CSC8499 Individual Project: Classification and Completion of Ancient Pottery Based on Deep Learning

Xuening Li

MSc in Advanced Computer Science,
School of Computing Science, Newcastle University.
x.li144@newcastle.ac.uk

Abstract. With the rapid development of computer vision research, the recognition, classification and completion of 3D graphics have become an important research direction. The development process has gone through a process from simple feature extraction-based methods to complex applications such as generative adversarial networks. Relevant researchers have used large datasets such as ShapeNet and ModelNet to conduct multiple experiments and achieved remarkable results. The practical content of this project is to apply three deep learning network structures to the multi-classification and completion tasks of ancient pottery in 3D point cloud data format, and to organize and evaluate the experimental results. The dataset consists of 8 different categories of 3D scanned graphics from different archaeological or cultural relics protection institutions. Each graphic sample has a corresponding damaged version manually produced. These data are used for model training after a series of preprocessing. For the classification task, PointNet/PointNet++, the most classic and commonly used network model for point cloud feature extraction and classification tasks, was selected. The model used for completion selected two more complex network architectures based on PointNet-PCN and PF-Net. Through multi-dimensional evaluation of the experimental results, it was found that under the experimental conditions of this project, the classification model PointNet/PointNet++ showed good accuracy in the multi-classification task of different categories of broken pottery, while the completion model PCN was more practical and reference-oriented in terms of overall performance than PF-Net. Overall, this project verified the potential of modern deep learning methods in the field of cultural heritage protection by applying them to the multi-classification and completion of ancient pottery. The experimental results not only provide technical support for pottery restoration, but also provide reference for other similar cultural heritage digital protection projects.

Declaration: I declare that this dissertation represents my own work except where otherwise explicitly stated.

1 Introduction

Pottery is one of the most important cultural heritages in human history. It has a simple and elegant appearance, rich and varied shapes, and carries a wealth of historical information. It is a precious cultural relic that combines aesthetic value, practical value, and historical value. As an important cultural heritage protection object, pottery has long been the focus of public and scholars. In recent years, with the rapid development of archaeology, pottery representing different historical cultures has been unearthed all over the world. However, due to the special nature of the material, some pottery has been damaged to varying degrees due to natural or human factors [1], which brings great challenges to the classification and restoration of excavated samples.

Traditional classification and restoration methods require relevant personnel to have rich experience and professional knowledge, but there is currently a shortage of qualified personnel, and the effectiveness and efficiency of related work cannot be guaranteed [2]. A more serious problem is that manual operation is highly subjective, prone to errors, and may even lead to irreparable losses, thereby reducing the quality of cultural heritage preservation. Therefore, there is an urgent need for a more efficient and accurate restoration technology to assist or partially replace the existing methods.

In the field of computer vision, 3D graphics classification and completion algorithms based on deep learning provide an effective solution to this problem [3,4]. First, network models such as PointNet/PointNet++ have powerful feature extraction capabilities and can extract useful geometric and structural features from three-dimensional images of pottery (such as point cloud data). These features can be used as the basis for automatic classification of pottery. Furthermore, these features can also help the algorithm understand and reconstruct the overall shape of pottery (such as completion models such as PCN), and then generate a complete pottery shape or complete missing parts. Figure 1 is the schematic diagram based on this solution.

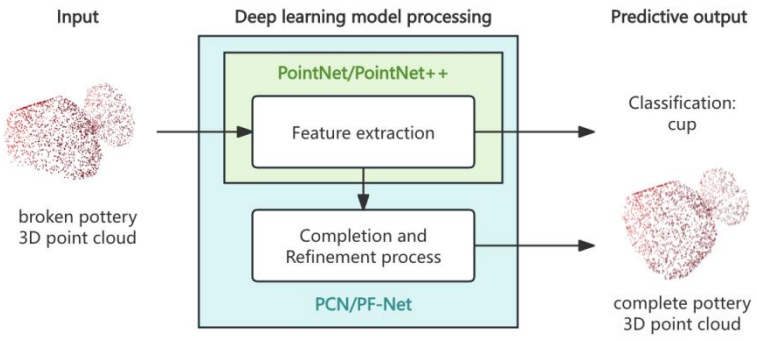


Figure 1. Schematic of the process of applying specific functional modules of the deep learning model to the classification and complementation of broken pottery.

In addition, with the popularization of 3D scanning technology in the field of historical and cultural relics protection [5], it has become easier to scan and process three-dimensional pottery of various shapes, which provides conditions for the extensive collection of training data. The improvement of hardware storage and computing power also provides support for model training. These technological advances have laid the foundation for the application of 3D graphics completion algorithms in pottery restoration. However, there are currently few public resources for high-quality 3D images of pottery, which is one of the reasons why the research and application of AI technology in this field is not yet widespread.

Compared with the traditional manual restoration model, the deep learning-based solution will have many advantages:

1. Improve accuracy. The 3D graphics-based restoration method can capture and reconstruct complex geometric details, providing higher accuracy than traditional manual restoration.
2. Improve efficiency. By using machine learning algorithms to achieve automation and efficient processing, the damaged areas can be automatically identified and repaired, which greatly improves the restoration efficiency and reduces the time and cost of manual intervention.
3. To deal with complex damage situations, this technology can handle complex geometric shapes and structures, especially suitable for severely damaged and complex pottery.
4. Generalization ability of the model: Through training, the model can be applied to pottery of different types and samples, with strong generalization ability and adaptability to a variety of restoration scenarios.
5. Better protection of the integrity of cultural relics, reducing direct contact with physical cultural relics, reducing the risk of secondary damage, and thus better protecting the integrity of cultural relics.
6. With research and educational significance, the restored digital model can not only be used for academic research, but also through virtual museums and online exhibitions [6], so that more people can access and understand these cultural heritages, and enhance the public's cultural awareness and interest.

1.1 Aim

The aim of this project is to explore the feasibility of applying 3D graphic classification and complementation algorithms based on deep learning to the automatic classification and broken complementation work of pottery, a specific cultural artefact, to obtain more ideal results and valuable evaluation results, and to accumulate the complete engineering implementation experience of deep learning in the field of cultural artefacts restoration, which will provide an important reference for the application of artificial intelligence in the field of cultural heritage protection.

1.2 Objectives

Based on the aim and the expected objectives, the project made several contributions Combined with the expected objectives, this project has made the following contributions:

1. Extensive investigation, understanding and organization of existing work related to the application of AI technology in the classification and restoration of cultural relics, providing a sufficient theoretical basis for the implementation of the project.
2. According to the existing resources and project requirements, a new comprehensive 3D dataset of ancient classic pottery types, 3D Pottery 8, was hand-made, and corresponding simulated damaged samples were hand-made for each sample.
3. The 3D point cloud classification model PointNet was used for the classification task of the 3D Pottery 8 dataset to obtain the best model under the current conditions.
4. The 3D point cloud completion models PCN and PF-Net were used for the classification task of the 3D Pottery 8 dataset to obtain the best model under the current conditions.
5. From data preprocessing to evaluation metric visualization, a fully functional deep learning engineering project was built to achieve the integration and flexible switching of multiple models.
6. On the basis of the existing model training basic process, new features such as data preprocessing scripts, automatic proportional allocation of datasets, and data preloading were added to improve the efficiency of big data training under limited resource conditions.
7. The evaluation metrics were compared, correlated and analyzed in multiple dimensions to infer the current limitations and provide improvement suggestions for subsequent work.

1.3 Structure of Dissertation

- The **Background Research** section will describe the initial extensive research work in the fields of AI technology, cultural heritage protection, computer vision, 3D graphics classification and completion algorithms in which the research topic is located.
- The **Methodology and Implementation** section will provide a detailed explanation of the theoretical design ideas and technical implementation solutions of the project.
- The **Experiments and Results** section will discuss the process of executing model training based on engineering implementation to achieve project results, as well as a specific evaluation of the model performance based on evaluation metrics.

- The **Conclusion** section will provide a detailed analysis and evaluation of the design ideas, engineering implementation, and results evaluation stages of the entire project.

2 Background Research

The background investigation process of this project went through three stages. First, A broad understanding was acquired of the application of AI and machine learning technologies in the field of historical cultural relics restoration, with a particular focus on the latest progress in restoring physical cultural relics such as paintings, pottery, and porcelain. Secondly, in order to find a suitable data set and learn from past practical cases, the scope of the investigation focused on cases of pattern repair or shape reconstruction represented by pottery. Finally, after deciding to try to apply the 3D point cloud processing model to the determined data set, the research work turned to comparing multiple existing 3D point cloud processing network models based on deep learning to select network models suitable for the specific scenario of this project.

The following content is the detailed description of the investigation results of these three stages.

2.1 Application Scenarios of AI Technology

Currently, AI and machine learning technologies are actively researched and widely used in the acquisition and management of cultural relics data and image-based cultural relics restoration.

2.1.1 Acquisition and Management of Heritage Data

A rich and effective cultural relic data model is a necessary condition for machine learning. In reality, although 3D scanning technology is assisted, its performanceand application scope are still limited. To address this problem, Gaber et al. (2023) provided a method to reconstruct cultural relic models through photos and line drawings [7]. The model supports cultural relic identification (through model retrieval) and classification (through model clustering). Especially in the retrieval scenario, this method has a stronger effect than using images for retrieval, and improves the retrieval method of multi-view documents and 3D models. It is worth mentioning that this method can also use neural networks to encode point clouds into neural network representations, support tasks such as classification and segmentation, and directly generate feature vectors representing these point clouds.

In terms of data management, natural language processing (NLP) technology or image recognition technology can be used to automatically extract the main features of cultural relics, thereby automatically generating labels, descriptions and attribute information of cultural relics, helping to build an efficient cultural relic database and reducing the workload of manual annotation. For example, when studying the

application of 3D shape descriptors, Acke et al. (2021) demonstrated a new type of compact shape descriptor and verified its effectiveness through system evaluation. The study created a dataset containing a large number of 3D models, conducted detailed performance comparisons, and implemented a practical 3D content retrieval system [8].

2.1.2 Image-based Classification and Restoration of Artefacts

The current application of deep learning algorithms in cultural relic restoration is mainly concentrated in the field of 2D data, that is, the processing of surface patterns of flat paintings or three-dimensional artworks. At first, Pathak et al. (2016) proposed Context Encoders to perform image inpainting through feature learning. This method performs well in filling large area gaps, but has limited effect in processing details and irregular shapes [9]. Subsequently, Liu et al. (2018) introduced an image inpainting method based on partial convolution, which is particularly suitable for repairing irregular holes. By convolving only valid pixels, the inpainting effect and image quality are significantly improved [10]. Furthermore, Xu et al. (2020) improved the example-based method, focusing on the texture consistency of document images and reducing artifacts in the inpainting process, providing new ideas for the restoration of historical documents [11]. On this basis, Xiao et al. (2016) proposed a historical relic image inpainting method based on salient shapes, which reconstructed the salient shapes of damaged objects through shape point set registration and curve fitting, and used shape guidance maps and gradients to generate new energy functions, which significantly improved the inpainting effect and semantic consistency [12]. These studies have made progress one after another, gradually improving in processing missing image areas, improving restoration details and reducing computational complexity, providing increasingly powerful technical support for cultural relics restoration.

In addition, AI and machine learning technologies are also expected to play a role in anti-counterfeiting technologies for authenticating artworks, analyzing and monitoring environmental changes in the preservation of cultural relics, and monitoring and assessing potential threats and vulnerabilities faced by cultural heritage sites and cultural relics through remote sensing technology and geospatial data [1].

2.2 Restoration Case Experience

Guided by the actual needs of the project, this phase of the investigation primarily focuses on three aspects: data collection methods, the feasibility of AI and machine learning technologies, and performance evaluation methods.

First, in terms of data collection, it is learned from a study that high-precision 3D scanning technology is essential for accurate modeling and restoration of early artifacts [13]. The study provided accurate visualization of Neolithic pottery through detailed 3D data collection and digital modeling. This method demonstrates the importance of data collection in restoration, especially when it comes to accurate

restoration of details, such an accurate data foundation is indispensable. At the same time, another paper introduces a 3D pottery modeling tool called qp, which can semi-automatically generate a series of random 3D pottery that is similar to ancient Greek pottery in morphological features [14]. The article also emphasizes the importance of ground turth databases in the development of content retrieval mechanisms. These studies have very practical reference and reference significance for the selection of data sets for this project.

Second, in terms of the feasibility of artificial intelligence and machine learning technologies, multiple research cases have demonstrated the effectiveness of directly applying deep learning models to 3D graphics processing tasks. The investigation found that the aforementioned study used deep neural networks for digital inpainting and used convolutional neural networks (CNNs) to complete the images of damaged cultural relics [3]. In contrast, another study introduced a more complex hybrid framework that combines 3D encoder-decoder generative adversarial networks (3D-ED-GAN) and long-term recurrent convolutional networks (LRCN) [15]. This method can reconstruct complete and high-resolution 3D objects from damaged models.

Finally, in terms of performanceevaluation, the evaluation content should include at least a comprehensive evaluation of both quantitative and qualitative dimensions. For example, from a qualitative evaluation perspective, a study used the peak signal-to-noise ratio (PSNR) metric, a common standard for measuring image quality, to quantitatively evaluate the completion performance and discussed in depth the impact of the proportion of sub-image overlap areas on PSNR. At the same time, from a quantitative research perspective, the actual performance of the completion scheme was evaluated by visually inspecting the restoration results [3].

2.3 Deep Learning 3D Point Cloud Processing Model

In the field of 3D point cloud processing, deep learning models have significantly promoted the development of classification and completion tasks and achieved a number of important research results. Table 1 and Table 2 are the tables of mainstream common models for 3D point cloud classification and completion.

