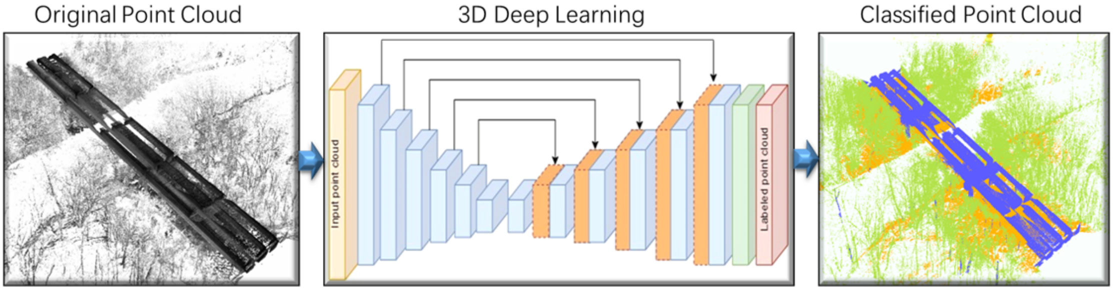
# User Manual of the DeepHyd system: Deep Learning-based Point Cloud Classification of Hydraulic Structures

Center for Applied Geographic Information Science (http://gis.charlotte.edu)
University of North Carolina at Charlotte
Contact: Wenwu Tang, Ph.D., (WenwuTang@uncc.edu)

**(Last Update by: 1-30-2022)**



**1. Installation of the DeepHyd system** **1**

1.1 Installation of Command Line Interface of the DeepHyd system: DeepHyd-CLI 2

1.2. Installation of Web-based Interface of the DeepHyd System: DeepHyd-Web 3

**2. User Manual of the DeepHyd System** **5**

2.1 Using Command Line Interface (CLI) of the DeepHyd System 5

2.2 Using Web-based Interface of the DeepHyd System 8

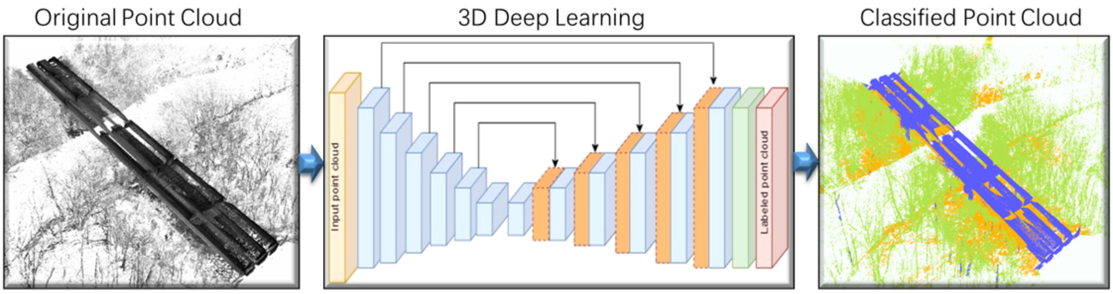
Appendix A **14**

Appendix B **15**

Appendix C **16**

# User Manual of the DeepHyd system: Deep Learning-based Point Cloud Classification of Hydraulic Structures

Center for Applied Geographic Information Science (http://gis.charlotte.edu)
University of North Carolina at Charlotte
Contact: Wenwu Tang, Ph.D., (WenwuTang@uncc.edu)



DeepHyd is a cutting-edge 3D deep learning-based tool for the point cloud classification of hydraulic structures. It is based on a two-tiered modeling architecture to classify a point cloud into 1) bridges, vegetation, and ground, and 2) bridge components (including wall, pier, beam, railing). DeepHyd includes two user interfaces: command line interface (CLI), and web interface. These two user interfaces are referred to as **DeepHyd-CLI** and **DeepHyd-Web**. DeepHyd-CLI, relies on the use of commands, is the core of the DeepHyd system for point cloud classification of hydraulic structures. DeepHyd-Web is based on DeepHyd-CLI and provides a web-based interactive environment for the use of DeepHyd for 3D point cloud classification.

The DeepHyd software platform is developed by Center for Applied GIScience at the University of North Carolina at Charlotte, from a project sponsored by North Carolina Department of Transportation. For any questions on the (re)use of the DeepHyd system, please contact Dr. Wenwu Tang (WenwuTang@uncc.edu). The team of software development includes: Tianyang Chen and Zachery Slocum.

The current version of DeepHyd is 1.0. DeepHyd v1.0 uses ConvPoint (https://github.com/aboulch/ConvPoint), state-of-the-art 3D deep learning platform developed from an open-source machine learning framework, Pytorch (https://pytorch.org/). This user manual focuses on the installation and use of the DeepHyd system (DeepHyd-CLI and DeepHyd-Web).

## 1. Installation of the DeepHyd system

Hardware and software requirements of the DeepHyd system (for both CLI and web interfaces):

- CPU: 2 GHz dual core processor or better.
- (Optional) NVIDIA CUDA-enabled GPU. The Compute Capacity of the GPU should not be less than 3.5; Tesla K40/GTX 730 or above, please check https://developer.nvidia.com/cuda-gpus.
- Memory: 32GB

- Operating System: Ubuntu Linux 18.04 (or higher) 64-bit
- Disk space: at least 20 GB (Solid-State-Drive storage is preferred)

### 1.1 Installation of Command Line Interface of the DeepHyd system: DeepHyd-CLI

1.1.1. Download the CLI Version of the DeepHyd system.

- Download CLI-based DeepHyd repository from github (you need to have access privilege to the github repository, which is routinely updated):
https://github.com/UNCC-CAGIS/deephyd-cli

- Or download from the NextCloud at Center for Applied GIScience (if you don't have access to DeepHyd github repository):
https://cybergis.uncc.edu/nextcloud/index.php/s/xSBLxoEj3aojR2F

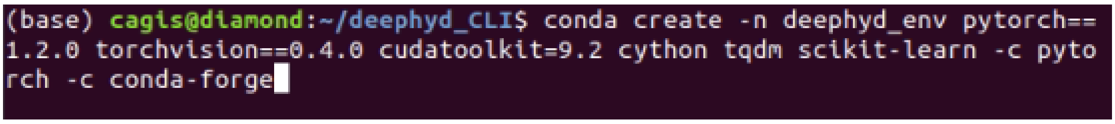
See Appendix A for the suite of software/libraries used by DeepHyd-CLI.
Once DeepHyd-CLI is downloaded, uncompress it to a directory (say, "deephyd_CLI") you prefer to store DeepHyd-CLI.

1.1.2. Setup Software Environment

We provided two approaches to set up the software environment for the DeepHyd-CLI. You can choose either one.

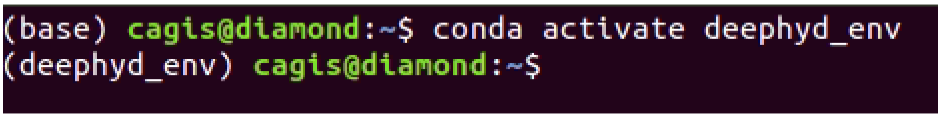
1. The first approach is straightforward for users with experience using Anaconda[1]. Anaconda is an open-source software to manage and deploy Python and R packages for scientific computing. You can build the environment with the provided command once you have Anaconda installed[2].

