CAVESEG YOLO MODEL

Overview:

The CAVESEG model uses YOLOv8 for efficient object detection with image segmentation. This guide outlines the installation process and provides instructions for running the model using a graphical user interface (GUI).

Install process:

Install Pytorch according to requirements: https://pytorch.org/get-started/locally/

The CAVESEG model relies on the Ultralytics YOLOv8 library. Install it using the following command:

```
pip install ultralytics
```

For more information you can refer to the following website:
https://docs.ultralytics.com/quickstart/

To use the GUI for image selection, you need Tkinter. Install it using the following command (for Ubuntu/Debian systems):

```
sudo apt-get install python3-tk
```

For other operating systems, refer to the specific installation instructions.

Download the YOLOv8 model and necessary scripts from this link: (insert github link)

- Before running the model, ensure all paths in the scripts are correctly set. Specifically, open `caveseg.py` and update the `model_path` variable to point to the location where you downloaded the YOLOv8 model.

Running the model:

After completing the installation and configuration, you can run the model with the following command:

```
python3 caveseg.py
```

GUI Interface:

Upon executing the command, a GUI will prompt you to select images for testing the YOLOv8 model. You can choose one or multiple images. The program will check for the existence of corresponding mask files and process the selected images through the YOLO model.

Output:

The results of the image processing will be saved in a newly created **`results`** folder, which will be generated by the **`caveseg.py`** script.

Trouble Shooting:

If you encounter issues during installation or execution:

- Ensure all dependencies are installed correctly.
- Check that your paths in **`caveseg.py`** are accurate.
- Refer to the documentation linked above for issues related to PyTorch and Ultralytics.

Setting Up the MMSegmentation Library and Using the CaveSeg Model

Step 1: Install the MMSegmentation Library

1. Visit the MMSegmentation GitHub repository and follow the installation guide provided in the "Get Started" section.
2. Ensure you adhere to the following best practices during installation:
    - Download and install up-to-date versions of all required packages and tools. Using outdated dependencies may cause errors during setup or execution.
    - Create and activate a dedicated virtual environment for your MMSegmentation setup. This ensures package compatibility and isolates the environment from conflicts with other installations.

Step 2: Troubleshooting Common Errors

- Package Conflicts:
  If you encounter errors, double-check that all installed packages are the latest compatible versions recommended in the guide. Update them as necessary.

- Resource Accessibility in Virtual Environments:
  Occasionally, packages or resources may install outside of the virtual environment. Ensure all necessary files and dependencies are accessible within the virtual environment. If required, move or re-install these resources while the virtual environment is active.

Step 3: Setting Up the CaveSeg Model

To use the CaveSeg model with the MMSegmentation library:

1. Place the CaveSeg model files and folders in the MMSegmentation directory. These files should include:
   - Configs: Contains configuration files for the model.
   - Demo: Demonstrates how the model functions.
   - Tools: Includes helper scripts and utilities.
   - Work_dirs: Directory for saving outputs and intermediate results.
   - Readme: Instructions and notes specific to CaveSeg.
   - Images: Sample images for testing.
   - Labels: Associated label data for the images.
   - test_change_label_color.py: Script for testing label color changes.
   - test_txt_prepare.py: Script for preparing .txt files for the model.
2. Within the MMSegmentation directory, ensure the following files are present and properly configured:
   - caveseg.py: This script defines and loads the CaveSeg model.
   - best.pt: This is the pre-trained CaveSeg model file.

Step 4: Testing and Using the CaveSeg Model

1. Use the standard MMSegmentation scripts to test and validate the CaveSeg model. These scripts are compatible as long as all required CaveSeg files are correctly integrated.
2. Double-check all file paths in the configuration scripts and update them as necessary to reflect the proper locations of resources within your MMSegmentation folder structure.

Additional Notes:

- Folder Structure:
  Ensure the MMSegmentation library is organized as per the instructions. The CaveSeg model and its associated resources must be placed within the correct subdirectories to ensure seamless functionality.

- Environment Setup:
  Always work within the virtual environment you created for MMSegmentation. Run all commands, upload resources, and make modifications within this environment to avoid errors.
- Best Practices:
  Regularly refer to the MMSegmentation documentation for updates or additional instructions. This ensures compatibility with the latest features and tools.

By carefully following these steps and addressing potential issues proactively, you can successfully set up the MMSegmentation library and integrate the CaveSeg model for use.

-

https://github.com/iCAS-Lab/edge-devices/blob/main/ncs2/README.md