Model Name	Theory	Advantages	Disadvantages	Year
PointNet [16]	Point-wise MLPs	Simplicity, permutation invariance	Limited local feature learning	2017
PointNet++[17]	Hierarchical PointNet	Captures local and global features	Increased complexity	2017
PointCNN [18]	Convolutional Neural Network	Captures local features effectively, permutation invariance	Complex architecture, higher computational requirements	2018
DGCNN [19]	Dynamic Graph CNN	Adaptive local neighborhood, captures complex relationships	High computational cost, complex implementation	2019
PointConv [20]	Convolutional Neural Network	Efficient feature extraction,	High memory usage, complex computations	2019

Table 1. Deep learning models used for 3D point cloud classification tasks.

Model Name	Theory	Advantages	Disadvantages	Year
PCN [21]	Autoencoder-based network	Effective in completing large missing regions	Struggles with fine details, limited resolution	2018
TopNet [22]	Tree-structured decoder	Hierarchical feature extraction	Complex architecture, high training time	2019
PF-Net [23]	Fractal-based hierarchical network	Handles various completion tasks well	High computational complexity	2020
GRNet [24]	Graph-based Neural Network	Effective in capturing spatial relationships	High memory usage	2020
PMP-Net [25]	Self-supervised learning with point-wise mask prediction	No need for ground truth data, robust to varying input sizes	May struggle with very complex shapes, lower accuracy	2021

Table 2. Deep Learning Models for 3D Point Cloud Completion Tasks

2.3.1 3D Point Clouds

The representation of 3D graphics, namely 3D shape descriptors, includes voxels, point clouds or implicit surfaces [16]. 3D point cloud is a collection used to represent objects or scenes in 3D space, and each point contains its coordinates (X, Y, Z) in 3D space. These points are usually captured by devices such as 3D scanners, stereo vision systems, laser scanners, etc. Point cloud representation has significant advantages in complex 3D modeling and advanced algorithm applications due to its high accuracy, simplicity, flexible storage and powerful visual perception capabilities. For the above considerations, this project chose point cloud representation as the training data format for pottery 3D models.

2.3.2 3D Point Cloud Datasets

Regarding existing point cloud datasets, the research mainly involves three datasets: ModelNet, ShapeNet, and 3D pottery. Among them, ModelNet and ShapeNet are widely used datasets for 3D object classification and completion. ModelNet contains 3D CAD models of 40 common objects, such as airplanes, chairs, and tables, with high data quality and diverse categories [26]; while ShapeNet provides larger-scale 3D shape data, covers more object categories, supports classification, completion, and segmentation tasks, and has richer categories and instances [27].

3D Pottery (A 3D Pottery Content Based Retrieval Benchmark Dataset) is a 3D point cloud model that focuses on pottery classification. The dataset consists of a total of 1012 digitized, manually modeled, and semi-automatically generated 3D pottery models (generated using a 3D container random generator) [8]. This dataset covers pottery of various shapes and styles, providing rich data support for the research and development of content-based retrieval and repair algorithms.

2.3.3 Classification Models PointNet/PointNet++

PointNet can effectively process and analyze the shape information of 3D point cloud data through the design of symmetric functions and global feature extraction. Its core advantage is that it can ignore the order and arrangement of points and accurately extract global features in the point cloud [17]. This method is very suitable for tasks such as point cloud object detection and classification. PointNet++ introduces hierarchical local feature extraction based on PointNet, and further improves the processing ability of complex shapes by capturing local structures and details in point clouds [18]. In addition, in the research of point cloud completion, many models rely on PointNet/PointNet++ models to complete feature extraction functions, which shows its core position and importance.

2.3.4 Complementation Models PCN and PF-Net

PCN(Point Cloud Network) is a deep learning model for processing 3D point cloud data, aiming at point cloud completion and reconstruction. The core of PCN lies in its

innovative point cloud encoder-decoder structure, which includes a local feature extractor and a global feature fuser. The network converts incomplete point cloud data into high-dimensional feature representation through the encoder, and then generates a complete point cloud through the decoder. The advantage of PCN is that it can effectively handle sparse and missing areas in the point cloud, infer missing points through the deep learning model, and thus achieve high-precision point cloud completion.

PF-Net (Point Fusion Network) is another deep learning model designed for point cloud completion, which uses point cloud fusion technology to improve the completion effect. The innovation of PF-Net lies in its point cloud fusion module, which enhances the ability to recover the integrity of the point cloud by fusing point cloud data from multiple perspectives into a unified feature representation. In addition, PF-Net adopts point cloud alignment and adaptive weight adjustment techniques to deal with inconsistencies and noise in point clouds.

3 Methodology and Implementation

This chapter will elaborate on the core achievements of the project from the perspectives of methodology and implementation. The challenging problems and corresponding solutions encountered during the project progress will be described separately in the subsection marked with "Challenge and Innovation" prefix.

3.1 The Process of Preparing the Dataset

3.1.1 Desired Dataset

To ensure the model effect, the data set is crucial for model training. The data set of this project should be composed of a series of real pottery point cloud sample data obtained by 3D scanning and other means. After comprehensive consideration, the specific requirements should meet the following points:

- **Accurate categories:** The data set should include representative pottery categories excavated from different periods and locations in ancient times. Due to the limitations of the project goals (classification and completion tasks) and other conditions, the category distinction criteria should give priority to shape differences, and there are no rigid requirements for patterns, colors and patterns.
- **Sufficient quantity:** Each category should contain dozens to hundreds of real sample data. In order to improve the generalization ability of the model and avoid overfitting, subtle differences in different dimensions between each sample should be maintained as much as possible to enrich the feature space.
- **Data format available:** The native data source can be a common 3D graphics representation that can be converted into point cloud data, such as mesh models, voxel data, depth maps, and implicit surfaces.
- **Information completeness:** In order to ensure that the model can learn the mapping relationship from damaged to complete and thus achieve accurate shape

completion, each sample should be divided into sample data of the complete shape and the corresponding damaged sample data.

3.1.2 Source Selection for Complete Pottery Data

In order to find a suitable source of pottery dataset, the initial step involved directly searching on public dataset platforms in the field of machine learning, such as Kaggle and VisualData. However, due to the difficulty of data collection and the legal and ethical issues of protecting cultural heritage, the available data is very scarce. Therefore, the source of the dataset turned to past research papers, especially research cases related to pottery and containers. The survey found that a study related to 3D Pottery content-based retrieval compiled a 3D pottery dataset during the experiment [8] and publicly stated that it can be used for research. After a detailed evaluation, the dataset basically meets the research requirements of the expected dataset and was used as the original source of the dataset for this project.

After checking all the samples of the 3D Pottery Dataset, due to the limitations of the project practice environment, the dataset actually used in this project consists of 8 representative categories with a large number of samples selected from the original 36 categories, with a total of 654 samples (although glass does not belong to pottery, it is also included in the dataset for the consideration of training performance due to its abundant sample number). Table 3 shows the category names, quantities and corresponding standard graphics of these 8 categories.

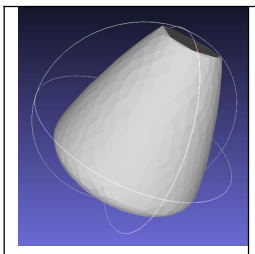
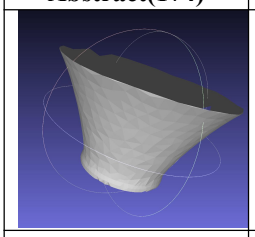
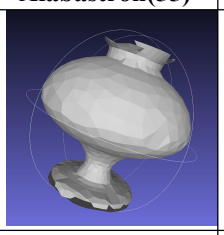
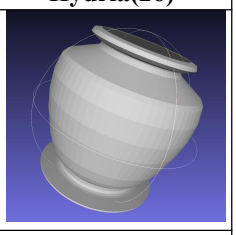
			
Abstract(174)	**Alabastron(55)**	**Amphora(96)**	**Hydria(26)**
			
Kalathos(54)	**Psykter(48)**	**Glass(48)**	**Vase(153)**

Table 3. Category names and corresponding standard graphs for the 8 classifications in the pottery dataset.

3.1.3 Challenge and Innovation: production of new broken pottery data

One of the goals of this project is to complete broken pottery. In order to ensure that the model can learn the mapping relationship from broken to complete, so as to achieve accurate shape completion, in addition to the complete data of pottery samples, the corresponding broken samples are also essential for training. However, under the condition of determining the complete data set, it is already very difficult to obtain broken samples through public resources and other channels, and almost only consider manually making the corresponding broken versions through data processing scripts or 3D model processing tools. In order to more accurately simulate the shape of broken pottery in actual scenes and obtain a more practical model effect, this project chose to use the 3D model processing tool Meshlab[1] to manually make broken data one by one.

In order to fit the actual situation and ensure the training performance, the following principles were considered when manually making new broken samples:

- Only one breakage is made for each pottery sample.
- The breakage mostly occurs in relatively "protruding" positions such as the edge or handle of the pottery.
- In the same category, most pottery samples showed small-area damage, and a few showed medium- and large-area damage.
- In the same category, most pottery samples showed transverse wear-type damage, and a few showed longitudinal crack-type damage.

Table 4 shows the graphs of 4 complete samples and the corresponding broken samples.

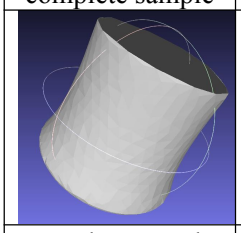
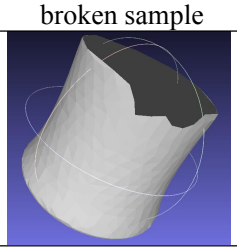
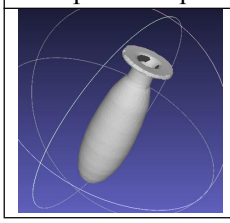
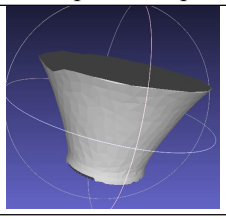
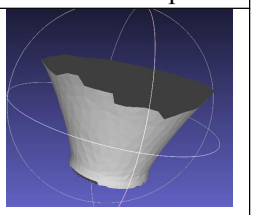
complete sample	broken sample	complete sample	broken sample
			
complete sample	broken sample	complete sample	broken sample
			

Table 4. New handmade broken samples versus original intact samples.

[1] MeshLab is an open source 3D mesh processing software that provides a variety of functions for editing, repairing, checking and converting 3D models.

After adding the broken samples, the new dataset was named 3D Pottery 8 dataset. Its final structure is a dataset containing 8 categories and a total of 654 sets of 3D pottery graphic samples. Each set of samples includes two types of data: complete sample and broken sample. This dataset resource has been made available for download in the project github repository as one of the results, and can be used as a reference for subsequent research involving 3D pottery completion. The detailed file structure of this dataset is shown in Figure 2.

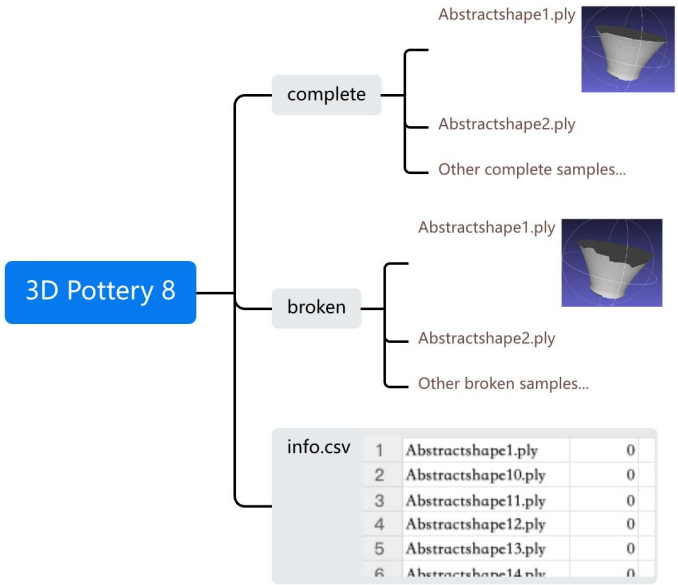


Figure 2. File structure of new manually produced dataset: 3D Pottery 8.

3.2 Selection of Network Models

Under the condition that the dataset is available[2], the decision-making process of the network model is to evaluate the existing 3D point cloud processing model from the following aspects to determine whether it is suitable for this project:

- **Model Performance:** The selected network model needs to perform well on public large-scale datasets (such as ModelNet, ShapeNet, etc.) to ensure its basic model effect.

[2] referring to the dataset with sufficient sample number, label information, and format (after format conversion, etc.) that can be directly used for model training.

- **Feature adaptation:** Different network model structures are suitable for different types of datasets. In order to obtain the best performance, the selected model should try to fit the characteristics of the pottery dataset, which requires in-depth analysis of the model structure and dataset characteristics.
- **Feasibility:** Models that have been verified by multiple experiments and have past cases for reference should be selected to ensure engineering within a limited time.
- **Efficiency:** Resource utilization needs to be efficient in order to achieve the ideal model performance within a limited time. This involves the reasonable allocation of computing resources and the optimization of model training strategies.
- **Consistent evaluation method:** When using multiple network models in the same task, ensure that the same evaluation metrics are used for effective comparison and analysis.

3.2.1 Model for the classification task

After the above evaluations, PointNet/PointNet++ was selected as the network model for multi-classification tasks. First, the model has demonstrated excellent results on large public datasets such as ModelNet40 and KITTI-360. These models can effectively process 3D point cloud data and perform classification tasks, showing high accuracy and robustness. Secondly, the network model adopts a method of directly processing 3D point cloud data instead of relying on traditional grid or voxel representation, which is also the main reason why PointNet/PointNet++ has a breakthrough in 3D graphics feature extraction (see Figure 3 for detailed network architecture). Because the shape of pottery is complex and rich in details, this method of directly processing point clouds is particularly suitable. In addition, PointNet/PointNet++ has relatively high computational efficiency and can complete training and reasoning tasks within reasonable computing resources and time. There are already a large number of research and application cases, and its open source implementation and detailed documentation also help to implement engineering within a limited time.