   Command to build the environment:
   $ conda create -n deephyd_env pytorch==1.2.0 torchvision==0.4.0 cudatoolkit=9.2 cython tqdm scikit-learn -c pytorch -c conda-forge



```
(base) cagis@diamond:~/deephyd_CLI$ conda create -n deephyd_env pytorch==
1.2.0 torchvision==0.4.0 cudatoolkit=9.2 cython tqdm scikit-learn -c pyto
rch -c conda-forge
```

   Command to activate the environment before using DeepHyd-CLI:
   $ conda activate deephyd_env



```
(base) cagis@diamond:~$ conda activate deephyd_env
(deephyd_env) cagis@diamond:~$
```

2. The second setup approach: if you are not familiar with Anaconda, you will have to download a compressed file of the built software environment (~1.2GB). You need to unpack, and activate it to run the DeepHyd-CLI.

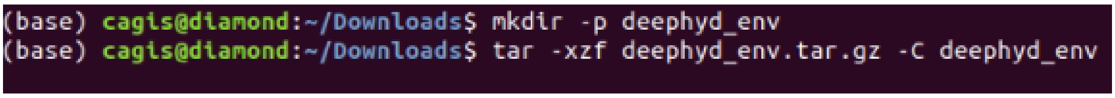
[1] See more about Anaconda at https://www.anaconda.com/

[2] See installation of Anaconda at https://docs.anaconda.com/anaconda/install/index.html

First, download the compressed file of the environment from the following link: https://cybergis.uncc.edu/nextcloud/index.php/s/RTY9XEyimbyxAqf

Second, uncompress the environment into a directory you prefer to store the environment. We just do this under the Downloads folder for a demonstration purpose. The following commands will create a folder named 'deephyd_env' under the folder and uncompress the environment into this folder.

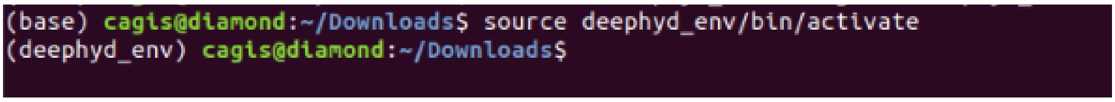
```
$ mkdir -p deephyd_env
$ tar -xzf deephyd_env.tar.gz -C deephyd_env
```



```
(base) cagis@diamond:~/Downloads$ mkdir -p deephyd_env
(base) cagis@diamond:~/Downloads$ tar -xzf deephyd_env.tar.gz -C deephyd_env
```

Third, activate the DeepHyd environment.

```
$ source deephyd_env/bin/activate
```



```
(base) cagis@diamond:~/Downloads$ source deephyd_env/bin/activate
(deephyd_env) cagis@diamond:~/Downloads$
```

Note:

1. If you use Ubuntu 18.04 64-bit Operating System as suggested, you will not need further compiling of the software libraries. However, if not or you run into an error saying *no module named convpoint.knn.lib.python.nearest_neighbors* when using CLI-based DeepHyd, you would need to compile the software libraries (for how to compile, see Appendix B).
2. If you meet a *CUDA error: no kernel image is available for execution on the device*. That means your GPU is not compatible with Pytorch. You should have a NVIDIA GPU with compute capability >= 3.0 as per the requirement of Pytorch 1.2.0[3] released in 2019. This requirement does not mention an upper limit and it might be out of date in the future; however, as of now, it still works well to use a GPU with compute capability = 7.5 as per the testing on Quadro RTX 5000. Therefore, you would need a version[4] of Pytorch compatible with your GPU if the compute capability[5] of the GPU was out of this range 3.0-7.5.

### 1.2. Installation of Web-based Interface of the DeepHyd System: DeepHyd-Web

The installation of the web-based interface for DeepHyd includes the following three steps (See Appendix A for the suite of software or libraries used by DeepHyd-Web):

**First, install Docker**

Docker is a constantly evolving technology for containerization of applications, therefore we refer you to the vendor's installation instructions here: https://docs.docker.com/get-docker/

**Second, create an SSH key for GitHub**

[3] See Pytorch 1.2.0 at https://pytorch.org/docs/1.2.0/torch.html?highlight=compute%20capability#

[4] See documentation of Pytorch versions at https://pytorch.org/docs/versions.html

[5] Check the GPU compute capability at https://developer.nvidia.com/cuda-gpus

https://docs.github.com/en/get-started/quickstart/set-up-git#next-steps-authenticating-with-github-from-git

**Third, clone the repository to /opt/deephyd/**

Note: This repository is not public at this time. For access, please use the contact information at the top of this document.

- cd /opt
- sudo mkdir deephyd && sudo chmod 777 deephyd && cd deephyd
- git clone --recurse-submodules git@github.com:UNCC-CAGIS/deephyd-deploy.git .

**Fourth, build the Docker containers**

- cd containers
- bash build-all.sh

Note: This process may take many minutes, depending on your internet connection and storage speed. With a 500mbps connection and a HDD, building took approximately 20 minutes.

**Fifth, configure DeepHyd-Web**

- cd /opt/deephyd/web
- cp sample.env .env
- Edit the new .env file in your preferred text editor
  - URL_SUBDIR: The path after the DNS name for accessing DeepHyd Web, e.g. /dhweb for example.com/dhweb
  - WEBPROJECTS_DIR: This is where DeepHyd Web will store all data such as user uploads, classified data, etc.
  - WEBLOGS_DIR, NOTEBOOKS_DIR, CLI_DIR: Modify these if you are installing in an uncommon environment.
  - ROOT_DIR: The directory where this repository has been cloned.
- chmod 777 -R /opt/deephyd/WEB-PROJECTS

**Sixth, start DeepHyd-Web**

- cd /opt/deephyd/web
- docker-compose up

If there are no apparent error messages, press CTRL-C to bring DeepHyd Web down again, and restart in the background:

- docker-compose up -d

**Seventh, setup reverse proxy (If required)**

DeepHyd-Web runs on port 5000 by default. For example, once installed, your URL to DeepHyd-Web is http://internal.example.com:5000/dhweb

It is suggested that a reverse proxy be used to access the web interface. Below, we have included an example configuration for Apache2, a common web server. For more information, consult the Apache2 documentation at https://httpd.apache.org/docs/2.4/mod/mod_proxy.html.

