The installation of this capstone project requires performing multiple procedures. Please follow the below instructions.

## Machine Learning Model Installation

Since the instructions provided by the model creator cannot install the model, therefore follow the below steps to install the Softgroup machine learning model correctly.

1. Install Anaconda in Ubuntu OS for environment management.
2. Create and activate a conda environment

|  |
| --- |
| conda create -n model python=3.7  conda activate model |

1. Clone the GitHub repository

|  |
| --- |
| git clone https://github.com/thangvubk/SoftGroup.git |

1. Install the dependencies

|  |
| --- |
| cd *the directory of the repo clone location*  conda install pytorch==1.11.0 torchvision==0.12.0 torchaudio==0.11.0 cudatoolkit=10.2 -c pytorch  pip install spconv-cu102  pip install -r requirement.txt |

1. Install build requirement

|  |
| --- |
| sudo apt-get install libsparsehash-dev |

1. Setup

|  |
| --- |
| python setup.py build\_ext develop |

## S3DIS Dataset Preparation

Since the S3DIS dataset is not opened for public to download, it requires users request the dataset through application (Stanford, n.d.), therefore, to ease the S3DIS dataset preparation, the dataset has been uploaded to the personal OneDrive [Stanford3dDataset\_v1.2.zip](https://connectpolyu-my.sharepoint.com/:u:/g/personal/21019855d_connect_polyu_hk/Edid4mLVzxlBkNaOM6GMnNEB4KaSRpodjYcjuG7r_gKGGg?e=I1hEgX). Download the zipped dataset, and unzip the dataset in the path of “/SoftGroup/dataset/s3dis”, follow the below steps:

1. Open the “prepare\_data\_inst.py” in “SoftGroup/dataset/s3dis”
2. Go to line 139 and comment this line
3. Uncomment line 140 and save
4. Run the command

|  |
| --- |
| bash prepare\_data.sh |

1. Wait for the program complete
2. Back to the prepare\_data\_inst.py
3. Comment line 140 and uncomment line 141 and save
4. Run the previous command
5. Repeat step 5-8, comment the current line and uncomment the next line, until line 145 comment and uncomment.

## S3DIS Pretrained Model Preparation

In this project, since the time cost and computational power are too demanding, therefore, pretrained model is adopted and it is recommend to use pretrained model provided by the model creator. There are 2 pretrained models for S3DIS dataset, there is only a slight score difference between 2 pretrained model, in this project “SoftGroup” is adopted instead of “SoftGroup++”.

1. Go to <https://drive.google.com/file/d/1-f7I6-eIma4OilBON928N6mVcYbhiUFP/view> to download the pretrained model
2. Put the pth file in the repository directory

## Perform 3D Instance Segmentation on S3DIS Dataset

Before, perform instance segmentation, by default all the pth files in “/dataset/s3dis/preprocess” with the prefix “Area\_5\_” will be input to the model to perform 3D instance segmentation, the computer resource will be very demanding if there are too many input files, and lead to failure since out of memory exception thrown. Therefore, it is recommended to leave only 1 or 2 “Area\_5\_” files in the “/dataset/s3dis/preprocess” directory each runtime. Run the following command to perform 3D instance segmentation.

|  |
| --- |
| ./tools/dist\_test.sh $CONFIG\_FILE $CHECKPOINT $NUM\_GPU --out results/ |

$CONFIG\_FILE: this should be replaced by:

|  |
| --- |
| ./configs/softgroup/softgroup\_s3dis\_fold5.yaml |

$CHECKPOINT: this should be replaced by:

|  |
| --- |
| *The path of the location of the downloaded pth file in 3.3.2. For example,*  ./softgroup\_s3dis\_spconv2.pth |

$NUM\_GPU: this should be replaced by the number of GPUs the PC has, if only 1 GPU, the give 1 as the command parameter.

Wait for the program to finish performing this task.

## Convert the 3D Instance Segmentation Result Output File

Run the following command to convert the output file to a PLY file.

|  |
| --- |
| cd *To the root directory of the cloned repository*  python3 ./tools/visualization.py --prediction\_path results/ --room\_name Area\_5\_conferenceRoom\_1 --task instance\_pred --out room1.ply |

“Area\_5\_conferenceRoom\_1” can be replaced by the file name performed instance segmentation. “room1.ply” can be replaced by a custom filename, but “.ply” must be included in the end of the filename.

By default, the output PLY file will be located in the root directory of the repository.

## Path Finding Prototype Installation

There are multiple python programs developed for this capstone project, which requires another conda environment. Run the following commands, make sure deactivate other conda environment first.

|  |
| --- |
| conda create -n fyp python=3.7  conda activate fyp  conda install numpy  conda install pytorch==1.11.0 torchvision==0.12.0 torchaudio==0.11.0 cudatoolkit=10.2 -c pytorch  conda install -c conda-forge plyfile  pip install open3d |

## Path Finding Preprocessing

Clone the GitHub repository.

|  |
| --- |
| git clone httsp://github.com/jimleungcc/fyp.git |

Copy the ply file “room1.ply” to the “custom\_room” directory of the repository just cloned.

Run the splitPLY.py by

|  |
| --- |
| cd custom\_room  python3 ./splitPLY.py |

Then, input the ply file name (include “.ply”) and wait for the point clouds extracting, when this process is finished, a new directory name “room1” (the directory name follows on the input ply file name) will appear in the root directory.

Here is an optional step for converting the encoding method of the extracted PLY files, move the “convertPLYencoding.py” to the directory “room1”, and then run the convertPLYencoding.py by

|  |
| --- |
| cd room1  python3 ./convertPLYencoding.py |

Use MeshLab to open those extracted ply files, and selected and rename the required ply files with meaningful name, for example, “westchair\_1” to indicate the first chair in the west/left.

If there is an object split by the model to multiple PLY files, which requires to combine into a single PLY file, use the combinePLY.py to do so.

Run the combinePLY.py by

|  |
| --- |
| python3 ./combinePLY.py |

Then, input the number of the PLY files to combine, and input the file name of the ply files required one by one, finally enter the name of the output file. Wait for the process to be completed, and the combined PLY file will be generated.

Open the python file “shortest\_path\_finding.py” using IDE, follow the instructions of the comments to modify related the program.

## Perform Shortest Path Finding

After modification, run the python file “shortest\_path\_finding.py” by

|  |
| --- |
| python3 ./shortest\_path\_finding.py |

Wait for the processing, and an open3D window pops up to display the shortest path.

## Run Demonstration

Since the steps to perform shortest path finding are quite time consuming, therefore, there are 2 demonstration folders named “Room1\_Demo” and “Room2\_Demo” in this repository, to examine the demonstration, run the “room1.py” or “room2.py” to see the demonstration of shortest path finding of room1 and room2 respectively.