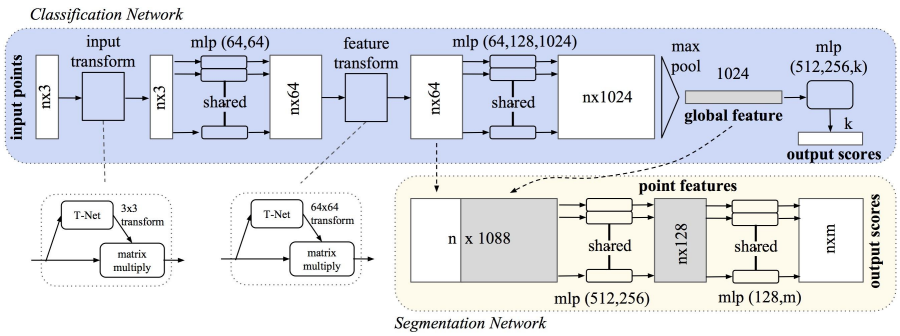


Figure 3. PointNet architecture.

3.2.2 Models for the Completion Task

Compared with the more common classification requirements, the pottery completion task is more innovative and challenging, and has more research value. Therefore, two network models, PCN and PF-Net, were used in the completion model for comparative experiments.

As mentioned above, Point Completion Network (PCN) is a network model designed specifically for point cloud completion tasks. The core idea is to generate more points to complete the missing parts based on the input incomplete point cloud. The key to PCN's suitability for pottery model completion tasks is that its encoder-decoder architecture can effectively generate complete shapes (see Figure 4 for detailed network architecture). By encoding and decoding the input incomplete point cloud into a complete point cloud, PCN can complete the missing parts of the pottery and restore its overall shape. This feature makes PCN particularly suitable for processing objects such as pottery that have obvious overall structures but are partially damaged. In addition, PCN performs well on large public datasets and can provide stable and high-quality completion results; it has rich implementation cases, and its feasibility and efficiency are also guaranteed. However, PCN has limited ability to complete very complex and detailed point clouds.

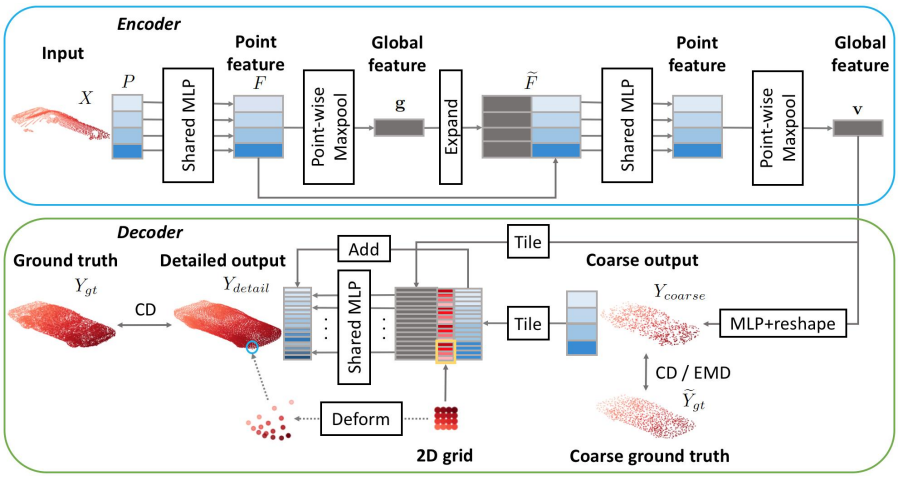


Figure 4. PCN Architecture.

Point Fractal Network (PF-Net) is another more complex network model designed to improve the point cloud completion performance. The reason for choosing PF-Net is that its recursive refinement module can capture and generate complex detail structures. Pottery usually has rich shape details. PF-Net adopts the idea of fractal network. By recursively refining the point cloud, gradually increasing the density and details of the points, and combining global and local features, it can not only restore the overall shape, but also finely reproduce the details of the pottery, making the completed model more accurate (see Figure 5 for the detailed network architecture).

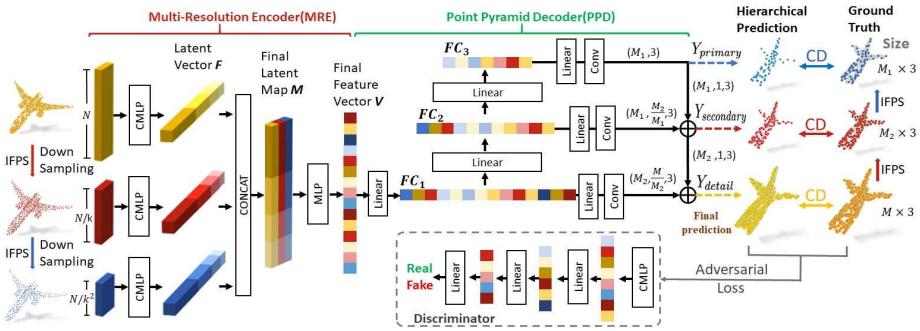


Figure 5. The architecture of PF-Net.

Another important reason to consider PF-Net is that it uses the training structure of Generative Adversarial Network (GAN) [30], which enables PF-Net to take advantage of the GAN training model. Specifically, GAN is able to generate higher quality samples through the adversarial training process of the generator and the discriminator. The generator tries to generate real samples, while the discriminator tries to distinguish between the generated samples and the real samples. This adversarial process prompts the generator to continuously improve its output, making the generated samples closer to the real data both visually and structurally. Combined with the actual needs of this project, the principle of the GAN training model based on the PF-Net model is shown in Figure 6.

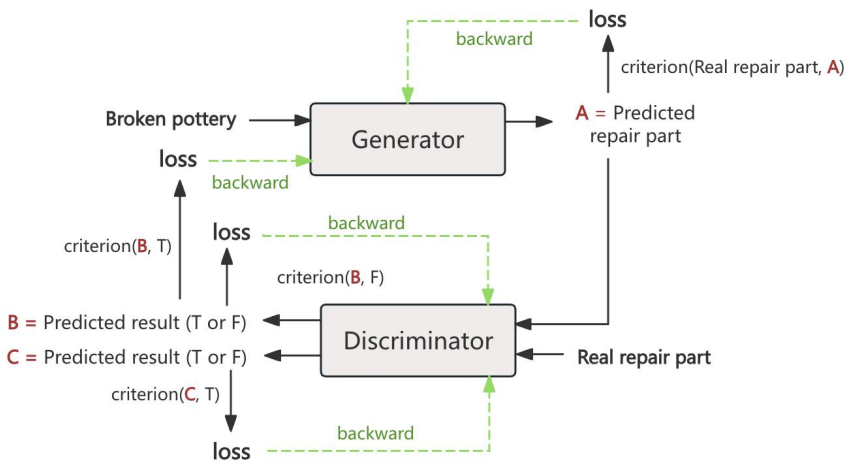


Figure 6. Schematic diagram of the principle of GAN training mode based on the PF-Net model. The figure includes the input and output data of the discriminator and generator in a training cycle, which are compared with the expected standard, and the loss values are calculated respectively, and then fed back to the model through backward.

As shown in the figure above, in the task of completing broken pottery, the GAN generator can be used to generate point cloud data for completing broken pottery, which is used as a completion sample to repair the missing parts of the pottery. The generator learns how to generate a detailed point cloud that conforms to the shape of the real pottery through adversarial training with the discriminator, making the completed pottery model more accurate and complete. This adversarial training not only improves the quality of the completion results, but also enhances the robustness of the model in dealing with complex details and diverse samples.

However, it cannot be ignored that compared with PCN, PF-Net has relatively few application cases, its model structure is more complex, and the training time may be longer, so there is a sacrifice in efficiency.

3.3 Evaluation Metrics and Assessment Methods

3.3.1 Evaluation of The Classification Task

In this project, the pottery dataset includes 8 categories (labels), so the task belongs to a multi-classification problem. In model training, in order to simplify model input and improve computational efficiency, the classification labels are actually identified using integer values 0 to 7. The evaluation performance is mainly based on the gap between the predicted label and the true label. Specifically, this task focuses on the following key metrics:

(1) **Accuracy**: the ratio of the number of samples correctly predicted by the model to the total number of samples.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}$$

(2) **Precision**: a measure of the accuracy of the model's predictions for each category.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

(3) **Recall**: a measure of the model's predictive coverage of each category.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

(4) **F1 Score:** combines precision and recall and is their reconciled average.

$$\text{F1 Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

In the above formula:

- TP (True Positive): True positive number, that is, the number of samples correctly predicted as positive by the model.

- TN (True Negative): True negative number, that is, the number of samples correctly predicted as negative by the model.
- FP (False Positive): False positive number, that is, the number of samples incorrectly predicted as positive by the model.
- FN (False Negative): False negative number, that is, the number of samples incorrectly predicted as positive by the model.

In addition, the Confusion Matrix, Precision-Recall curve, and AUC (Area Under the Curve) / ROC (Receiver Operating Characteristic) curve will be used to comprehensively evaluate the performance of the model. The Confusion Matrix shows the prediction results of the model on each category, including the true positives (TP), false positives (FP), true negatives (TN), and false negatives (FN) mentioned above, which helps to identify which categories have good classification performance and which categories have poor performance. The Precision-Recall curve evaluates the effectiveness of the model in processing unbalanced data by showing the changes in precision and recall under different thresholds. The AUC/ROC curve evaluates the overall performance of the model under various thresholds. The AUC value represents the area under the ROC curve, which shows the model's ability to distinguish different categories. Using these metrics can provide a more comprehensive understanding of the classification performanceand performance of the model.

3.3.2 Evaluation of The Complementation Task

In the point cloud completion task of this project, the key metrics for evaluating model performance include Chamfer Distance and the intuitive comparison between the predicted 3D graphics and the ground truth value.

Chamfer distance is a commonly used indicator to measure the similarity between point clouds [31]. It measures the degree of matching between two point clouds by calculating the average distance between them. Specifically, Chamfer Distance is calculated by the following formula:

$$D_{Chamfer}(P,Q) = \frac{1}{|P|}\sum_{p\in P} min_{q\in Q} ||p-q||^2 + \frac{1}{|Q|}\sum_{q\in Q} min_{p\in P} ||p-q||^2$$

where P and Q are the predicted and real point clouds, respectively, and denote the Euclidean distance between point p and point q.

Chamfer Distance is widely used because it can effectively quantify the spatial consistency and structural similarity of point clouds. It not only considers the point-to-point distance between the predicted point cloud and the real point cloud, but also calculates the minimum distance of each point in the other point cloud, thus providing a comprehensive matching evaluation of the point cloud. This makes Chamfer Distance particularly suitable for completion tasks because it can quantify the difference between the point cloud generated by the completion model and the actual target.

In addition, the intuitive comparison of the predicted 3D graphics and the Ground Truth is also very important. This comparison helps to intuitively evaluate the completion performance of the model by visualizing the difference between the predicted results and the actual target. By comparing, the performance of the model in detail recovery, overall shape, and texture reconstruction can be found, so as to further optimize the performance of the model. In general, these evaluation methods can comprehensively measure the accuracy and performance of the completion model, ensuring that the generated 3D pottery is as close to the real sample as possible in shape and details.

3.4 Overall Theoretical Plan of The Project

In summary, after determining the specific data set, the network model for training, and the metrics for evaluation, the theoretical solution for the project has been basically clarified, forming the basis for subsequent engineering implementation and experiments. For a more intuitive display, the theoretical solution of this project is summarized in the following table:

theory	goal	model stucture	metric
PointNet (Classification)	Input: complete or broken samples Output: correct classification	Shared MLP and global feature pooling operations to extract global features	Accuracy, Precision, Recall, F1 Score, Confusion Matrix, Precision-Recall Curve and AUC / ROC
PCN (Completion)	Input broken sample, generate complete sample	Encoder-decoder structure	ChamferDistance and Visual comparison of predicted point cloud with Ground Truth
PF-Net (Completion)	Input broken sample, generate partially repaired sample	GAN structure: Discriminator and Generator	

Table 5. Summary of the theoretical scheme of the 3D pottery classification and complementation project.

3.5 Implementation

This project achieves the experimental purpose by building a modular project based on Python, that is, using the homemade pottery dataset (3D Pottery 8) as training data, applying three different network models to train the corresponding classification and completion models, and testing and evaluating their performance. In actual

development, on the basis of following the general process of machine learning model training (loading data-training-evaluation), the open source github projects provided by the research papers of the three models were mainly referred to. For example, the implementation of functions such as parameter setting based on the GPU environment, 3D point cloud data preprocessing, network structure definition, and prediction performance visualization. In addition, several innovative developments and improvements were made in the development process to address defects and deficiencies in the process, such as automatic dataset segmentation and tracking and recording mechanism.

This chapter will first introduce the basic situation of the development environment, and then divide the overall implementation process into two stages: training preparation and training execution according to the development logic structure. The technical architecture diagram containing the main functions and dependencies of these two stages is shown in Figure 7.

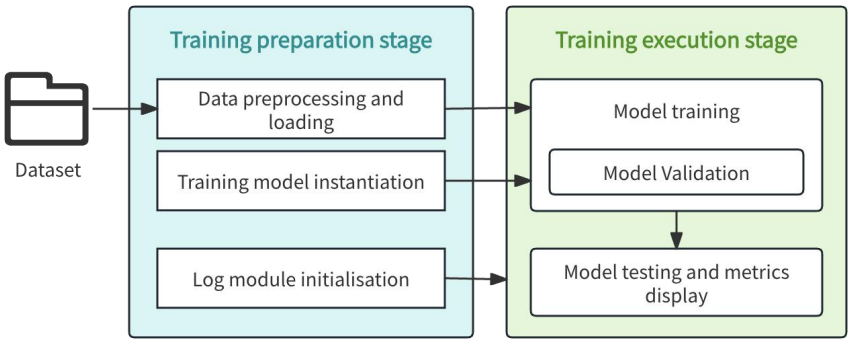


Figure 7. Architecture of the overall functional flow, the training preparation phase supports the training execution phase. The log module will be called multiple times during the training execution phase after instantiation.

