Available noise reduction models

Please use the following models in project KAIR (integrated and updated)：https://github.com/cszn/KAIR?tab=readme-ov-file

1. DnCNN：https://github.com/cszn/DnCNN?tab=readme-ov-file
2. FFDNet：https://github.com/cszn/FFDNet?tab=readme-ov-file
3. DPIR：https://github.com/cszn/DPIR
4. IRCNN Noise reducer：https://github.com/cszn/KAIR?tab=readme-ov-file

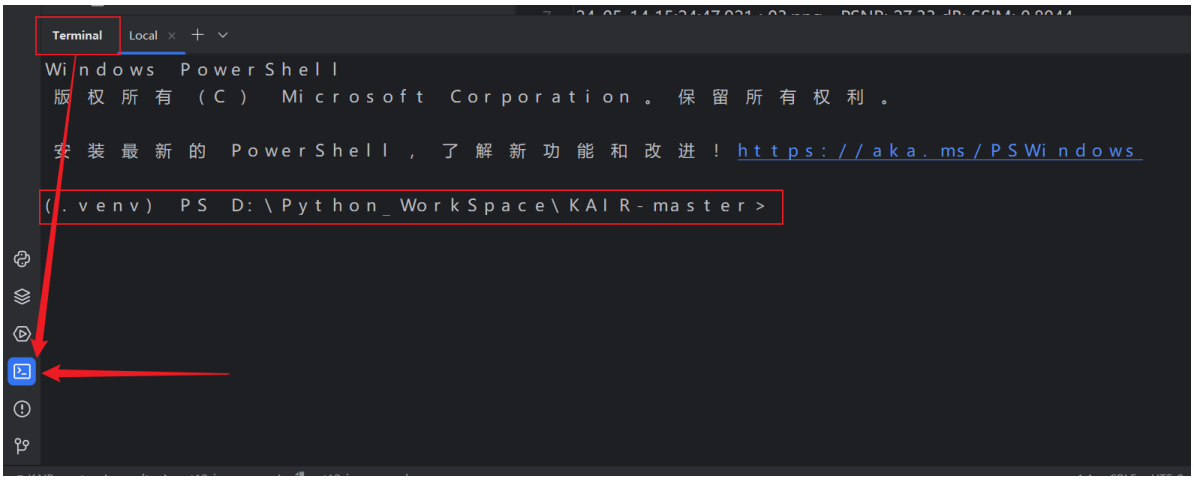
Instructions

一.DnCnn(Using Gaussian denoising)

Environment configuration, for using DnCNN, use python version 3.8.10 portable

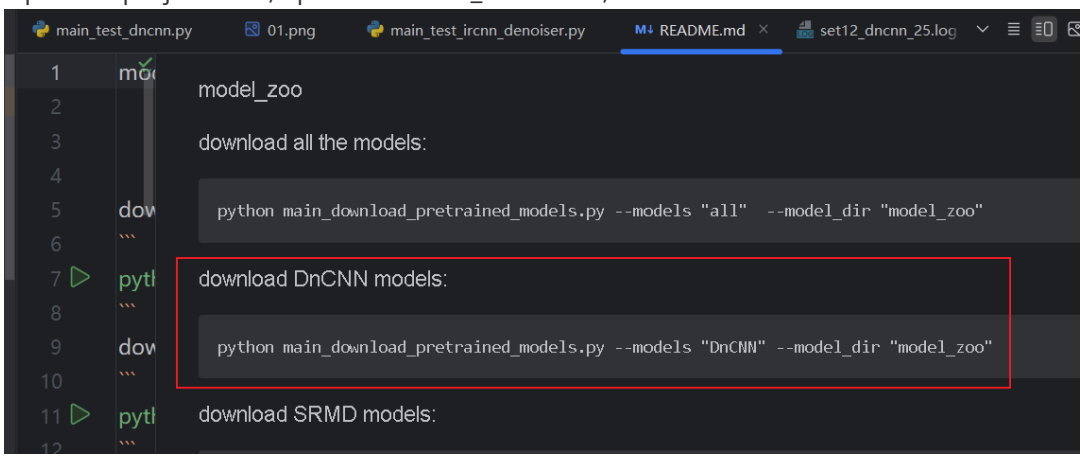
- Install dependencies (please execute in terminal). Please don't worry. The installation instructions are executed in one line. The first instruction installs multiple dependencies. After the installation is completed, wait a moment. Pycharm takes time to load.

```
pip install -r requirement.txt
pip install requests
pip install matplotlib
```