- Apache2 example configuration:
  - ProxyPass /dhweb http://internal.example.com:5000/dhweb
  - ProxyPassReverse /dhweb http://internal.example.com:5000/dhweb

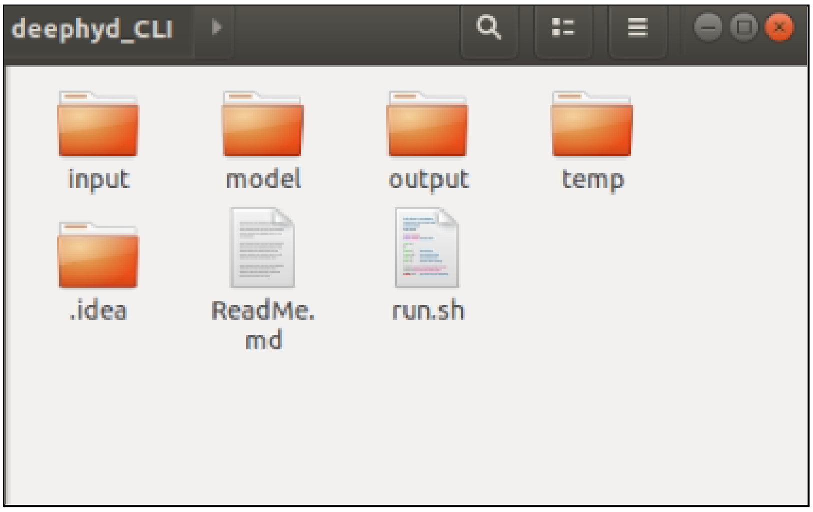
## 2. User Manual of the DeepHyd System

### 2.1 Using Command Line Interface (CLI) of the DeepHyd System

#### 2.1.1. Model repository structure

DeepHyd-CLI uses the following model repository structure. That is, if you have DeepHyd-CLI installed, within the "deephyd_CLI" folder, it has the following directories and files:

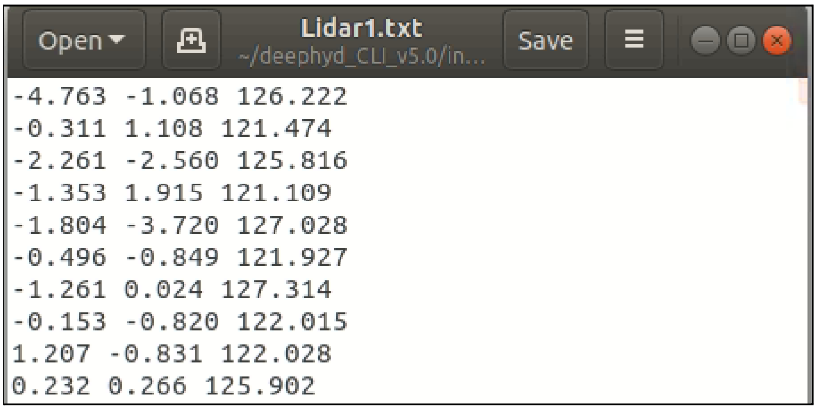
./input: For point cloud file(s) to be predicted and a configuration file
./model: For the python-based workflows and the two trained deep neural networks.
./output: For output files.
./temp/: For intermediate files.
./run.sh: A bash script to run DeepHyd system as per the configuration file.
./ReadMe.md: readme file (open it first!).



#### 2.1.2. Input requirements

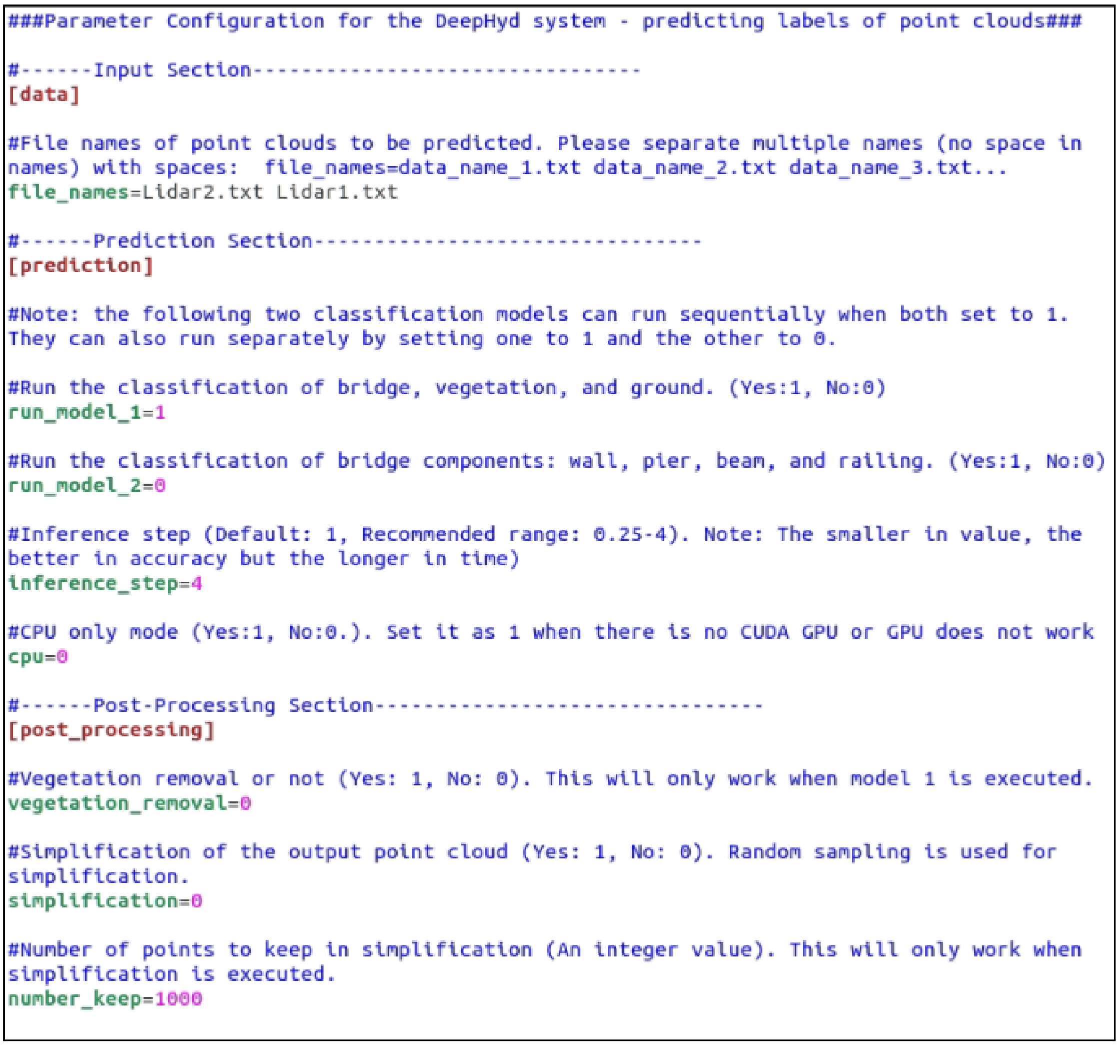
To classify point clouds, DeepHyd-CLI needs two input files to be deposited into the "input" folder: 1) point cloud that needs to be classified; 2) configuration file for parameters used by DeepHyd-CLI.

- **1) Point cloud:** Point cloud should be ASCII format in a text file. The first three columns must be the coordinates: x,y,z. The values are separated by spaces and there should not be any headers (see an example below).



```
-4.763 -1.068 126.222
-0.311 1.108 121.474
-2.261 -2.560 125.816
-1.353 1.915 121.109
-1.804 -3.720 127.028
-0.496 -0.849 121.927
-1.261 0.024 127.314
-0.153 -0.820 122.015
1.207 -0.831 122.028
0.232 0.266 125.902
```

- **2) Configuration file:** Please follow the instructions in the configuration file to configure the CLI-based DeepHyd. You just need to make changes in this file without making a new one (see configuration.ini below).