3.5.1 Basic information

3.5.1.1 Development Environment

Development language: Python
Main framework: PyTorch
Development platform: Google Colab
Data storage: Google Drive
Resource configuration:
Hardware configuration:
GPU model: GPU L4 (based on NVIDIA T4 Tensor Core GPU)
System RAM: 53.0 GB
GPU RAM: 22.5 GB

Disk: 201.2 GB

Software configuration:

Python version: Python 3

Key application packages: torch, numpy, pandas, trimesh, pyvista, etc.

3.5.1.2 Project Structure

The following file structure diagram shows the core modules in the project structure and a brief description of their functions. These modules (file names) will be mentioned in the subsequent detailed module architecture diagram.

```
Pottery Project
├── Data   Placement of the dataset
│   └── PotteryDataset
│       ├── complete
│       ├── broken
│       └── info.csv
├── Dataset
│   └── pottery_dataset.py  Definition of the dataset
├── Prepare
│   ├── prepare.py  Definition of logging mechanisms
│   └── data_pre_PF_Net.py  Data preprocessing script
├── Models  Definition of the models
│   ├── pointnet_cls.py
│   ├── PCN.py
│   └── model_PFNet.py
├── train_PointNet.py ┐
├── train_PCN.py      ├─ Training Scripts
└── train_PF_Net.py   ┘
```

Figure 8. Structure of the project files.

3.5.2 Training Preparation Phase

This stage is the preparation work before training, which mainly includes the preprocessing of the dataset, the instantiation of the dataset and model, and the initialization of the tracking and recording module.

3.5.2.1 Dataset Preprocessing

- **Format conversion and sampling**

The dataset after adding the broken samples was originally in .obj format, which is mainly used to represent polygonal meshes, including vertices, texture coordinates, normal vectors, and faces. To adapt to the input format of the network model, the format of all samples has been converted to a more streamlined point cloud format,

and the vertices on the polygonal mesh surface are uniformly sampled (2048 sampling points).

- **Normalization processing**

Scale each coordinate value of all samples to the range of [0, 1]. Specifically, first calculate the minimum and maximum values of each point value, subtract the minimum value of the dimension for each point in the sample, and then divide it by the range of the dimension (maximum value minus minimum value) to obtain the normalized sample data. Through this processing, the dimensional differences between different coordinate dimensions can be eliminated, thereby improving the stability and performance of subsequent model training and prediction.

- **Challenge and innovation: Set up a separate dataset processing script for PF-Net**

The PF-Net model used for the completion task generates broken parts for broken pottery. To achieve the purpose of learning, the training data should consist of broken pottery and its corresponding correct broken parts. This is different from the datasets required by the PointNet and PCN models (see Table 6 for a detailed comparison). Referring to the logic of the original PF-Net github project, when traversing the model training, a lot of resources are needed to do complicated processing on the complete samples in the original dataset to obtain the damaged samples and damaged parts for subsequent training. This project improves this part of the code, abstracts this process separately, and saves the processing results to a file on the hard disk. A more general data preprocessing script is formed, which makes the overall logic more reasonable, simplifies the training process, and greatly improves the training efficiency.

<table>
<tr><th rowspan="2">Task</th><th rowspan="2">Model</th><th rowspan="2">Dataset source and composition</th><th colspan="2">Mapping relationships to be learnt</th></tr>
<tr><th>Input sample</th><th>Predicted result</th></tr>
<tr><td>Multi-classification</td><td>PointNet
PointNet++</td><td rowspan="2">NEW 3D Pottery 8 Dataset:
Complete samples from the original dataset
+
Manually produced broken samples
+
Correct classification labels</td><td>broken samples
complete samples</td><td>correct classification labels</td></tr>
<tr><td rowspan="2">Completion</td><td>PCN</td><td>broken samples</td><td>complete samples</td></tr>
<tr><td>PF-Net</td><td>Obtained by scripting the original dataset:
broken samples
+
corresponding broken parts</td><td>broken samples</td><td>corresponding broken parts</td></tr>
</table>

Table 6. Comparison of the requirements of the different models on the dataset, where the part in red font is the step that has been improved.

Compared with manually making broken pottery samples, the function implemented by the script processing mentioned above is to realize the cropping (broken) of 3D point cloud graphics by calling a series of 3D point cloud data processing methods. Taking the pottery category as an example, the processing first

selects a view direction center from the complete pottery data sample, and then crops a fixed number of point sets closest to it with this center, and finally obtains the remaining data as the cropped (broken) pottery sample, and the cropped point set is used as the standard damaged part, which is also the part that the model needs to predict (the process diagram is shown in Figure 9, and the code implementation can be viewed in the Github project). Obviously, this method is more efficient than manually making a dataset, but it also has the problem of unrealistic damage and reduced reference value.

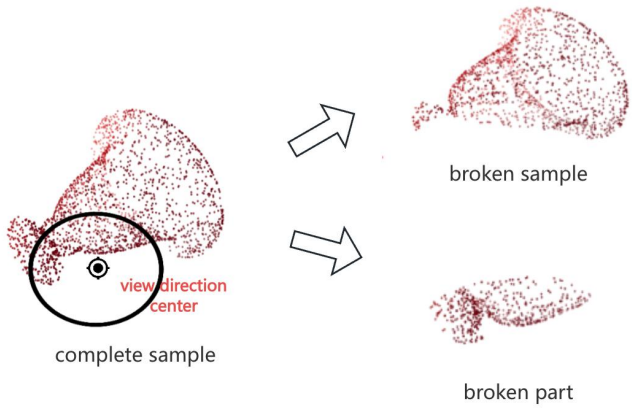


Figure 9. Schematic illustration of the cropping process for 3D point cloud graphics.

- **Challenge and innovation: automatic data set segmentation**

When training a model, it is usually necessary to divide the data set into three parts: training set, validation set, and test set. For example, in this project, the division ratios are 80%, 10%, and 10%, respectively. The data set used in the project contains 8 pottery categories, each of which contains dozens to hundreds of graphic samples. In order to achieve the ideal model performance, it is necessary to randomly select a certain number of samples in each category and place them in the training set, validation set, and test set respectively. However, manual operation is cumbersome, time-consuming, and laborious. This project provides an independent method for automatically dividing the data set (dataset_divider, the code implementation of this method can be viewed in the Github project). Only the three parameters—the original dataset path, the proportion of the training set (e.g., 0.8, with the validation set and test set automatically divided from the remainder), and the path for the divided dataset—need to be entered to automatically achieve the division. This can improve development efficiency on the one hand, and can also perform random division multiple times to obtain the best experimental performance on the other hand.

3.5.2.2 Definition and Loading of Dataset Class and Model Class

In this project, the custom dataset class PotteryDataset follows the general form of the PyTorch framework, that is, it inherits torch.utils.data. Dataset and implements the following three methods: __init__, __len__, and __getitem__. This definition method makes the data loading process flexible and convenient. In the training preparation stage, the dataset class PotteryDataset is instantiated and directly referenced to the torch.utils.data.DataLoader object to load it.

The model class mainly refers to the corresponding model classes in the open source github projects of the three network models (PointNet [32], PCN [33], and PF-Net [34]). In order to make the model compatible with this project, the code of the model definition was sorted and improved in combination with the network structure analysis content in its paper during development. In the training preparation stage, the introduced model class is directly instantiated for subsequent training. In addition, if there is a previously saved "Checkpoint" file under the pre-trained model path (in this project, the checkpoint of the optimal model in the test or evaluation stage will be automatically saved), the program can choose to load the model parameters from the Checkpoint file into the current model and restore it to the corresponding training state, thereby accelerating convergence and improving performance.

- **Challenge and innovation: Pre-loading data into memory for PF-Net**

In actual engineering development, in order to reduce the amount of calculation and focus on important features, PF-Net performs multiple key point sampling on damaged samples and damaged parts, which greatly increases the amount of data used to input the network model. Traversing samples and reading data during training will consume a lot of resources and time, slowing down training efficiency. This project improves the original logic to address this problem. The operation of reading all data is pre-placed, that is, the data set is read into memory before training begins, completely decoupling it from the training process. This greatly reduces the impact on the efficiency of the training execution process. In particular, when the training is interrupted unexpectedly or the training parameters are modified, there is no need to worry about wasting time by re-reading the data (the code implementation can be viewed in the Github project).

3.5.2.3 Challenge and Innovation: Tracking and Recording Mechanism

For machine learning projects, multiple model training and testing processes are often required to achieve the experimental purpose. Therefore, traceable log records are very important. In the development process of this project, a relatively complete tracking and recording mechanism was implemented based on the relevant experience of past cases. Calling the relevant method (prepare_logger, code implementation can be viewed in the github project) before model training can automatically generate a series of files or storage paths for recording training status, as shown in Figure 10. At the same time, a log method is also provided, which can be called at any time during the entire training execution phase to achieve writing log files while printing logs,

which not only guarantees the need for real-time viewing, but also achieves traceable log backup. In general, this function makes the development process of the entire project more orderly and controllable, ensuring that every step in the training process can be traced back and reviewed at any stage. This not only improves the transparency and repeatability of the experiment, but also provides a reliable foundation for subsequent model optimization and debugging.

```
Pottery Project
├── record_log
│   └── pottery_demo_PF_Net_20240718013323
│       ├── dataset        current training set, validation set, test set
│       ├── checkpoint     the best model dictionary
│       ├── val_result     validation result images
│       ├── test_result    test result images
│       ├── val_tb         validation data for TensorBoard display
│       ├── test_tb        test data for TensorBoard display
│       └── logger.log     records of all logs
│
```

Figure 10. Automatically generated folders in the trace logging mechanism and their usages.

3.5.3 Training Execution Phase

3.5.3.1 Hyperparameter Setting

- **Loss Calculation**

The loss function is used to measure the difference between the model's prediction results and the actual target, so as to guide the model on how to adjust its parameters during training to minimize the prediction error. In this project, the Loss functions used by the three network models are different due to the differences in their training tasks.

Specifically, in the multi-classification task using the PointNet model, the total loss is obtained by adding the negative log-likelihood loss (classification loss) and the feature transformation regularization loss in proportion. Among them, the classification loss ensures the classification accuracy of the model, and the feature transformation regularization loss constrains the feature transformation matrix, thereby improving the generalization ability of the model.

On the other hand, for the completion task, the loss of the PCN model mainly relies on calculating the commonly used point cloud matching loss function - Chamfer DistanceLoss[3]. It measures the similarity between two point clouds by calculating the mean distance between each point and the closest point. The loss calculation of the PCN model is more complicated because of its GAN training structure, the generator and the discriminator have their own different loss calculation methods. The goal of the generator is to generate realistic point cloud data so that the discriminator cannot

[3] About ChamferDistance, there is a detailed introduction in "3.3.2 Evaluation of The Completion Task".

distinguish the generated point cloud from the real point cloud. To achieve this goal, the generator uses Binary Cross-Entropy Loss (BCE Loss). The goal of the discriminator is to distinguish between the real point cloud and the generated point cloud. Therefore, the discriminator uses a combination of losses, namely Binary Cross-Entropy Loss and ChamferDistanceLoss.

The following table summarized the loss function calculation method used in this project.

Task	Model	Loss (λ represents the scaling factor)
Multi-classification	PointNet PointNet++	Total Loss=NLL Loss+λ·Regularization Loss
Completion	PCN	Loss=ChamferDistance
	PF-Net	Discriminator_loss = BCE Loss Generator_loss=BCE Loss+λ·Chamfer Distance Loss

Table 7. Comparison of loss function calculations for three network models.

- **Optimizer and Scheduler**

Optimizer and learning rate scheduler are key components for training neural network models. They are used to adjust model parameters to minimize the loss function and thus improve the performance of the model. In this project, the Optimizer uses the Adam optimizer, which is an adaptive learning rate optimization algorithm commonly used in deep learning. The Adam optimizer combines the advantages of momentum optimization and adaptive learning rate adjustment, and usually performs well in training practice. The following is a specific example used in the project.

```
optimizer = optim.Adam(model.parameters(),
lr=params.lr, betas=(0.9, 0.999))
lr_schedual = optim.lr_scheduler.StepLR(optimizer,
step_size=50, gamma=0.7)
```

3.5.3.2 Model Training and Testing

This step is the core content of the training execution phase. Although the purpose of training for different tasks is different, the training process follows the same basic execution flow. In this project, the training process is divided into the preparation phase, training phase, verification phase, and test phase. First, initialize the necessary components such as the model, data loader, and optimizer. In each training cycle, the model first enters the training mode, performs forward propagation, loss calculation, backpropagation, and parameter update through data iteration, and adjusts the learning rate during this period. Subsequently, the model switches to the verification

mode, evaluates the verification set data, calculates the verification loss, and saves the verification results. After the entire training is completed, the model enters the test mode, calculates the final test loss through the test data, and saves the test results. This process ensures that the training, verification, and testing of the model are carried out in an orderly manner, and provides real-time logging for easy monitoring and traceability.The following content will take the PCN network model training as an example to explain the operation steps in detail in the form of the flowchart.

