build. Mater. 25 (3) (2011) 1181–1185, https://doi.org/10.1016/j.conbuildmat.2010.09.043.
[2] O. Xu, Z.J. Wang, R. Wang, Effects of aggregate gradations and binder contents on engineering properties of cement emulsified asphalt mixtures, Constr. Build. Mater. 135 (2017) 632–640, https://doi.org/10.1016/j.conbuildmat.2016.12.095.

[3] B. McCabe, S. AbouRizk, J. Gavin, Time of sampling strategies for asphalt pavement quality assurance, J. Constr. Eng. Manag. 128 (1) (2002) 85–89, https://doi.org/10.1061/(ASCE)0733-9364(2002)128:1(85).
[4] T.M. Breakah, J.P. Bausano, R.C. Williams, S. Vitton, The impact of fine aggregate characteristics on asphalt concrete pavement design life, Int. J. Pavement Eng. 12 (2) (2011) 101–109, https://doi.org/10.1080/10298430903578937.
[5] B. Lira, J. Ekblad, R. Lundstrom, Evaluation of asphalt rutting based on mixture aggregate gradation, Road Mater. Pavement Design. 22 (5) (2021) 1160–1177, https://doi.org/10.1080/14680629.2019.1683061.
[6] J.C. Guan, X. Yang, V.C.S. Lee, W.B. Liu, Y. Li, L. Ding, B. Hui, Full field-of-view pavement stereo reconstruction under dynamic traffic conditions: incorporating height-adaptive vehicle detection and multi-view occlusion optimization, Autom. Constr. 144 (2022), 104615, https://doi.org/10.1016/j.autcon.2022.104615.
[7] S.Y. Chen, S. Adhikari, Z.P. You, Relationship of coefficient of permeability, porosity, and air voids in fine-graded HMA, J. Mater. Civ. Eng. 3 (1) (2019) 04018359, https://doi.org/10.1061/(ASCE)MT.1943-5533.0002573.
[8] K. Liu, P.X. Xu, F. Wang, L.Y. You, X.C. Zhang, C.L. Fu, Chaoliang, Assessment of automatic induction self-healing treatment applied to steel deck asphalt pavement, Autom. Constr. 133 (2022), 104011, https://doi.org/10.1016/j.autcon.2021.104011.
[9] M.R. Ganji, A. Golroo, H. Sheikhzadeh, A. Ghelmani, M.A. Bidgoli, Dense-graded asphalt pavement macrotexture measurement using tire/road noise monitoring, Autom. Constr. 106 (2019), 102887, https://doi.org/10.1016/j.autcon.2019.102887.
[10] X.L. Li, S.Y. Chen, K.Y. Xiong, X.Y. Liu, Gradation segregation analysis of warm mix asphalt mixture, J. Mater. Civ. Eng. 30 (4) (2018) 04018027, https://doi.org/10.1061/(ASCE)MT.1943-5533.0002208.
[11] D.D. Ge, Z.P. You, S.Y. Chen, C.C. Liu, J.F. Gao, S.T. Lv, The performance of asphalt binder with trichloroethylene: improving the efficiency of using reclaimed asphalt pavement, J. Clean. Prod. 232 (2019) 205–212, https://doi.org/10.1016/j.jclepro.2019.05.164.
[12] H.N. Xu, Y.Q. Tan, X.A. Yao, X-ray computed tomography in hydraulics of asphalt mixtures: procedure, accuracy, and application, Constr. Build. Mater. 108 (2016) 10–21, https://doi.org/10.1016/j.conbuildmat.2016.01.032.
[13] H.D. Yang, H.Y. Ju, T. Ma, Z. Tong, C.J. Han, T.Y. Xie, Novel computer tomography image enhancement deep neural networks for asphalt mixtures, Constr. Build. Mater. 352 (2022), 129067, https://doi.org/10.1016/j.conbuildmat.2022.129067.