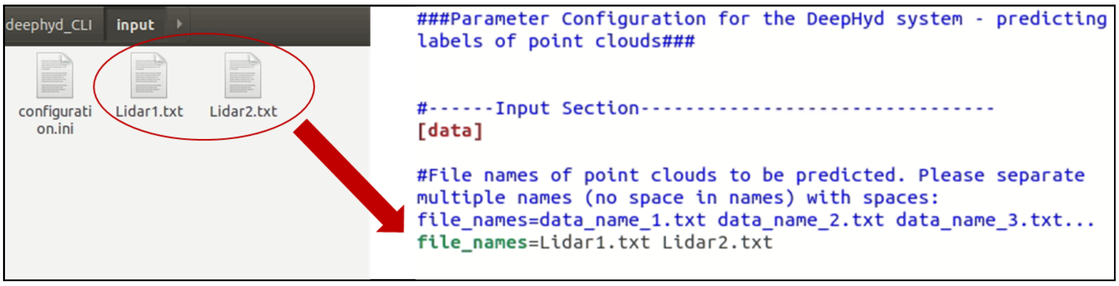


```
###Parameter Configuration for the DeepHyd system - predicting labels of point clouds###

#------Input Section------------------------------
[data]

#File names of point clouds to be predicted. Please separate multiple names (no space in
names) with spaces:  file_names=data_name_1.txt data_name_2.txt data_name_3.txt...
file_names=Lidar2.txt Lidar1.txt

#------Prediction Section------------------------------
[prediction]

#Note: the following two classification models can run sequentially when both set to 1.
They can also run separately by setting one to 1 and the other to 0.

#Run the classification of bridge, vegetation, and ground. (Yes:1, No:0)
run_model_1=1

#Run the classification of bridge components: wall, pier, beam, and railing. (Yes:1, No:0)
run_model_2=0

#Inference step (Default: 1, Recommended range: 0.25-4). Note: The smaller in value, the
better in accuracy but the longer in time)
inference_step=4

#CPU only mode (Yes:1, No:0.). Set it as 1 when there is no CUDA GPU or GPU does not work
cpu=0

#------Post-Processing Section--------------------------------
[post_processing]

#Vegetation removal or not (Yes: 1, No: 0). This will only work when model 1 is executed.
vegetation_removal=0

#Simplification of the output point cloud (Yes: 1, No: 0). Random sampling is used for
simplification.
simplification=0

#Number of points to keep in simplification (An integer value). This will only work when
simplification is executed.
number_keep=1000
```

### 2.1.3. Run DeepHyd-CLI

To run DeeHyd_CLI for point cloud classification, please follow the following steps:

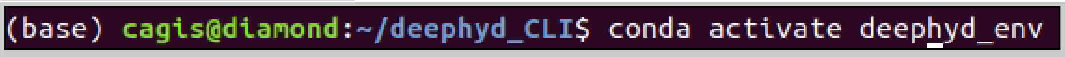
- First, put point cloud files into the input folder and revise the configuration file correspondingly. Use a space to separate point cloud files to be labeled. An example is shown below if you need to label more than one point cloud files.



```
###Parameter Configuration for the DeepHyd system - predicting
labels of point clouds###


#------Input Section------------------------------
[data]

#File names of point clouds to be predicted. Please separate
multiple names (no space in names) with spaces:
file_names=data_name_1.txt data_name_2.txt data_name_3.txt...
file_names=Lidar1.txt Lidar2.txt
```

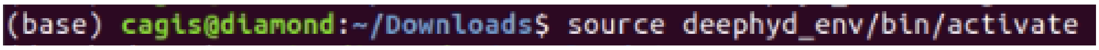
- Second, activate the software environment.

a. If you use Anaconda to build deephyd_env, you can use the command
   $ conda activate deephyd_env



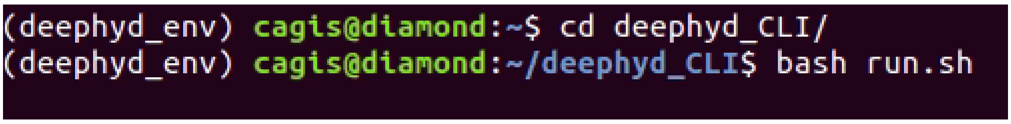
```
(base) cagis@diamond:~/deephyd_CLI$ conda activate deephyd_env
```

b. If you use the built environment without Anaconda, you should use
   $ source deephyd_env/bin/activate



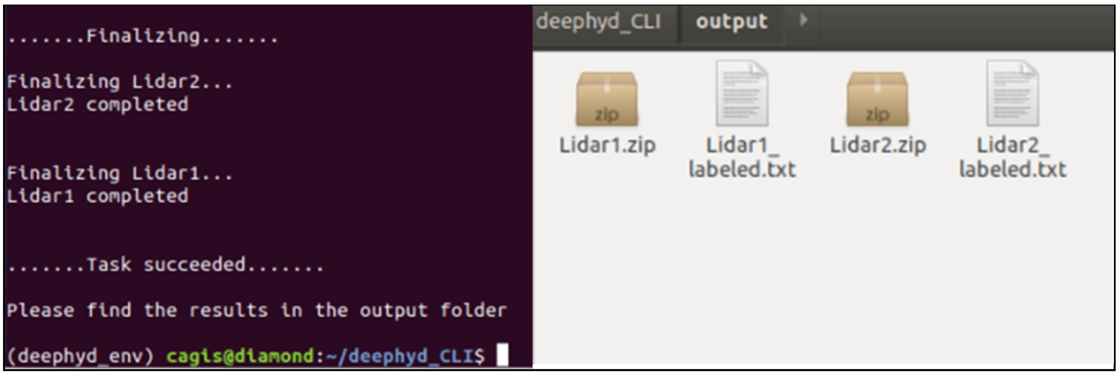
```
(base) cagis@diamond:~/Downloads$ source deephyd_env/bin/activate
```

- Third, go to the folder of DeepHyd-CLI (the directory may differ depending on where you put the DeepHyd-CLI). Execute run.sh under the directory.
  $ cd deephyd_CLI/
  $ bash run.sh



```
(deephyd_env) cagis@diamond:~$ cd deephyd_CLI/
(deephyd_env) cagis@diamond:~/deephyd_CLI$ bash run.sh
```

- Last, if DeepHyd successfully completes the tasks, it will tell you to find the results in the "output" folder. You can then obtain the point cloud with predicted labels in ASCII text files.