(1) Training and Validation Loop

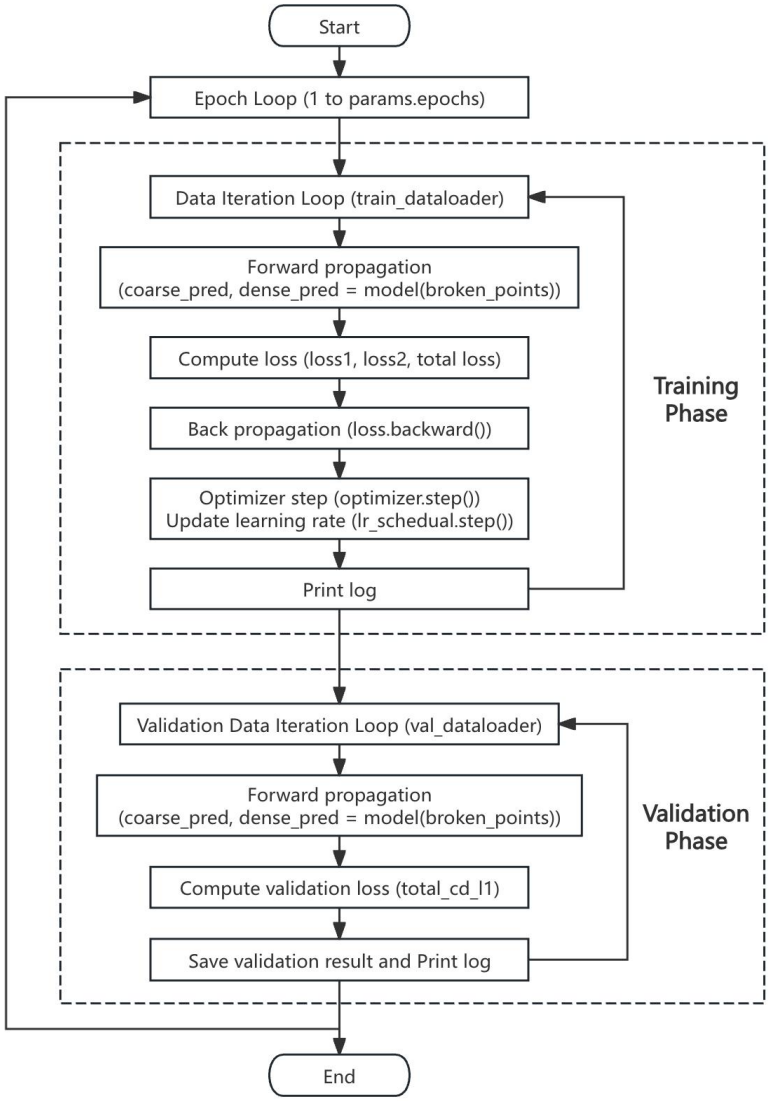


Figure 11. Flowchart of the training and validation loop of the PCN network model.

(2) Testing Loop

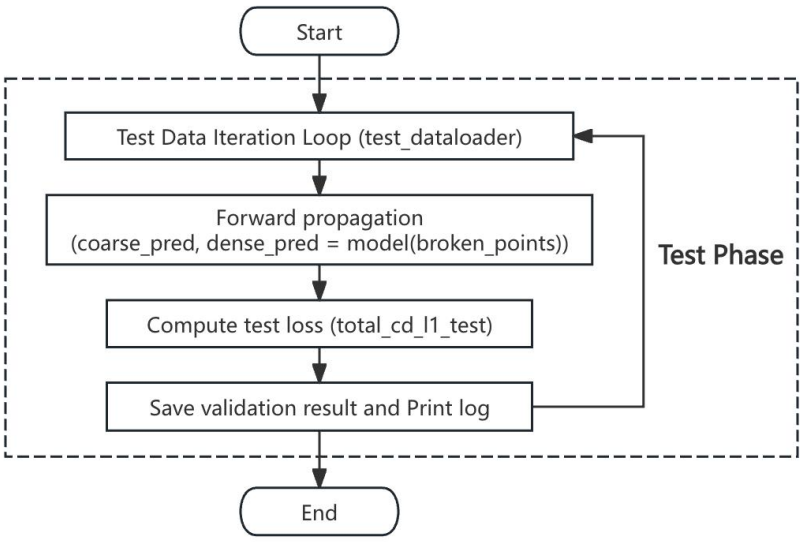


Figure 12. Flowchart of the test loop of the PCN network model.

3.5.3.3 Performance Display

At the end of the training execution phase, the current performance metrics are intuitively displayed in a variety of ways to ensure a comprehensive evaluation of the model performance. In the code development, the following key visualization and evaluation methods are mainly implemented:

For multi-classification tasks, Confusion Matrix, Precision-Recall Curve and AUC/ROC curve (Area Under the Curve / Receiver Operating Characteristic Curve) are mainly used. These metrics can fully display the classification performance of the model on different categories, help identify the pattern of classification errors and optimize model performance.

For completion tasks, the point cloud gap measurement metric ChamferDistance is mainly relied on to horizontally display the model performance. In addition, the result pictures saved during training are also an important way to evaluate model performance. By intuitively comparing the completion results generated by the model with the real data, the effectiveness and accuracy of the model can be further verified. As an auxiliary, TensorBoard[4] will also be used to view the downward trend of loss during training, so as to intuitively display the progress of model training.

[4] TensorBoard is a tool for visualizing and analyzing the training process of machine learning models. It can intuitively display the changing trends of indicators such as loss and accuracy, which helps to debug and optimize the model.

3.5.4 Overall Functional Module Architecture Diagram of The Project

Based on the above content, according to the bottom-up logic, the overall functional module architecture diagram of the project is shown in Figure 13.

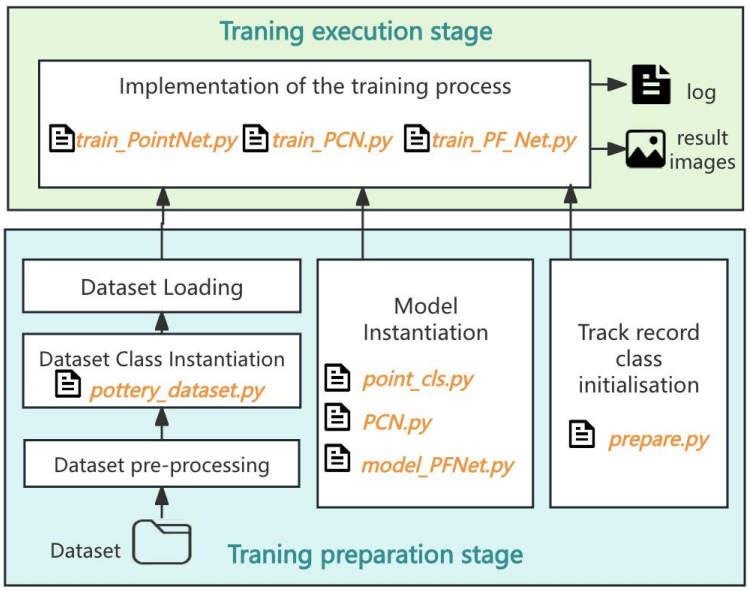


Figure 13. 3D Pottery Multi Classification and Completion project overall functional module architecture (bottom-up logic).

4 Experiments and Results

This chapter will focus on the entire process of executing the experiment after the project is realized. In order to obtain accurate and effective experimental results as efficiently as possible, the experimental process strictly follows the controlled variable method and is carried out according to a plan with a clear purpose guide. This chapter will first introduce the experimental plan, and then further introduce the specific situation of each experiment and the comprehensive analysis and evaluation of the results.

4.1 Experimental Plan

The experimental premise is to be based on the same dataset (3D Pottery 8) and the same hardware and software environment. When making the plan, the main consideration was to adjust the hyperparameter variables such as optimizer type, learning rate, and regularization parameter. Table 8 lists the multiple experiments

planned to be performed in this project. In actual operation, each experiment was conducted multiple times under preset conditions.

Task	Exp No.	Model	Optimizer Type(Hyperparameters)	lr_schedual(Hyperparameters)
Multi-classification	1	PointNet	Adam(learning_rate=0.001, betas=(0.9, 0.999), epsilon=1e-08, weight_decay=1e-4)	step_size=20, gamma=0.7
	2	PointNet	Adam(learning_rate=0.0001, betas=(0.9, 0.999), epsilon=1e-08, weight_decay=1e-4)	step_size=20, gamma=0.7
	3	PointNet	SGD(learning_rate=0.001, momentum=0.9)	step_size=20, gamma=0.7
	4	PointNet	SGD(learning_rate=0.0001, momentum=0.9)	step_size=20, gamma=0.7
Completion	5	PCN	Adam(learning_rate=0.001, betas=(0.9, 0.999))	step_size=50, gamma=0.7
	6	PCN	Adam(learning_rate=0.0001, betas=(0.9, 0.999))	step_size=50, gamma=0.7
	7	PF-Net	Adam(learning_rate=0.0001, betas=(0.9, 0.999), epsilon=1e-05, weight_decay=0.001)	step_size=40, gamma=0.2
	8	PF-Net	Adam(learning_rate=0.00001, betas=(0.9, 0.999), epsilon=1e-05, weight_decay=0.001)	step_size=40, gamma=0.2

Table 8. Project experimentation plan, including network modelling and pre-conditions.

4.2 Experimental Result

Based on the experimental results reported, this section will evaluate from two main perspectives: the overall evaluation of the model performance and the comparative evaluation of the performance between pottery classification. From the perspective of further analysis, the multi-classification task is only applied to one model - PointNet, and the model performance under different optimizers and learning rates are mainly compared. The completion task involves two network models, so the training performance between the models will be compared and analyzed horizontally.

4.2.1 Classification Task

4.2.1.1 Overall Performance Evaluation

The multi-classification task will be evaluated mainly based on the four commonly used indicators of multi-classification tasks: Accuracy, Precision, Recall, and F1 Score. The results obtained based on the experimental plan are shown in Table 9.

Exp No.	Settings differences	Accuracy	Precision	Recall	F1 Score
1	Adam(learning_rate=0.001)	0.8260	0.8365	0.8260	0.8219
2	Adam(learning_rate=0.0001)	0.7536	0.7096	0.7536	0.7152
3	SGD(learning_rate=0.001)	0.5333	0.4802	0.5333	0.4667
4	SGD(learning_rate=0.0001)	0.6811	0.7219	0.6811	0.6771

Table 9. Summary of experimental results of model training for multi-classification task.

From the experimental results, the performance of training the model with Adam optimizer is significantly better than that with SGD optimizer. Under different learning rate settings (0.001 and 0.0001), the scores of Adam optimizer on the four evaluation metrics are about 0.3 and 0.1 higher respectively. Combined with the specific situation of the project and the characteristics of different optimizers, this difference may be due to the advantages of Adam optimizer in dealing with gradient sparsity and adaptive learning rate, making it more stable and efficient in the model training process of this project.

Further analysis of the two sets of experimental results of Adam optimizer shows that Experiment 1 with a learning rate of 0.001 achieved the best model performance. Compared with the setting of lower learning rate, this may be because a higher learning rate can converge to a better local optimal solution faster, and the adaptive mechanism of Adam optimizer can avoid the risk of shock caused by too high learning rate in this process. In Experiment 1, the scores of various metrics are relatively close, and Precision has the highest score, reaching 0.8365. This shows that this model can better identify positive pottery samples in the experiment.

From a broader perspective, although the PointNet model performs well on some mainstream datasets (such as ModelNet and ShapeNet), its performance on the multi-classification task of pottery is relatively average. The main factors affecting this result may be the number and quality of samples in the pottery dataset. First, the pottery dataset contains only 654 samples, which is much lower than the mainstream datasets ModelNet and ShapeNet, which usually contain thousands to tens of thousands of samples. This difference in the number of samples directly affects the training performance of the model. Fewer data samples make it difficult for the model to fully learn the characteristics of each category, thereby limiting its generalization ability. Second, from the perspective of data quality, the pottery dataset may have higher sample noise and larger intra-class differences, which makes it more difficult for the model to distinguish categories. In contrast, the ModelNet and ShapeNet datasets are usually carefully preprocessed, with higher discrimination between samples and more obvious category characteristics, so models trained on these datasets tend to perform better.

4.2.1.2 Performance Evaluation of Each Category

The evaluation of the performance of each category mainly focuses on the best performing experiment 1. The evaluation basis comes from its Precision-Recall Curve, ROC Curve and Confusion Matrix. Among them, each category is represented by a number in the project implementation, and its mapping relationship with the corresponding real category and sample quantity is shown in Figure 14.

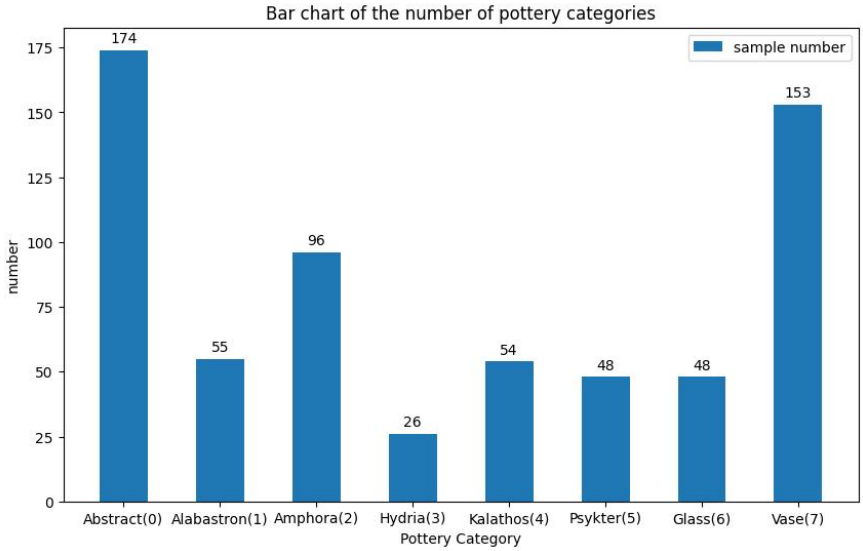


Figure 14. Bar chart of pottery categories in the dataset and their corresponding numbers, sample size.

- **Precision-Recall Curve Analysis**

The Precision-Recall Curve shows the trade-off between the Precision and Recall of the model. In multi-classification tasks, the closer the curve is to the upper right corner, the better the performance of the model for the classification, which means that the model can maintain high precision while maintaining high recall. Among them, AP (Average Precision) is the area under the Precision-Recall Curve, that is, the area under the P-R curve. The value of AP ranges from 0 to 1. The closer the value is to 1, the better the performance of the model for the classification.