[14] Y. Gao, K. Hou, Y.S. Jia, Z.Y. Wei, S.Q. Wang, Z.R. Li, F. Ding, X.W. Gong, Variability evaluation of gradation for asphalt mixture in asphalt pavement construction, Autom. Constr. 128 (2021), 103742, https://doi.org/10.1016/j.autcon.2021.103742.
[15] Y.C. Wang, B. Yu, X.Y. Zhang, J. Liang, Automatic extraction and evaluation of pavement three-dimensional surface texture using laser scanning technology, Autom. Constr. 141 (2022), 104410, https://doi.org/10.1016/j.autcon.2022.104410.
[16] Y. Ma, S. Easa, J.C. Cheng, B. Yu, Automatic framework for detecting obstacles restricting 3D highway sight distance using mobile laser scanning data, J. Comput. Civ. Eng. 35 (4) (2021), https://doi.org/10.1061/(ASCE)CP.1943-5487.0000973.
[17] J.W. Liu, X. Yang, S. Lau, X. Wang, S. Luo, V.C.S. Lee, L. Ding, Automated pavement crack detection and segmentation based on two-step convolutional neural network, Comput. Aided Civil Infrastruct. Eng. 35 (11) (2020) 1291–1305, https://doi.org/10.1111/mice.12622.
[18] X.Y. Zhu, Y. Yang, H.D. Zhao, D. Jelagin, F. Chen, F.A. Gilabert, A. Guarin, Effects of surface texture deterioration and wet surface conditions on asphalt runway skid resistance, Tribol. Int. 153 (2021), 106589, https://doi.org/10.1016/j.triboint.2020.106589.
[19] S.H. Dong, S. Han, C. Wu, O.M. Xu, H.Y. Kong, Asphalt pavement macrotexture reconstruction from monocular image based on deep convolutional neural network, Comput. Aided Civil Infrastruct. Eng. 37 (13) (2022) 1754–1768, https://doi.org/10.1111/mice.12878.
[20] H.Y. Ju, W. Li, S. Tighe, Z.Y. Sun, H.C. Sun, Quantitative analysis of macrotexture of asphalt concrete pavement surface based on 3D data, Transp. Res. Rec. 2674 (8) (2020) 732–744, https://doi.org/10.1177/0361198120920269.
[21] Z.H. Weng, G. Ablat, D.F. Wu, C.L. Liu, F. Li, Y.C. Du, J. Cao, Rapid pavement aggregate gradation estimation based on 3D data using a multi-feature fusion network, Autom. Constr. 134 (2022), 104050, https://doi.org/10.1016/j.autcon.2021.104050.
[22] M. Medeiros, L. Babadopulos, R. Maia, V.C. Branco, 3D pavement macrotexture parameters from close range photogrammetry, Int. J. Pavement Eng. Early Access (2021), https://doi.org/10.1080/10298436.2021.2020784.
[23] Y.D. Niu, S.X. Zhang, G.J. Tian, H.B. Zhu, W. Zhou, Estimation for runway friction coefficient based on multi-sensor information fusion and model correlation, Sensors. 20 (14) (2020) 3886, https://doi.org/10.3390/s20143886.
[24] C. Plati, M. Pomoni, Impact of traffic volume on pavement macrotexture and skid resistance long-term performance, Transp. Res. Rec. 2673 (2) (2019) 314–322, https://doi.org/10.1177/0361198118821343.
[25] Y.Y. Wang, Z.Q. Yang, Y.Y. Liu, L. Sun, The characterisation of three-dimensional texture morphology of pavement for describing pavement sliding resistance, Road Mater. Pavement Design. 20 (5) (2019) 1076–1095, https://doi.org/10.1080/14680629.2018.1433710.
[26] J.Y. Chen, X.M. Huang, B.S. Zheng, R.M. Zhao, X.Y. Liu, Q.Q. Cao, S.Z. Zhu, Real-time identification system of asphalt pavement texture based on the close-range photogrammetry, Constr. Build. Mater. 226 (2019) 910–919, https://doi.org/10.1016/j.conbuildmat.2019.07.321.
[27] M.M. Kanafi, A.J. Tuononen, Top topography surface roughness power spectrum for pavement friction evaluation, Tribol. Int. 107 (2017) 240–249, https://doi.org/10.1016/j.triboint.2016.11.038.
