**Seismic Image Deconvolution using Deep Learning**

**Overview**

This project aims to enhance the quality of Reverse Time Migration (RTM) images in the field of seismic exploration using deep learning techniques. When seismic waves propagate and are imaged, various factors (such as limited acquisition aperture, inaccurate velocity models, etc.) can degrade the final image, resulting in low resolution, noise, and artifacts.

This project consists of two core scripts:

**train.py:** Trains a neural network end-to-end using synthetic seismic data.

**test.py:** Applies the trained model to target survey area.

**Requirements**

You will need **Python 3.8+** and the following libraries. It is recommended to use a virtual environment like conda.

PyTorch (>=1.8, CUDA version recommended for GPU acceleration)

NumPy

SciPy

OpenCV-Python

tqdm

Matplotlib

**Running the Demo**

**Please follow the steps below to train and test the model.**

**Step 1: Preparation**

1. **Clone/Download the Project: Download all files to your local machine.**
2. **Create Directories:** Manually create the required directories as shown in the "Project Structure" diagram, such as data/seam, checkpoints4train, result, and testresult.
3. **Install Dependencies:** From the project's root directory, run pip install -r requirements.txt.
4. **Prepare Data:**

Place the target RTM image model\_image\_1501x1751.dat and its corresponding PSF file model\_psf\_1501x1751.dat into the data/seam/ directory.

Place the true reflectivity model for training, ref.dat, into the data/ directory.

1. **Prepare Model & Utilities:** Ensure that the model/ and utils/ directories and their respective .py files are in place.

**Step 2: Train the Model**

This step runs the train.py script to generate training samples from data/ref.dat and train a deconvolution neural network.

1. **Open a Terminal: Navigate to the project's root directory.**
2. **Run the Training Script:**

python train.py

1. **Monitor the Process:**

You will see a progress bar in the terminal showing the training progress and the current loss value.

The script will periodically save sample test images in the result/ directory, allowing you to visualize the model's performance over time.

Upon completion, the final model weights file (e.g., Network\_10.pth) will be saved in the checkpoints4train/ directory.

**step 3: Test the Model on Target Data**

This step uses the model trained in the previous step to process the real data located in data/seam/.

1. **Check Model Path:**

Open the test.py script.

Locate the modelpath variable and ensure its value points to the model weights file you trained in Step 2. For example:

modelpath = './checkpoints4train/Network\_10.pth'

1. **Run the Test Script:**

python test.py Network\_10.pth

1. **Review the Results:**

The script will process the entire large RTM image using the sliding window method, which may take some time.

Once finished, the terminal will display "Processing complete."

The final, high-resolution seismic profile will be saved in binary format in the testresult/ directory with the filename result\_seam.dat. This file can be directly used for subsequent geophysical analysis.