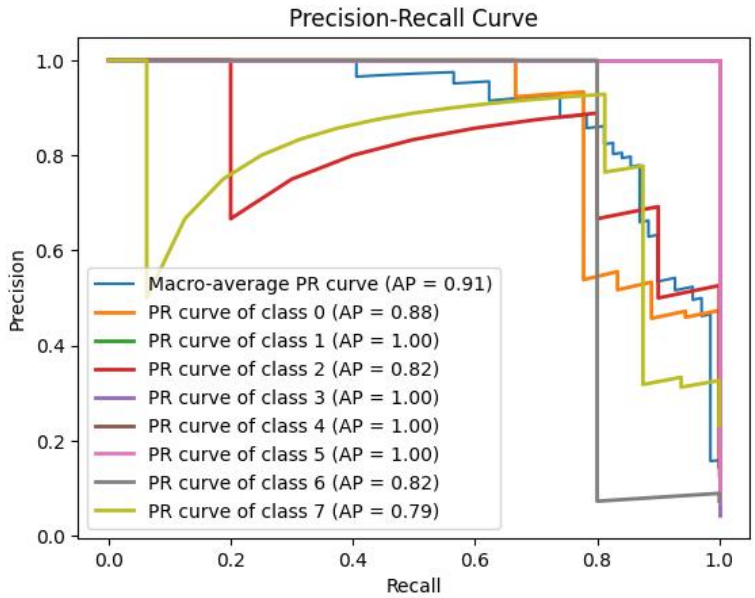


Figure 15. Precision-Recall Curve, resultant data from Exp No.1.

For the analysis of Figure 15, the AP of the Macro-average PR curve is equal to 0.91, indicating that the PointNet model has a very good performance overall. The area under the precision-recall curve of most pottery categories is large, which means that this model can balance precision and recall well in most categories. When it comes to the PR curve of each category, the performance of different categories is significantly different, ranging from 0.79 to 1.00. This shows that the model is easier to classify some categories. For example, the AP of 1, 3, 4, and 5 reached 1.00, indicating that the model grasped the characteristics of these pottery categories very accurately with almost no error. At the same time, the AP of categories 2, 6, and 7 was lower, indicating that these categories may be confused and the model had difficulty distinguishing these pottery categories.

- **ROC and AUC Analysis**

Receiver Operating Characteristic (ROC) shows the trade-off between the True Positive Rate (TPR, also known as Recall) and False Positive Rate (FPR) of the model. Area Under the Curve (AUC) is the area under the ROC curve, ranging from 0 to 1. The closer the AUC is to 1, the better the classifier distinguishes.

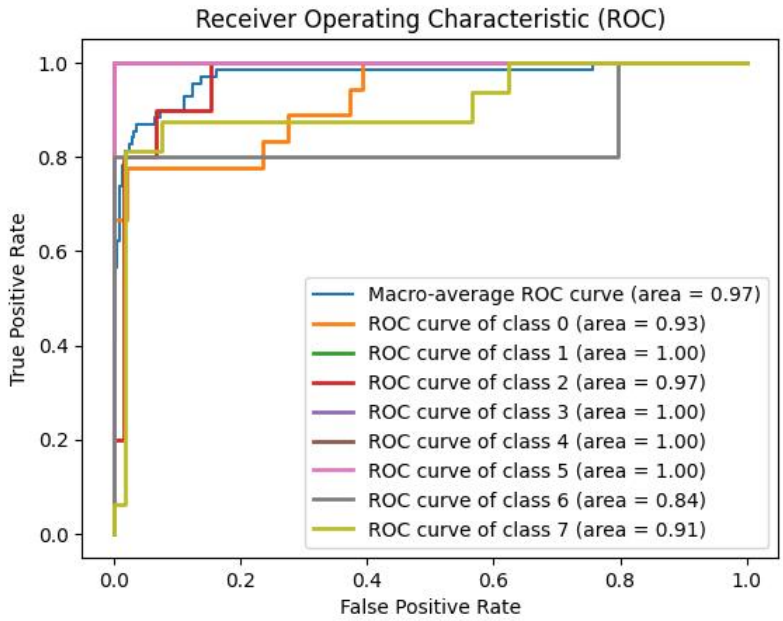


Figure 16. ROC Curve, resultant data from Exp No.1.

For a specific analysis of Figure 16, the area value of the Macro-average ROC curve reaches 0.97. This shows that the PointNet model performs very well overall, and most categories can effectively distinguish between positive and negative samples. The AUC of most categories is close to 1.00, indicating that the model classifies almost perfectly on these categories, but the performance of category 6 (AUC = 0.84), category 7 (AUC = 0.91), and category 0 (AUC = 0.93) is slightly lower, indicating that the model has some misclassification problems in these two categories.

- **Confusion Matrix Analysis**

Confusion Matrix is suitable for the pottery multi-classification problem of this project. It shows the matching between the model's prediction results and the actual labels in a table. The main diagonal elements represent the number of correct classifications of each category by the model. The higher the value on the main diagonal, the better the classification performance of the model. Each off-diagonal element represents the number of times a category is misclassified as another category. The larger the number, the more serious the confusion of the model on this pair of categories.

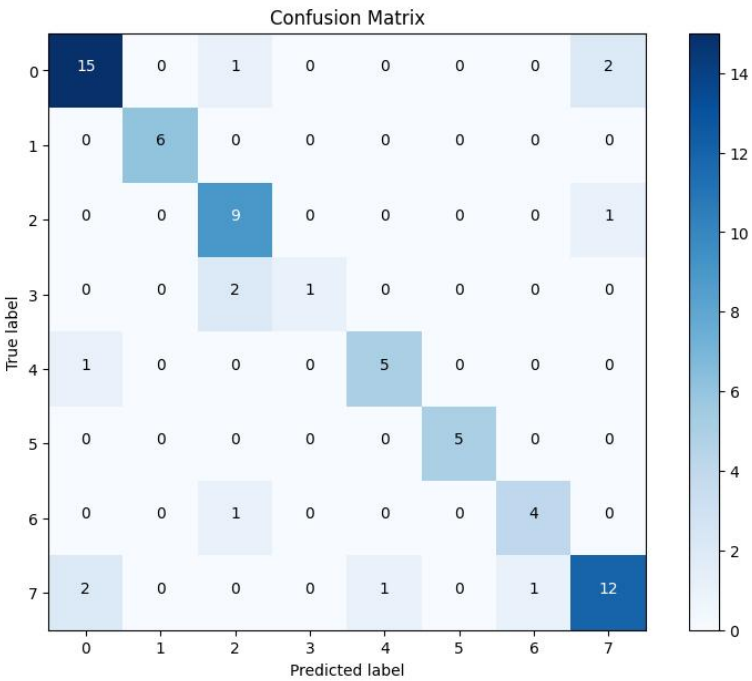


Figure 17. Confusion Matrix, resultant data from Exp No.1.

Looking back at Figure 17 above, except for category 1 and category 5, all other categories have misclassification. Considering the cardinality of each category sample in the test set, measured by the proportion of misclassification frequency, category 3 has the highest misclassification probability, reaching 0.66. In addition, the misclassification of categories 0, 4, 6, and 7 is also obvious. Relatively speaking, the probability of misclassification of the remaining categories remains within the estimated range.

In summary, in terms of the prediction accuracy of each category, the overall performanceis less ideal for categories 3, 6, and 7. The reason is that the total number of samples in category 3 is too small, only 26, which makes it difficult for the model to fully learn the characteristics of this category, thus affecting its prediction accuracy. The problems with categories 6 and 7 may be due to the large differences in sample characteristics within the category, making it difficult for the model to capture the common characteristics of these categories, resulting in a decrease in classification performance.

4.2.2 Completion Task

4.2.2.1 Overall Performance Evaluation

According to the experimental plan, the completion task was based on two network models, PCN and PF-Net, and a total of 4 training experiments were conducted. Table 10 below shows the values of its main measurement indicator ChamferDistance and samples taken from the experimental performance pictures. Please note that for a more intuitive display and comparison, the results of Chamfer Distance in the table are magnified 1000 times by multiplying "1e3".

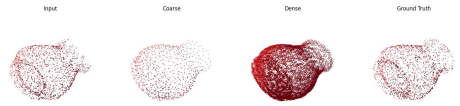
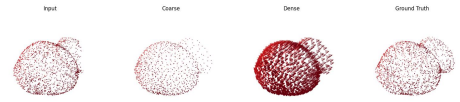
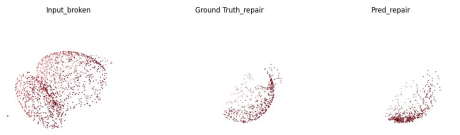
Exp No.	Model	Learning Rate	Chamfer Distance	Example of result images input predict result ground turth
5	PCN	0.001	5.6238	
6	PCN	0.0001	5.7984	
7	PF-Net	0.0001	10.4123	
8	PF-Net	0.00001	11.3154	

Table 10. Summary of the results of the pottery completion task.For visualisation and comparison, the Chamfer Distance results are magnified 1000 times by multiplying by "1e3".

From the above results, it can be seen that the difference between the predicted graphics of the PCN model and the ground turth graphics point cloud is significantly better than that of the PF-Net model. This may be due to the large differences in the principles and logic of the training of the two models.

Using the Chamfer Distance metric to measure the training performance under the same model, the Chamfer Distance obtained in the two experiments of the PCN

model did not change significantly due to the reduction of the learning rate, and remained stable at around 5.7. Compared with the metric performance of the PCN model in public large-scale data set resources (such as the Chamfer Distance result of ShapeNet is 9.63[29]), this model performs very well in the pottery multi-classification task. The results of the two experiments of the PF-Net model are not significantly different, and the average is stable at 10.9. As for the performance of the learning rate on the two models, the performance is slightly better under a larger learning rate than under a smaller learning rate.

From the result images, it can be seen that both models can effectively predict the completion sample based on the broken pottery sample, but the specific effects are slightly different. The PCN model has a strong learning ability for the overall shape of pottery, and can achieve relatively smooth completion for categories with simple shapes, obvious features, and a large number of cardinalities, with relatively ideal results. However, the completion parts generated by the PF-Net model based on damaged samples may be misplaced or missing in some cases, and the results are relatively unstable, resulting in relatively low accuracy.

4.2.2.2 Performance Evaluation of Each Category

The model training performance of each category will be evaluated by two methods: Chamfer Distance value and result image visual comparison.

- **Chamfer Distance Analysis**

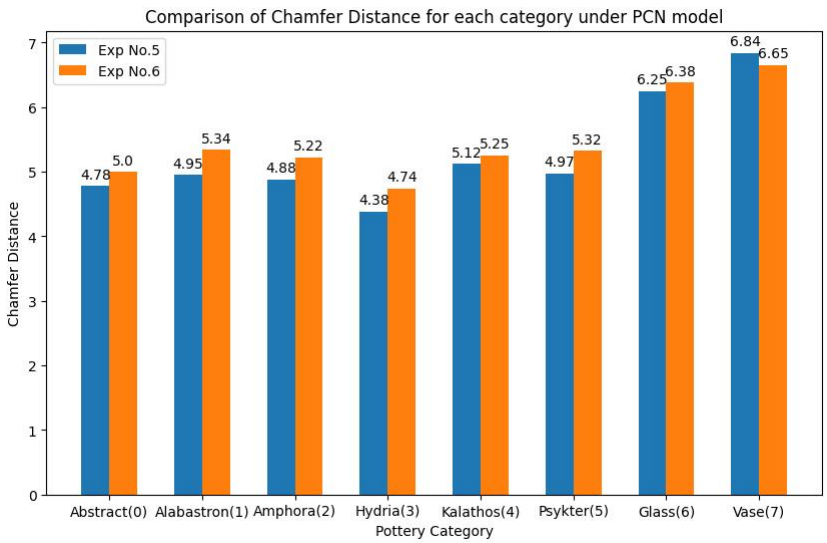


Figure 18. Comparison of Chamfer Distance for each category under PCN model.

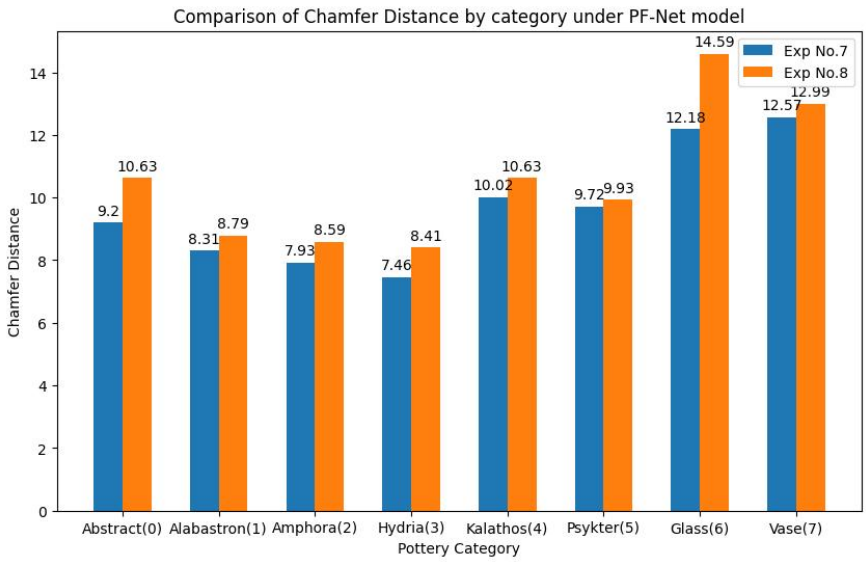


Figure 19. Comparison of Chamfer Distance by category under PF-Net model.

Figure 18 and Figure 19 show the Chamfer Distance of each category using the two models. Two significant features can be seen from these two bar charts.