[28] Y.C. Du, Z.H. Weng, F. Li, G. Ablat, D.F. Wu, C.L. Liu, A novel approach for pavement texture characterisation using 2D-wavelet decomposition, Int. J. Pavement Eng. 23 (6) (2022) 1851–1866, https://doi.org/10.1080/10298436.2020.1825712.
[29] J.N. Meegoda, G.M. Rowe, A.A. Jumikis, C.H. Hettiarachchi, N. Bandara, N. C. Gephart, Estimation of surface macrotexture in hot mix asphalt concrete pavements using laser texture data, J. Test. Eval. 33 (5) (2005) 305–315, https://doi.org/10.1520/JTE12343.
[30] A.F.T. Martins, N.A. Smith, E.P. Xing, P.M.Q. Aguiar, M.A.T. Figueiredo, Nonextensive information theoretic kernels on measures, J. Mach. Learn. Res. 10 (2009) 935–975, https://doi.org/10.1145/1577069.1577104.
[31] C. Zhang, H.C. Wan, X.Y. Shen, Z.Z. Wu, PVT: point-voxel transformer for point cloud learning, Int. J. Intel. Syst. Early Access (2022), https://doi.org/10.1002/int.23073.
[32] X. Zhang, G. Cheung, J.H. Pang, Y. Sanghvi, A. Gnanasambandam, S.H. Chan, Graph-based depth denoising & dequantization for point cloud enhancement, IEEE Trans. Image Process. 31 (2022) 6863–6878, https://doi.org/10.1109/TIP.2022.3214077.
[33] C.R. Qi, H. Su, K.C. Mo, L.J. Guibas, PointNet: deep learning on point sets for 3D classification and segmentation, in: Proceedings of the 30th IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 77–85, https://doi.org/10.1109/CVPR.2017.16.
[34] C.R. Qi, L. Yi, H. Su, L.J. Guibas, PointNet plus plus: deep hierarchical feature learning on point sets in a metric space, in: Proceedings of the 31st Annual Conference on Neural Information Processing Systems (NIPS), 2017, https://doi.org/10.48550/arXiv.1706.02413.
[35] P. Cao, H. Chen, Y. Zhang, G. Wang, Multi-view frustum pointnet for object detection in autonomous driving, in: Proceedings of the 26th IEEE International Conference on Image Processing (ICIP), 2019, pp. 3896–3899, https://doi.org/10.1109/ICIP.2019.8803572.
[36] A.V. Phan, M.L. Nguyen, Y.L.H. Nguyen, L.T. Bui, DGCNN: a convolutional neural network over large-scale labeled graphs, Neural Netw. 108 (2018) 533–543, https://doi.org/10.1016/j.neunet.2018.09.001.
[37] R. Girshick, Fast R-CNN, in: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), 2015, pp. 1440–1448, https://doi.org/10.1109/ICCV.2015.169.
[38] K.M. He, X.Y. Zhang, S.Q. Ren, J. Sun, Identity mappings in deep residual networks, in: Proceedings of the 14th European Conference on Computer Vision (ECCV). 9908, 2016, pp. 630–645, https://doi.org/10.1007/978-3-319-46493-0_38.
[39] J. Redmon, S. Divvala, R. Girshick, A. Farhadi, You only look once: unified, real-time object detection, in: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779–788, https://doi.org/10.1109/CVPR.2016.91.
[40] C. Szegedy, S. Ioffe, V. Vanhoucke, A.A. Alemi, Inception-v4, Inception-ResNet and the impact of residual connections on learning, in: Proceedings of the 31st AAAI Conference on Artificial Intelligence, 2017, pp. 4278–4284, https://doi.org/10.48550/arXiv.1602.0726.
[41] Y. Zhou, O. Tuzel, VoxelNet: end-to-end learning for point cloud based 3D object detection, in: Proceedings of the 31st IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2018, pp. 4490–4499, https://doi.org/10.1109/CVPR.2018.00472.