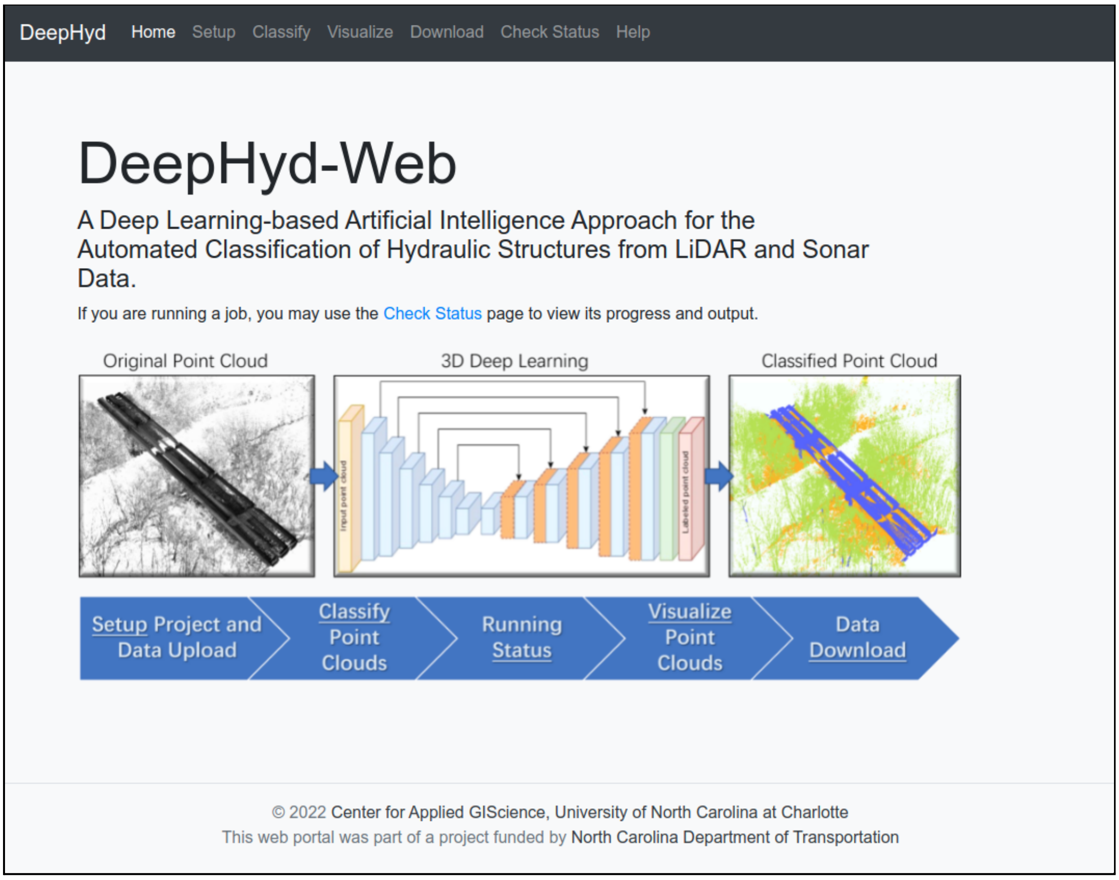


```
.......Finalizing.......

Finalizing Lidar2...
Lidar2 completed


Finalizing Lidar1...
Lidar1 completed


.......Task succeeded.......

Please find the results in the output folder

(deephyd_env) cagis@diamond:~/deephyd_CLI$
```

See Appendix C for labels for corresponding classes (e.g., bridge, vegetation, etc.)

### 2.2 Using Web-based Interface of the DeepHyd System

The web-based interface of the DeepHyd system, DeepHyd-Web, provides a user-friendly graphic interface for using DeepHyd for point cloud classification. Once you have DeepHyd-Web installed on your server infrastructure, the web portal of DeepHyd is available for conducting point cloud classification within web-based environments. The following Figure shows the home page of the DeepHyd-Web.



DeepHyd-Web

A Deep Learning-based Artificial Intelligence Approach for the Automated Classification of Hydraulic Structures from LiDAR and Sonar Data.

If you are running a job, you may use the Check Status page to view its progress and output.

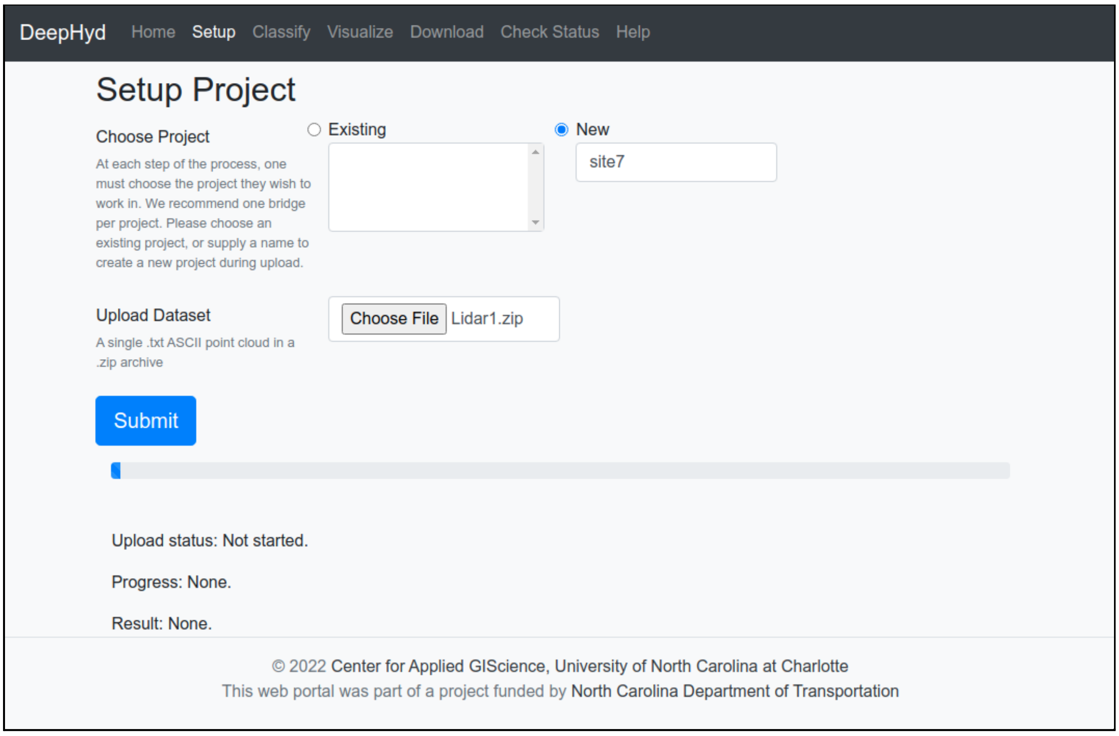
© 2022 Center for Applied GIScience, University of North Carolina at Charlotte
This web portal was part of a project funded by North Carolina Department of Transportation

The main steps of using web-based interface for point cloud classification include the following five steps:

1) Setup project and upload point cloud data
2) Classify point cloud data using DeepHyd
3) Check running status of classification jobs
4) Visualize classified point cloud
5) Download classified point clouds and data management

In this manual, we use a sample point cloud dataset "lidar1.txt" compressed in a zip file "Lidar1.zip" to create a new project "site 7" to show how to use the web-based interface by using the following specific steps.

### 2.2.1 Setup Project and Upload Data



Setup Project

Choose Project

At each step of the process, one must choose the project they wish to work in. We recommend one bridge per project. Please choose an existing project, or supply a name to create a new project during upload.

○ Existing

● New: site7

Upload Dataset

A single .txt ASCII point cloud in a .zip archive

Choose File: Lidar1.zip

Upload status: Not started.