First, when the learning rate is large (Exp No. 5 and Exp No. 7), the Chamfer Distance value is small, and the prediction performance is closer to the ground truth. It can be seen that under the conditions of this project, the learning rate is set to 0.001 and 0.0001 to obtain the best model performance. Secondly, the Chamfer Distance values of the categories "Glass (6)" and "Vase (7)" after training the two models are higher than those of other categories, that is, the predicted completion performance is far from the ground truth. This may be similar to the reason for the poor performance in multi-classification tasks, that is, the sample features within the category are too different, which makes it more difficult for the model to learn the mapping relationship from damage to completion. For the PCN model, the Chamfer Distance values of the remaining categories are very different. For the PF-Net model, the performance of the categories 'Alabastron(1)', 'Amphora(2)', and 'Hydria(3)' is better than the other three categories. Considering the data set, this may be because the number of samples in these three categories is small and their characteristics are more obvious, so the model can be more targeted when learning. However, it should be noted that this situation also has certain limitations, that is, the generalization ability of the model may be insufficient.

- **Result image comparison**

The following table shows representative samples of the average performance in the result images of the PCN and PF-Net models respectively.

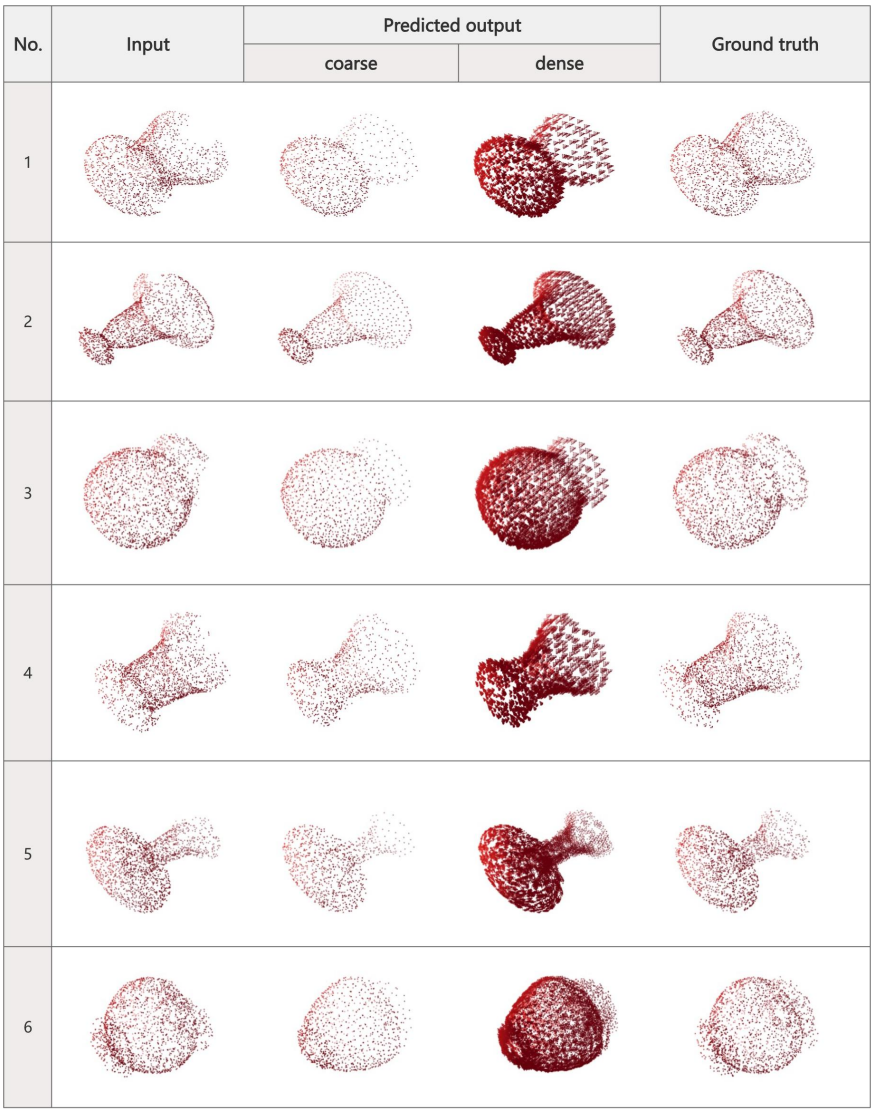


Figure 20. Examples of PCN Model completion result images.

The above results indicate that PCN demonstrates a strong learning ability for capturing the overall shape features of the model, effectively predicting similar completion samples across most categories. The completed point cloud part is more coherent and more consistent with the shape of the real pottery. However, since the broken pottery samples are handmade, most of the gaps are small and concentrated on the edges, and there are insufficient samples for large-area defects. For this case, the performanceof the PCN model still needs to be verified.

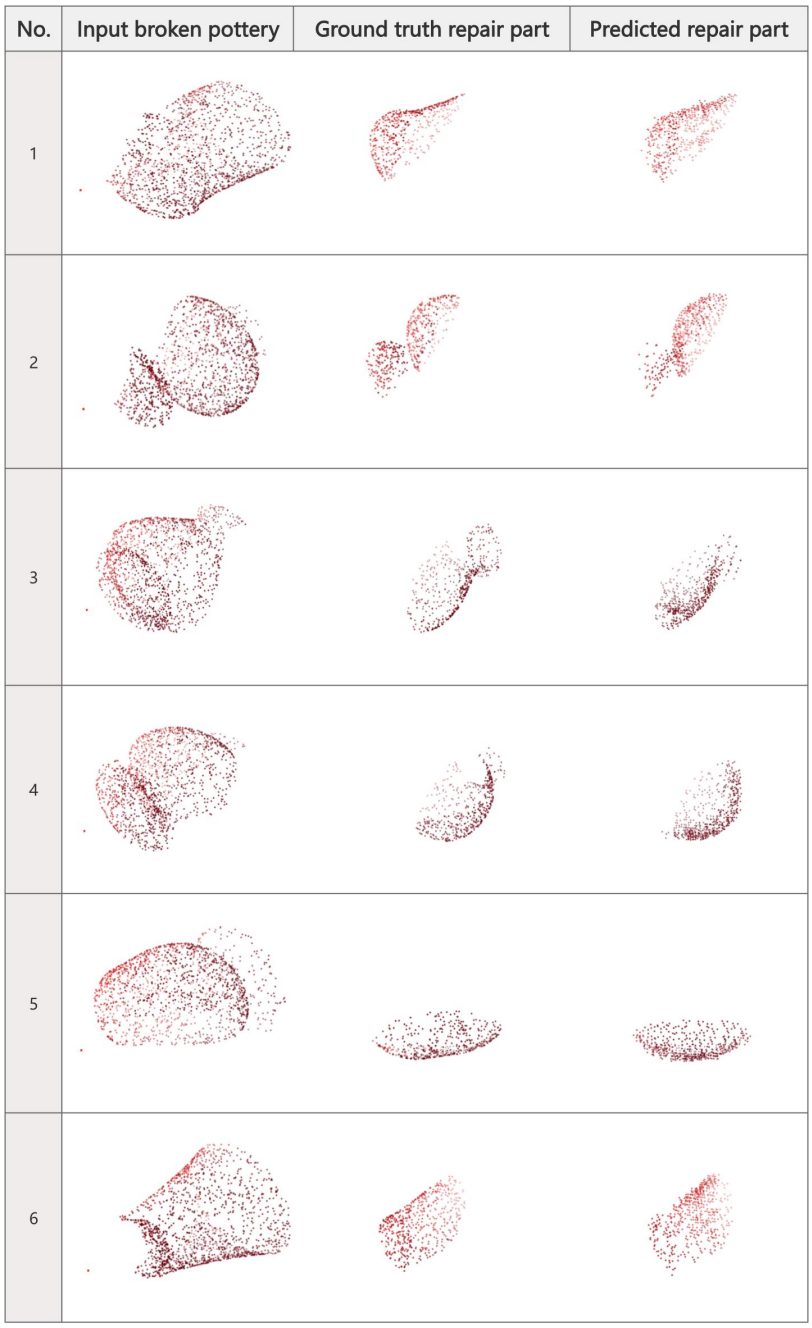


Figure 21. Examples of PF-Net Model completion result images.

From the above results, it is evident that the completed parts generated by the PF-Net model generally align with the expected shape when compared to the ground truth. However, a common issue is the presence of scattered edges in the point cloud graphics. Especially when the completed parts include more complex shape changes (such as the pottery rims in samples No. 3 and No. 4 in the figure above), the generated performance is missing or has large errors. After analysis, this may be because the training of PF-Net may be sensitive to the size and diversity of the data set. The number of samples of the pottery data set used for training in this project is small, and the model is prone to overfitting on a small amount of data, resulting in poor generalization of the model on the test set. In addition, PF-Net generates more refined point clouds in each layer, but this method may ignore some key local features. For the task of pottery completion, local features (such as edges or specific geometric shapes) may be very important. If the model cannot fully learn these local features, the generated completion results may be far from the actual situation.

5 Conclusion

This chapter will summarize the project in three parts. First, combined with the expected aim and objectives, the main results and significance of the research are explained. Secondly, based on the actual results of the classification and completion tasks, the shortcomings and challenges encountered in the project are discussed. Finally, based on the project experience, some suggestions are given for future research-related work.

5.1 Overall review

Reviewing the aim originally set for the project, that is, exploring the feasibility of applying 3D graphics classification and completion algorithms to the automatic classification and damage completion of pottery, obtaining relatively ideal results and valuable evaluation results, and accumulating engineering implementation experience of machine learning in the field of cultural relics protection. The following content will start from the expected content of the aim, and review from three perspectives: model performance, engineering development practice, and practical significance of the research.

5.1.1 Model Performance

The three deep learning models involved in the project all showed their obvious feature extraction and simulation generation capabilities, which can effectively and accurately classify and complete pottery automatically. Among them, the model performance of PointNet and PCN are relatively excellent, and can become valuable reference cases for further research or simple practical applications. However, under the limited data set samples and hardware facilities of this project, the completion performance of PF-Net did not meet expectations, mainly reflected in the lack of

accuracy and precision of the prediction results. This problem is more obvious when the complexity of its network structure leads to high resource consumption.

5.1.2 Engineering Development

The progress of the project engineering development process is relatively in line with expectations. It has successfully and completely realized the necessary steps from data set selection, theoretical scheme design, code implementation, and result presentation, analysis and evaluation, forming a set of reference-worthy solutions for 3D pottery multi-classification and completion tasks. For example, a new broken pottery data set was manually produced to achieve supplementary training. It provides valuable practical experience for the research topic of 3D graphics in machine learning.

5.1.3 Practical significance of the research

This project focuses on the practical application field of cultural relics protection. By using 3D scanning data of ancient pottery as a direct data source, it aims to verify the feasibility of deep learning algorithms in actual needs. This project combines theory with practice, not only demonstrating the practical application potential of deep learning algorithms in cultural relics protection, but also filling the current research gap in this field to a certain extent, providing new technical support and solutions for cultural relics protection.

5.2 Limitations

First, due to the limitation of hardware resources and time, this project has obvious deficiencies in data set capacity and quality. This problem is particularly prominent, especially in the application of PF-Net model, where the insufficient number of samples and the lack of fineness of the graphics in the feature-prominent parts have a direct impact on the performance of the model. Secondly, during the experiment, the adjustment and attempts of hyperparameters were not diverse enough. This led to the possibility that the model failed to achieve its optimal performance. Hyperparameter settings are crucial to the performance of deep learning models, and the exploration and experimentation of different hyperparameter combinations in this project are relatively limited, which may affect the optimization of the final results and the accuracy of the model.

5.3 Future work

Combined with the practical experience of this project, future work can be improved and expanded from the following aspects. First, the data set needs to be expanded and optimized, and the number of samples and data quality need to be increased to support more complex model training and improve prediction accuracy.

Second, more extensive hyperparameter adjustments should be made and more combinations should be explored to optimize model performance and accuracy. Further, it is possible to consider adopting more advanced model architectures or combining multiple deep learning techniques to improve the performance of pottery classification and completion tasks. At the same time, we will expand the application scenarios of research and verify the applicability of the model in other cultural relics categories and protection tasks to expand the influence of research results. Finally, investing in higher-performance computing resources and technology platforms will help process large-scale data and train complex models, thereby promoting further development in the field of cultural relics protection.

Acknowledgments.
First of all, I would like to thank Dr Varun Ojha for his patient guidance and indispensable feedback throughout the project. Whether in the background investigation, technology development or paper writing stage, Dr Varun Ojha provided a lot of valuable experience and practical suggestions. Every time I reported the progress, he also gave me a lot of encouragement and affirmation, which gave me the confidence to complete the entire project independently.

In addition, I would like to thank the scholars who insist on studying and concentrating on the field of machine learning vision and the people who provide open source resources for free. Standing on their shoulders, I can realize and see the desired effect with my own hands.

Finally, thank you for sticking to yourself in the topic selection stage. The idea of this project came from the Gansu painted pottery exhibition I saw at the National Museum of China many years ago. The simple pottery with a heavy sense of history deeply attracted me. Although this topic is very challenging for me, I still hope to take the opportunity of this project to contribute my meager strength to the protection of cultural relics.

References

1. L. Bayliss and L. D. Jones, "The role of artificial intelligence and machine learning in predicting orthopaedic outcomes," The Bone & Joint Journal, vol. 101-B, no. 12, pp. 1476–1478, Dec. 2019, doi: https://doi.org/10.1302/0301-620x.101b12.bjj-2019-0850.r1.
2. E. Oates, "Heritage Pulse: Spotlight on Skills," CultureHive, 2024. https://www.culturehive.co.uk/resources/heritage-pulse-spotlight-on-skills/ (accessed Jul. 24, 2024).
3. N. Farajzadeh and M. Hashemzadeh, "A deep neural network based framework for restoring the damaged persian pottery via digital inpainting," Journal of Computational Science, vol. 56, p. 101486, Nov. 2021, doi: https://doi.org/10.1016/j.jocs.2021.101486.
4. R. Hermoza and I. Sipiran, "3D Reconstruction of Incomplete Archaeological Objects Using a Generative Adversarial Network," arXiv (Cornell University), Jun. 2018, doi: https://doi.org/10.1145/3208159.3208173.

