Documentation for Deep Learning-based Small Scale Mining Mapping Script

Before running the script, you need to set up a Python environment named ‘deeplearn’. You can do this using Miniconda.

Miniconda is a free minimal installer for Conda. It is a small, bootstrap version of Anaconda that includes only conda, Python, and the packages they depend on. Conda is an open-source package management system and environment management system that runs on Windows, macOS, and Linux. Conda quickly installs, runs, and updates packages and their dependencies. It was created for Python programs but it can package and distribute software for any language.

How to Install Miniconda

Before you can use Miniconda, you need to install it on your computer. Here’s how:

1. **Download Miniconda**: Go to the Miniconda page in your web browser. Download the installer that matches your operating system.
2. **Run the Installer**: Locate the Miniconda installer that you downloaded. Double-click the installer file to start the installation process.
3. **Follow the Prompts**: The installer will guide you through the installation process. Accept the license agreement and choose the installation location.
4. **Complete the Installation**: The installer will install Miniconda and Conda in the location you specified. Once the installation is complete, you can close the installer.

Environment Setup

Here’s how you can create a new environment in Miniconda:

```
conda create --name deeplearn
```

Activate the environment:

```
conda activate deeplearn
```

Next, you need to install the necessary Python packages. This script requires the following packages: `numpy, tensorflow, keras, PIL, osgeo`, and custom modules like `image_utils, numpy_utils, and geotiff_utils`. You can install them using conda:

```
conda install numpy tensorflow keras pillow gdal
```

The custom modules (`image_utils, numpy_utils, geotiff_utils`) should be located in the same directory as your script or installed as packages in your Python environment.

Code Explanation

The script uses a deep learning model to predict and map small scale mining activity based on input images. Here's a breakdown of what each part of the script does:

- **Imports**: The script imports necessary modules and functions. It uses Numpy for numerical operations, TensorFlow and Keras for the deep learning model, PIL for image manipulation, and gdal for handling geospatial data formats.
- **real_image_preprocessing_func**: This function normalizes the input image by clipping pixel values at 6000 and dividing by 6000.
- **create_mask**: This function generates a mask from the model's prediction. It selects the class with the highest probability for each pixel.
- **predict_mask**: This function takes an image and the trained model as input, pre-processes the image, feeds it into the model, and returns a mask of the predicted mining activity.
- **pre_process_func**: This function applies the `predict_mask` function to each grid in the geospatial data.
- **Main Script**: The script asks for the source path of the geospatial data and the destination path for the output. It loads the trained model and applies the `pre_process_func` to each geotiff file in the source path. The predicted masks are saved in the destination path.

Please ensure that the paths for the source data and the trained model are correctly specified and that the necessary Python packages and custom modules are installed in your environment.

This script is designed to run on a machine with sufficient memory and a GPU for the deep learning model. If you encounter any issues, you may need to adjust the size of the images or the batch size to fit your machine's capabilities.

How to Run the Script

1. **Open the Miniconda Prompt**: You can do this by searching for "Miniconda Prompt" in your computer's search bar and clicking on the application.

2. **Navigate to the Script Directory**: Use the `cd` command to change the directory to where your script and utility files are stored. For example, if your files are in a folder named 'project' on your Desktop, you would type:

```
cd Desktop/project
```

3. **Activate the Environment**: Activate the 'deeplearn' environment that you created earlier:

```
conda activate deeplearn
```

4. **Run the Script**: You can run the script using the python command followed by the script name. When the script prompts for the source path, enter the path to the folder where your source images are stored. When it prompts for the destination path, enter the path to the folder where you want the output to be saved. For example, if your script is named ‘generate_mask.py’, you would type:

   ```
   python generate_mask.py
   ```

Then, when prompted, enter the source and destination paths. For example:

```
Source path: C:/Users/username/Desktop/project/source
Destination path: C:/Users/username/Desktop/project/output
```

Please replace ‘username’, ‘Desktop/project’, ‘source’, and ‘output’ with your actual paths.

Remember, paths in Windows are usually with backslashes (\), but Python interprets backslashes as escape characters, so you need to use forward slashes (/) or double backslashes (\\) in your paths.

Alternatively, you can run the script via Jupyter Notebook:

Installing Jupyter Notebook

1. **Open the Miniconda Prompt**: You can search for “Miniconda Prompt” in your computer’s search bar and click on the application.

2. **Activate the Environment**: Activate the ‘deeplearn’ environment that you created earlier:

```
conda activate deeplearn
```

3. **Install Jupyter Notebook**: You can install Jupyter Notebook using conda with the following command:

```
conda install -c anaconda jupyter
```

This will install Jupyter Notebook and its dependencies.

Running the Script in Jupyter Notebook

1. **Open Jupyter Notebook**: You can do this by typing `jupyter notebook` in your Miniconda Prompt and pressing enter. This will open a new tab in your web browser with the Jupyter Notebook interface.

2. **Navigate to the Script Directory**: In the Jupyter Notebook interface, navigate to the directory where your script and utility files are stored. You can do this by clicking on the folders.

3. **Create a New Notebook**: Once you’re in the correct directory, you can create a new notebook by clicking on ‘New’ and selecting ‘Python 3’ under ‘Notebook’.

4. **Copy the Script**: Open the ‘generate_mask.py’ file in a text editor, copy the entire script, and paste it into a cell in your new Jupyter Notebook.

5. **Run the Script**: You can run the script by clicking on ‘Cell’ in the menu and selecting ‘Run All’. When the script prompts for the source path, enter the path to the folder where your source images are stored. When it prompts the destination path, enter the path to the folder where you want the output to be saved.