Progress: None.

Result: None.

© 2022 Center for Applied GIScience, University of North Carolina at Charlotte
This web portal was part of a project funded by North Carolina Department of Transportation

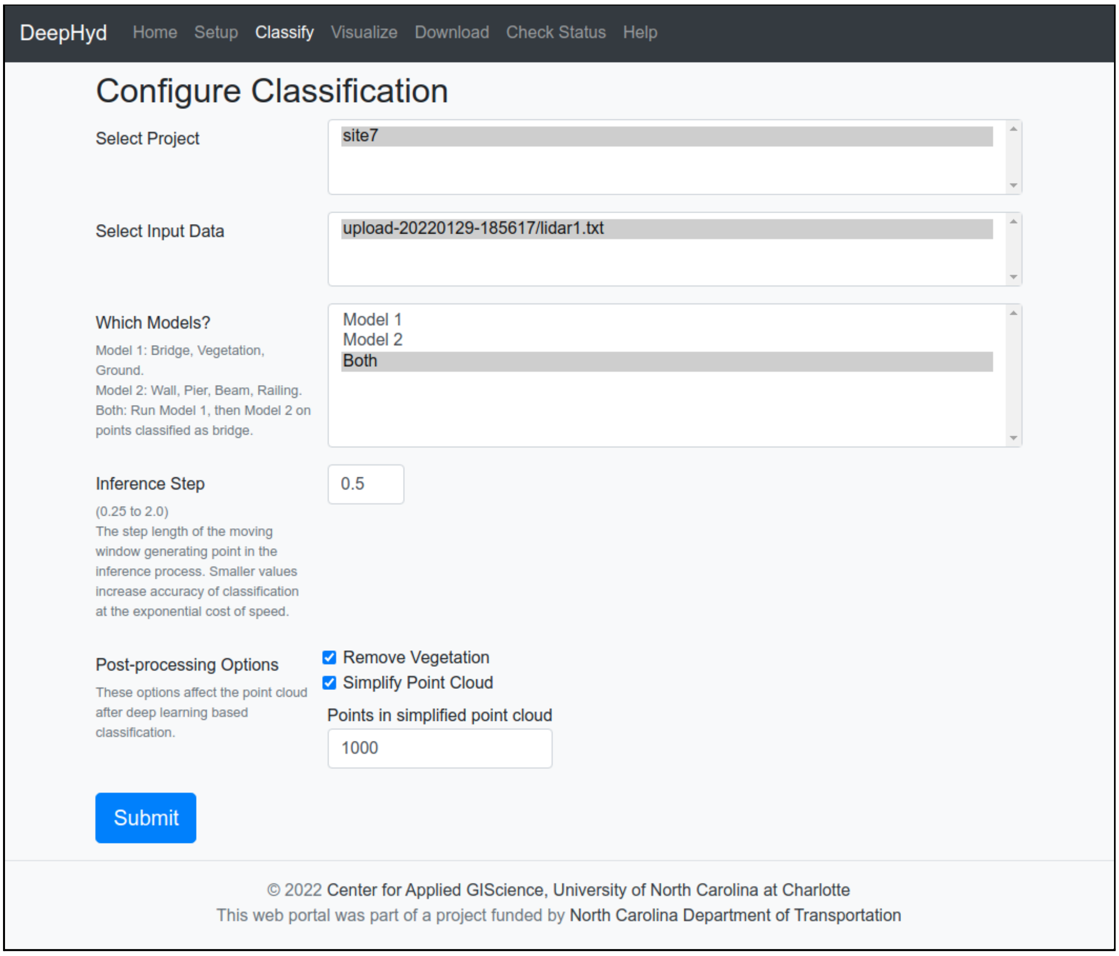
<u>Summary</u>

Upload LiDAR data in an ASCII text format to the DeepHyd system.

<u>Usage</u>

- You can start from uploading the data collected to the DeepHyd system for further processing. It currently supports LiDAR point cloud in an ASCII text format.
- By clicking the **Upload** tab on the menu bar, you will see the above page.
- **Choose Project** lets you create a new project or retrieve existing projects from the pool.
- **Upload dataset** is used to select the files you are going to upload from your local directory. It is suggested that a point cloud ASCII text file be compressed in a zip file.
- You can then click the **Submit** button to start uploading.

### 2.2.2 Classify Point Cloud Using DeepHyd



Configure Classification

Select Project: site7

Select Input Data: upload-20220129-185617/lidar1.txt

Which Models?
Model 1: Bridge, Vegetation, Ground.
Model 2: Wall, Pier, Beam, Railing.
Both: Run Model 1, then Model 2 on points classified as bridge.

Model 1
Model 2
Both

Inference Step: 0.5
(0.25 to 2.0)
The step length of the moving window generating point in the inference process. Smaller values increase accuracy of classification at the exponential cost of speed.

Post-processing Options
These options affect the point cloud after deep learning based classification.

☑ Remove Vegetation
☑ Simplify Point Cloud

Points in simplified point cloud: 1000

© 2022 Center for Applied GIScience, University of North Carolina at Charlotte
This web portal was part of a project funded by North Carolina Department of Transportation