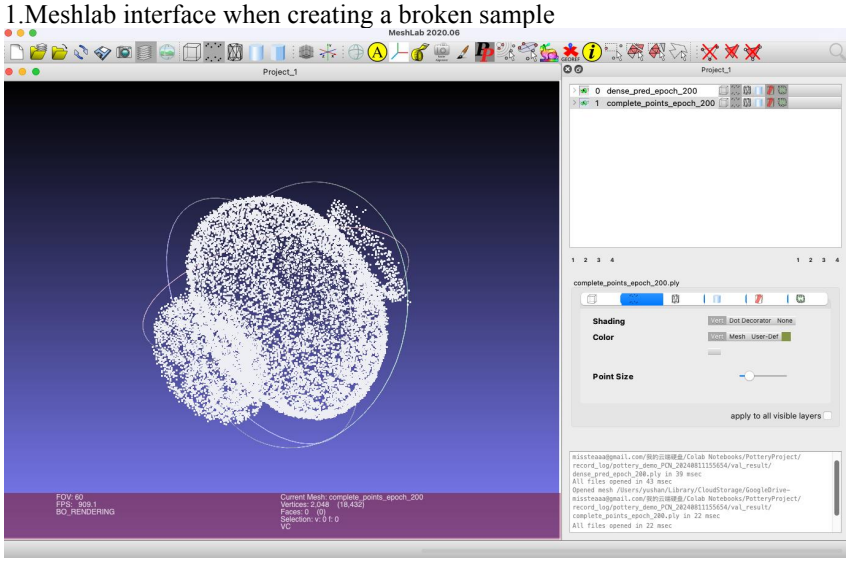
5. L. Acke, K. De Vis, S. Verwulgen, and J. Verlinden, “Survey and literature study to provide insights on the application of 3D technologies in objects conservation and restoration,” Journal of Cultural Heritage, vol. 49, pp. 272–288, May 2021, doi: https://doi.org/10.1016/j.culher.2020.12.003.
6. F. Tian et al., “An empirical study of virtual museum based on dual-mode mixed visualization: the Sanxingdui bronzes,” Heritage science, vol. 12, no. 1, May 2024, doi: https://doi.org/10.1186/s40494-024-01241-1.
7. C. Flagg and O. Frieder, “Reconstruction of Artifacts from Digital Image Repositories,” Journal on Computing and Cultural Heritage, vol. 16, no. 1, pp. 1–22, Dec. 2022, doi: https://doi.org/10.1145/3552298.
8. A. Koutsoudis, G. Pavlidis, V. Liami, D. Tsiafakis, and C. Chamzas, “3D Pottery content-based retrieval based on pose normalisation and segmentation,” Journal of Cultural Heritage, vol. 11, no. 3, pp. 329–338, Jul. 2010, doi: https://doi.org/10.1016/j.culher.2010.02.002.
9. D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, and A. A. Efros, “Context Encoders: Feature Learning by Inpainting,” arXiv:1604.07379 [cs], Nov. 2016, Available: https://arxiv.org/abs/1604.07379
10. M. Xiao, G. Li, L. Peng, Y. Lv, and Y. Mao, “Completion of images of historical artifacts based on salient shapes,” Optik, vol. 127, no. 1, pp. 396–400, Jan. 2016, doi: https://doi.org/10.1016/j.ijleo.2015.10.002.
11. G. Liu, F. A. Reda, K. J. Shih, T. Wang, A. Tao, and B. Catanzaro, “Image Inpainting for Irregular Holes Using Partial Convolutions,” Apr. 2018, doi: https://doi.org/10.48550/arxiv.1804.07723.
12. Anupam, P. Goyal, and Sapan Diwakar, “Fast and Enhanced Algorithm for Exemplar Based Image Inpainting,” Infoscience (Ecole Polytechnique Fédérale de Lausanne), Nov. 2010, doi: https://doi.org/10.1109/psivt.2010.61.
13. K. Adamczak et al., “Digital restoration and 3D visualisation of the early Neolithic pottery from Ryńsk, Poland,” Digital Applications in Archaeology and Cultural Heritage, vol. 33, pp. e00337–e00337, Jun. 2024, doi: https://doi.org/10.1016/j.daach.2024.e00337.
14. A. Koutsoudis, G. Pavlidis, F. Arnaoutoglou, D. Tsiafakis, and C. Chamzas, “Qp: A tool for generating 3D models of ancient Greek pottery,” Journal of Cultural Heritage, vol. 10, no. 2, pp. 281–295, Apr. 2009, doi: https://doi.org/10.1016/j.culher.2008.07.012.
15. W. Wang, Q. Huang, S. You, C. Yang, and U. Neumann, “Shape Inpainting Using 3D Generative Adversarial Network and Recurrent Convolutional Networks,” International Conference on Computer Vision, Oct. 2017, doi: https://doi.org/10.1109/iccv.2017.252.
16. X. Qun and Y. Liu, “3D shape completion via deep learning: a method survey,” Journal of Graphics, vol. 44, no. 2, pp. 201–215, 2023, doi: https://doi.org/10.11996/JG.j.2095-302X.2023020201.
17. C. R. Qi, H. Su, K. Mo, and Guibas, Leonidas J, “PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation,” Dec. 2016, doi: https://doi.org/10.48550/arxiv.1612.00593.
18. C. R. Qi, L. Yi, H. Su, and Guibas, Leonidas J, “PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space,” arXiv.org, 2017. https://arxiv.org/abs/1706.02413

19. Y. Li, R. Bu, M. Sun, W. Wu, X. Di, and B. Chen, “PointCNN: Convolution On X-Transformed Points,” arXiv (Cornell University), Jan. 2018.
20. Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, “Dynamic Graph CNN for Learning on Point Clouds,” ACM Transactions on Graphics, vol. 38, no. 5, pp. 1–12, Nov. 2019, doi: https://doi.org/10.1145/3326362.
21. W. Wu, Z. Qi, and L. Fuxin, “PointConv: Deep Convolutional Networks on 3D Point Clouds,” arXiv:1811.07246 [cs], Nov. 2020, Available: https://arxiv.org/abs/1811.07246
22. W. Yuan, Tejas Khot, D. Held, C. Mertz, and M. Hebert, “PCN: Point Completion Network,” Sep. 2018, doi: https://doi.org/10.1109/3dv.2018.00088.
23. L. P. Tchapmi, Vineet Kosaraju, Hamid Rezatofighi, I. Reid, and S. Savarese, “TopNet: Structural Point Cloud Decoder,” Jun. 2019, doi: https://doi.org/10.1109/cvpr.2019.00047.
24. Z. Huang, Y. Yu, J. Xu, F. Ni, and X. Le, “PF-Net: Point Fractal Network for 3D Point Cloud Completion.”
25. H. Xie, H. Yao, S. Zhou, J. Mao, S. Zhang, and W. Sun, “GRNet: Gridding Residual Network for Dense Point Cloud Completion,” arXiv (Cornell University), Jun. 2020, doi: https://doi.org/10.48550/arxiv.2006.03761.
26. X. Wen et al., “PMP-Net: Point Cloud Completion by Learning Multi-step Point Moving Paths,” arXiv (Cornell University), Jun. 2021, doi: https://doi.org/10.1109/cvpr46437.2021.00736.
27. Z. Wu et al., “3D ShapeNets: A deep representation for volumetric shapes,” IEEE Xplore, Jun. 01, 2015. https://ieeexplore.ieee.org/document/7298801
28. A. X. Chang et al., “ShapeNet: An Information-Rich 3D Model Repository,” arXiv (Cornell University), Dec. 2015.
29. “Papers with Code - ShapeNet Benchmark (Point Cloud Completion),” Paperswithcode.com, 2022. https://paperswithcode.com/sota/point-cloud-completion-on-shapenet (accessed Aug. 12, 2024).
30. I. Goodfellow et al., “Generative Adversarial Nets,” 2014. Available: https://arxiv.org/pdf/1406.2661
31. P. Achlioptas, O. Diamanti, I. Mitliagkas, and L. Guibas, “Learning Representations and Generative Models for 3D Point Clouds,” arXiv:1707.02392 [cs], Jun. 2018, Accessed: May 13, 2022. [Online]. Available: https://arxiv.org/abs/1707.02392
32. charlesq34, “charlesq34/pointnet,” GitHub, Sep. 19, 2019. https://github.com/charlesq34/pointnet
33. qinglew, “GitHub - qinglew/PCN-PyTorch: Implementation of PCN(Point Completion Network) in PyTorch.,” GitHub, 2021. https://github.com/qinglew/PCN-PyTorch (accessed Aug. 14, 2024).
34. zztianzz, “GitHub - zztianzz/PF-Net-Point-Fractal-Network: CVPR2020 PF-Net: Point Fractal Network for 3D Point Cloud Completion,” GitHub, 2020. https://github.com/zztianzz/PF-Net-Point-Fractal-Network (accessed Aug. 14, 2024).

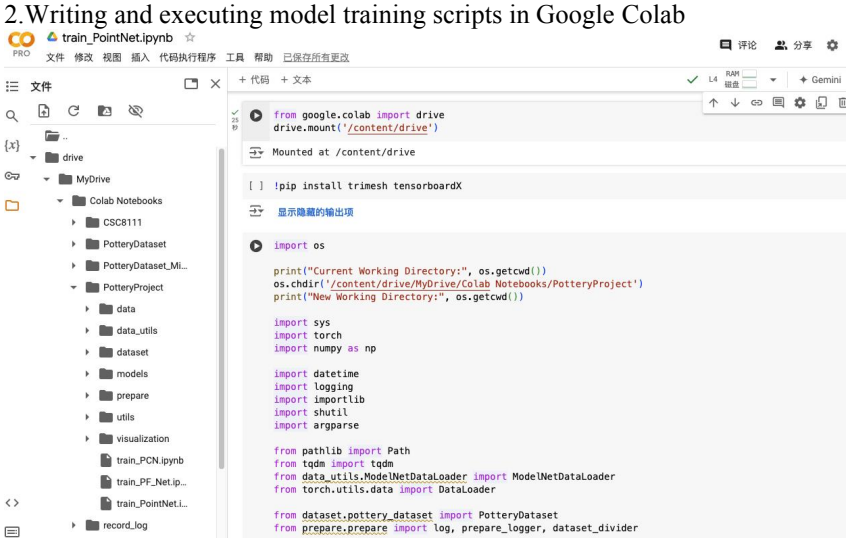
Appendix

Project related Github project link: https://github.com/XueningLii/3D-Pottery-Classification-and-Completion

1.Meshlab interface when creating a broken sample



2.Writing and executing model training scripts in Google Colab



3. View real-time score results during training

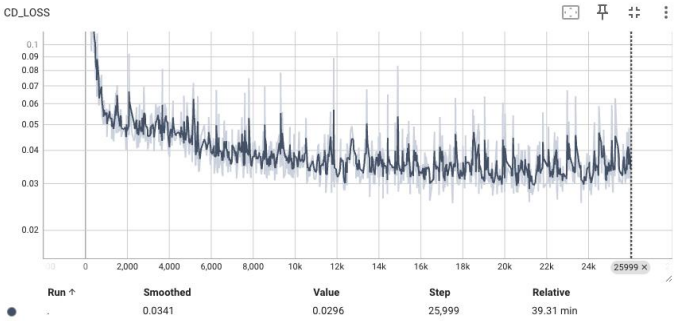
```
                'class_acc': class_acc,
                'model_state_dict': classifier.state_dict(),
                'optimizer_state_dict': optimizer.state_dict(),
            }
            torch.save(state, savepath)

        val_step += 1

log(log_fd, 'End of training...')
```

```
2024-08-08 14:20:35 ==> Start training...
2024-08-08 14:20:39 ==> Train Epoch [000/200]: Train Instance Accuracy = 0.251923
2024-08-08 14:20:39 ==> Test Instance Accuracy: 0.279412, Class Accuracy: 0.214062
2024-08-08 14:20:39 ==> Best Instance Accuracy: 0.279412, Class Accuracy: 0.214062
2024-08-08 14:20:39 ==> Save model...
2024-08-08 14:20:39 ==> Saving at record_log/pottery_demo_PointNet++_20240808142020/checkpoint/best_model.pth
2024-08-08 14:20:43 ==> Train Epoch [001/200]: Train Instance Accuracy = 0.319231
2024-08-08 14:20:44 ==> Test Instance Accuracy: 0.308824, Class Accuracy: 0.200521
2024-08-08 14:20:44 ==> Best Instance Accuracy: 0.308824, Class Accuracy: 0.214062
2024-08-08 14:20:44 ==> Save model...
2024-08-08 14:20:44 ==> Saving at record_log/pottery_demo_PointNet++_20240808142020/checkpoint/best_model.pth
2024-08-08 14:20:48 ==> Train Epoch [002/200]: Train Instance Accuracy = 0.294231
2024-08-08 14:20:48 ==> Test Instance Accuracy: 0.323529, Class Accuracy: 0.250521
2024-08-08 14:20:48 ==> Best Instance Accuracy: 0.323529, Class Accuracy: 0.250521
2024-08-08 14:20:48 ==> Save model...
2024-08-08 14:20:48 ==> Saving at record_log/pottery_demo_PointNet++_20240808142020/checkpoint/best_model.pth
2024-08-08 14:20:52 ==> Train Epoch [003/200]: Train Instance Accuracy = 0.319231
2024-08-08 14:20:53 ==> Test Instance Accuracy: 0.294118, Class Accuracy: 0.209375
2024-08-08 14:20:53 ==> Best Instance Accuracy: 0.323529, Class Accuracy: 0.250521
```

4. Monitor the downward trend of loss



5. View the result image