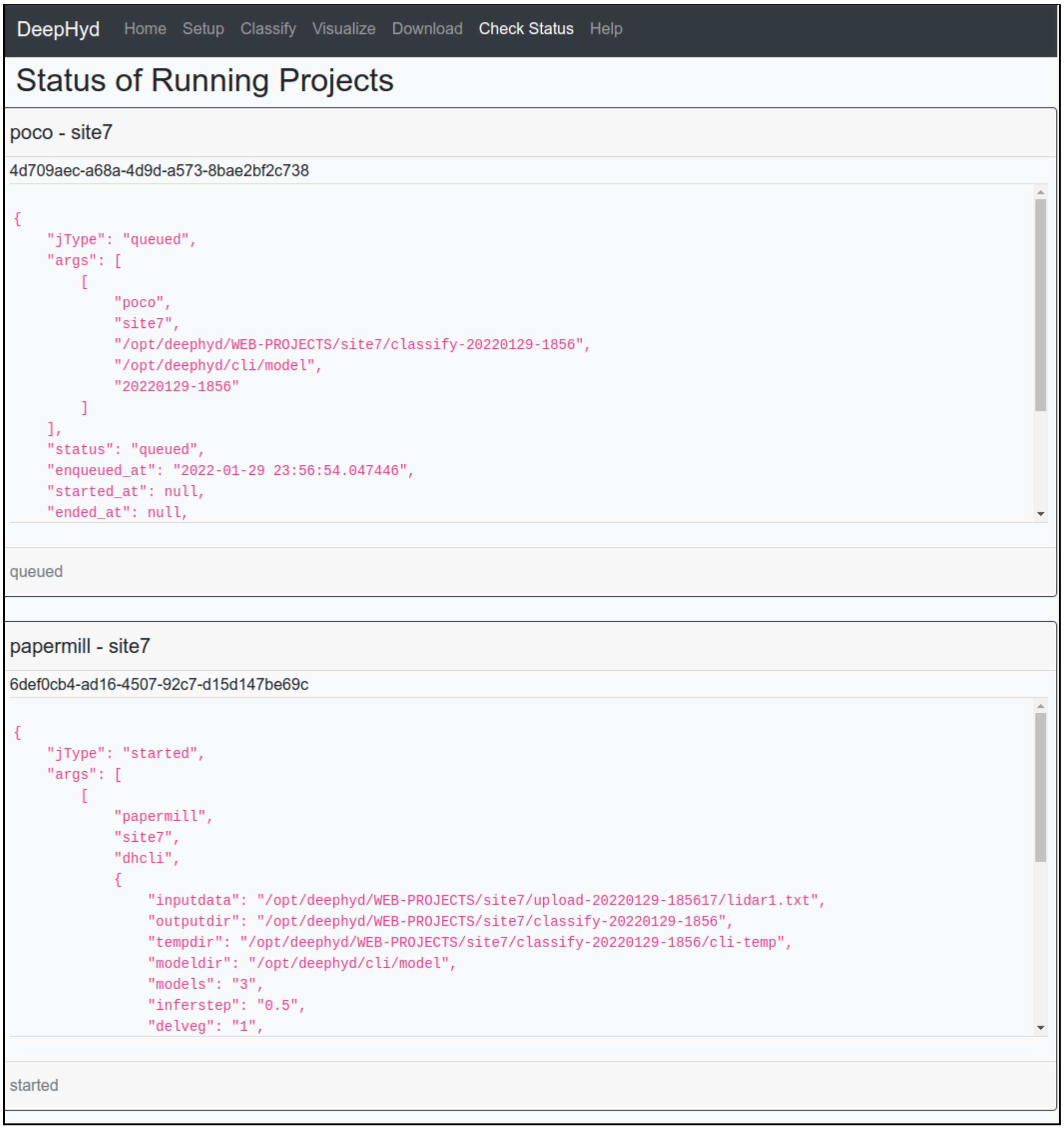
Summary

Inferencing is used to label the pre-processed data by a trained deep learning model for the classification of point clouds of hydraulic structures from LiDAR.

Usage

- After the data is uploaded, click the **Inference** button on the menu bar and the page will appear as shown.
- First, use **Select Project** to find the project for which you will conduct inferencing.
- **Select Input Data** is used to let you choose the pre-processed data to be inferenced.
- **Inference Step** affects the classification accuracy by changing the size of the step window used during classification. Higher values are faster, but less accurate.
- **Post-processing Options** allows you to customize the output point cloud. Remove Vegetation deletes vegetation-classified points from the output. Simplify Point Cloud uses the following value, Points in simplified point cloud, to randomly down-sample the point cloud to a set number of points.
- Finally, click the **Submit** button to start inference.

### 2.2.3 Check Running Status of Classification Jobs



Status of Running Projects

poco - site7

4d709aec-a68a-4d9d-a573-8bae2bf2c738

```
{
    "jType": "queued",
    "args": [
        [
            "poco",
            "site7",
            "/opt/deephyd/WEB-PROJECTS/site7/classify-20220129-1856",
            "/opt/deephyd/cli/model",
            "20220129-1856"
        ]
    ],
    "status": "queued",
    "enqueued_at": "2022-01-29 23:56:54.047446",
    "started_at": null,
    "ended_at": null,
```

queued

papermill - site7

6def0cb4-ad16-4507-92c7-d15d147be69c

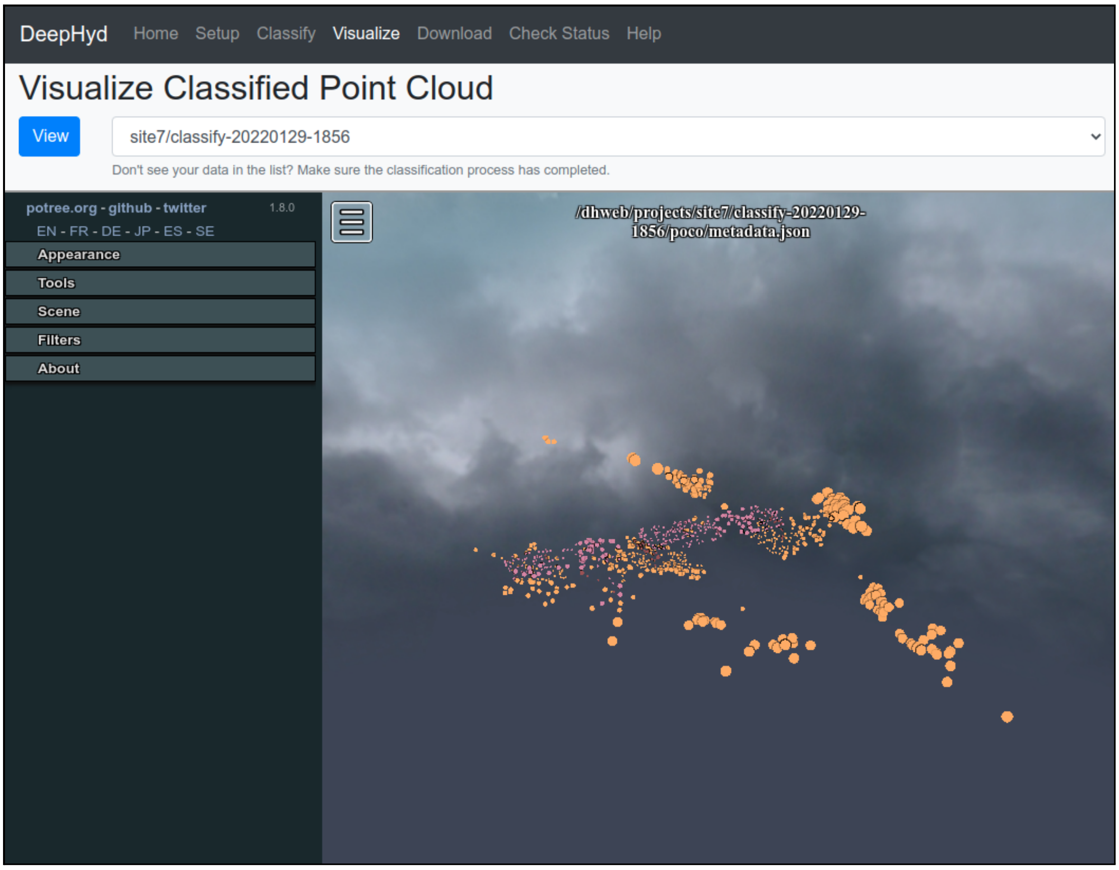
```
{
    "jType": "started",
    "args": [
        [
            "papermill",
            "site7",
            "dhcli",
            {
                "inputdata": "/opt/deephyd/WEB-PROJECTS/site7/upload-20220129-185617/lidar1.txt",
                "outputdir": "/opt/deephyd/WEB-PROJECTS/site7/classify-20220129-1856",
                "tempdir": "/opt/deephyd/WEB-PROJECTS/site7/classify-20220129-1856/cli-temp",
                "modeldir": "/opt/deephyd/cli/model",
                "models": "3",
                "inferstep": "0.5",
                "delveg": "1",
```

started

**<u>Summary</u>**

This page is used to monitor the pending, running, and completed classification jobs. If there are jobs running, it will give you status information like the Figure above. Otherwise, it will tell you no jobs are running.

### 2.2.4 Visualize Classified Point Clouds


DeepHyd
Home
Setup
Classify
Visualize
Download
Check Status
Help
View
potree.org - github - twitter
1.8.0
EN - FR - DE - JP - ES - SE
Appearance
Tools
Scene
Filters
About
/dhweb/projects/site7/classify-20220129-1856/poco/metadata.json

Visualize Classified Point Cloud

site7/classify-20220129-1856

Don't see your data in the list? Make sure the classification process has completed.

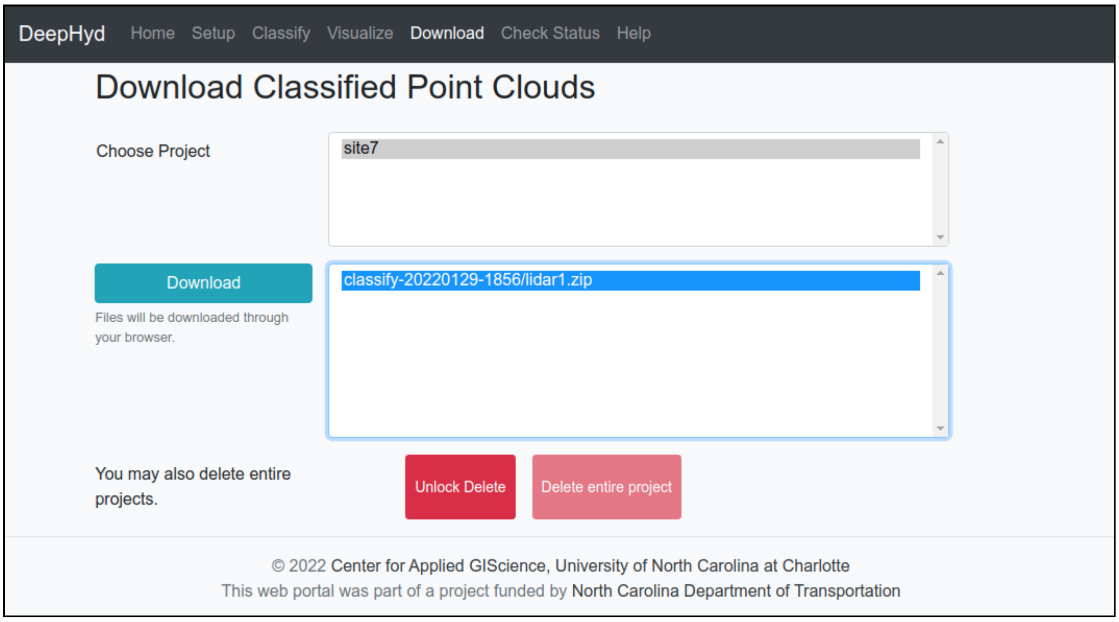
Summary

This web page is to visualize the classified point cloud. Please make sure your point cloud classification is complete by checking the running status of your job.

Usage

In the dropdown menu, select the classified point cloud for visualization. The visualization interface is based on Potree (https://github.com/potree/potree). In the left panel of the interface, Potree has many sections to explore. You may change the appearance of the point cloud, take measurements, or isolate specific areas of interest. Potree is a constantly evolving software for visualizing large point clouds. Therefore, we refer you to the project's documentation:
https://github.com/potree/potree/blob/9c98347b19a3626193f02cd4d9497ce63cba73ed/docs/user_interface.md
https://archive.fosdem.org/2015/schedule/event/potree/attachments/slides/655/export/events/attachments/potree/slides/655/potree_rendering_large_point_clouds_in_web_browsers.pdf

### 2.2.5 Data Download



Download Classified Point Clouds

Choose Project: site7

classify-20220129-1856/lidar1.zip

Files will be downloaded through your browser.

You may also delete entire projects.

© 2022 Center for Applied GIScience, University of North Carolina at Charlotte
This web portal was part of a project funded by North Carolina Department of Transportation

Summary

The download page is used to download classified point cloud results, delete unwanted projects, and to delete unwanted zip files. Please make sure your point cloud classification is complete by checking the running status of your job.

Usage

- Files may be downloaded using the **Download** button after selecting a project and a desired file.
- To **delete** a project, **select** the project in the select-box and click **Unlock Delete** button, then **Delete** entire project button.

## Appendix A

Software dependencies within the DeepHyd-CLI environment:

- Anaconda (optional)
- Python 3.7.7
- Pytorch 1.2.0
- Scikit-learn 0.23.1
- TQDM 4.48.0
- torchvision 0.4.0
- cudatoolkit 9.2
- cython

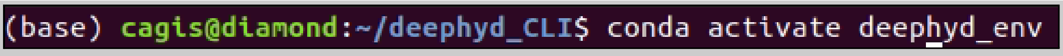
Software libraries contained in the DeepHyd-Web environment:

- All of DeepHyd-CLI’s dependencies
- open3d==0.10.0
- seaborn
- jupyter_contrib_nbextensions
- papermill

## Appendix B

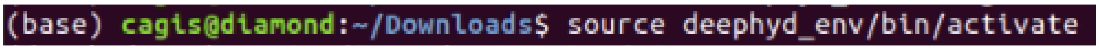
If you meet an error “*no module named convpoint.knn.lib.python.nearest_neighbors*” when using CLI-based DeepHyd, you would need to compile the library that it needs by using the following steps.

- Activate the environment.
  a. If you use Anaconda to build deephyd_env, you can use the command
  $ conda activate deephyd_env



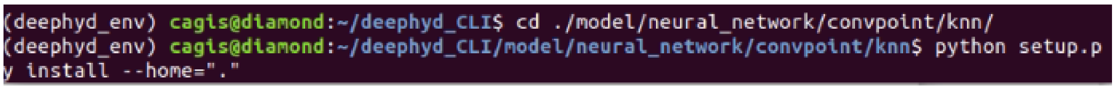
```
(base) cagis@diamond:~/deephyd_CLI$ conda activate deephyd_env
```

  b. If you use the built environment without Anaconda, you should use
  $ source deephyd_env/bin/activate



```
(base) cagis@diamond:~/Downloads$ source deephyd_env/bin/activate
```

- Go to the knn directory in CLI-based DeepHyd, and compile by using the following commands:
  $ cd ./deephyd_CLI/model/neural_network/convpoint/knn/
  $ python setup.py install –home=”.”



```
(deephyd_env) cagis@diamond:~/deephyd_CLI$ cd ./model/neural_network/convpoint/knn/
(deephyd_env) cagis@diamond:~/deephyd_CLI/model/neural_network/convpoint/knn$ python setup.p
y install --home="."
```

- The error should be solved.

**Appendix C**

List of labels for classes used by the DeepHyd system (Model 1: classification of bridges, vegetation, and ground; Model 2: classification of bridge components).

| Model 1 Classes | Model 2 Classes | Labels |
|---|---|---|
| Bridge | | 1 |
| | Wall | 11 |
| | Pier | 12 |
| | Beam | 13 |
| | Railing | 14 |
| Vegetation | | 2 |
| Ground | | 3 |