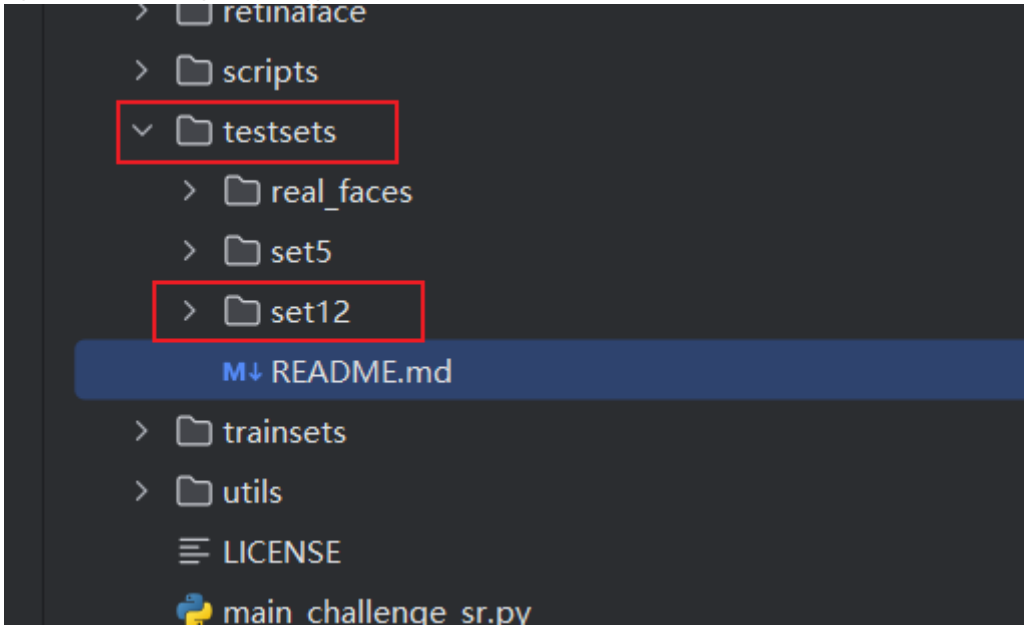
When executing in the terminal, the path will be displayed when each installation instruction is completed. If it is not displayed, it means it is loading.

1. Open the project KAIR, open the model_zoo folder, and view the documentation inside

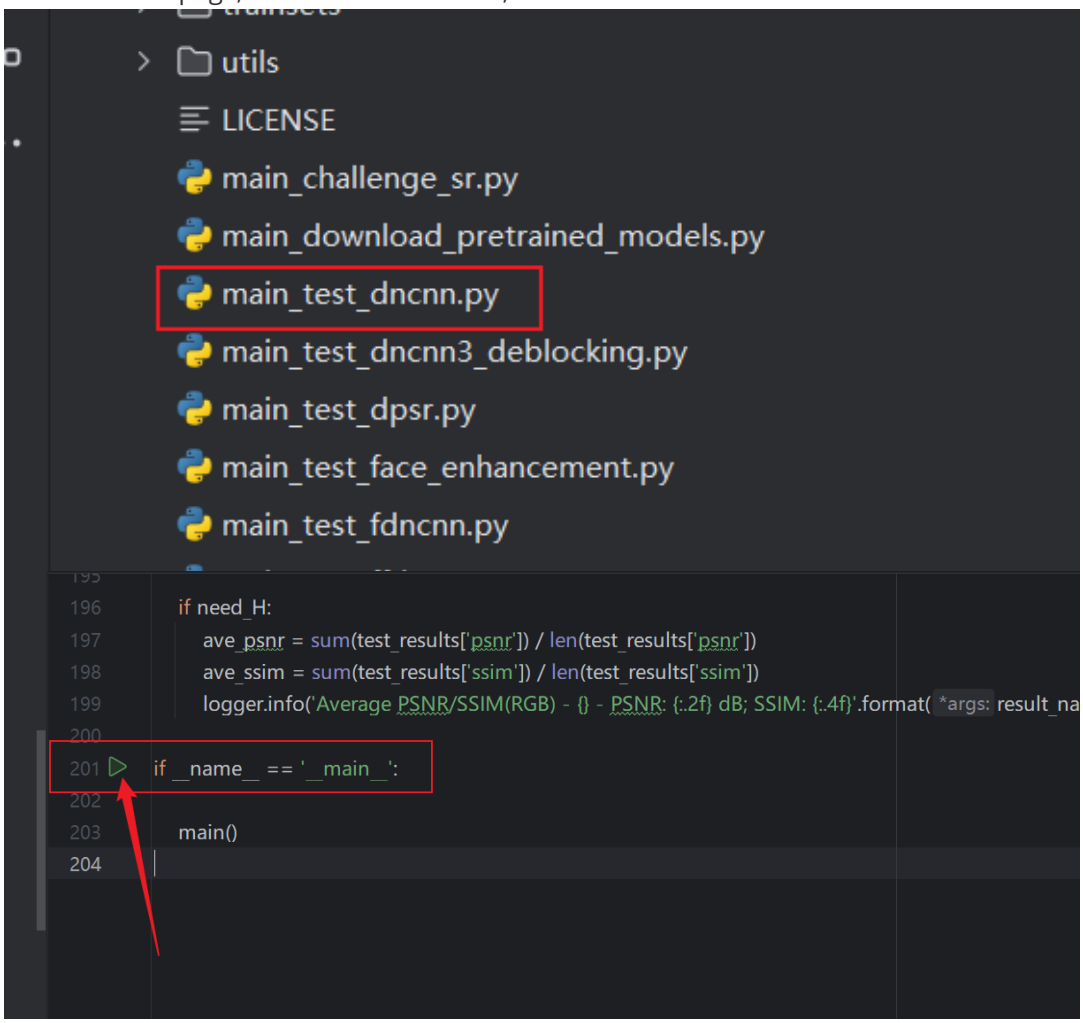


2. You can execute the code above in the terminal to download the model required by DnCNN, or download it from the download URL (model_zoo folder, the download URL is provided at the bottom of the documentation inside, and all files starting with dncnn are included). The downloaded model files are placed in the model_zoo folder

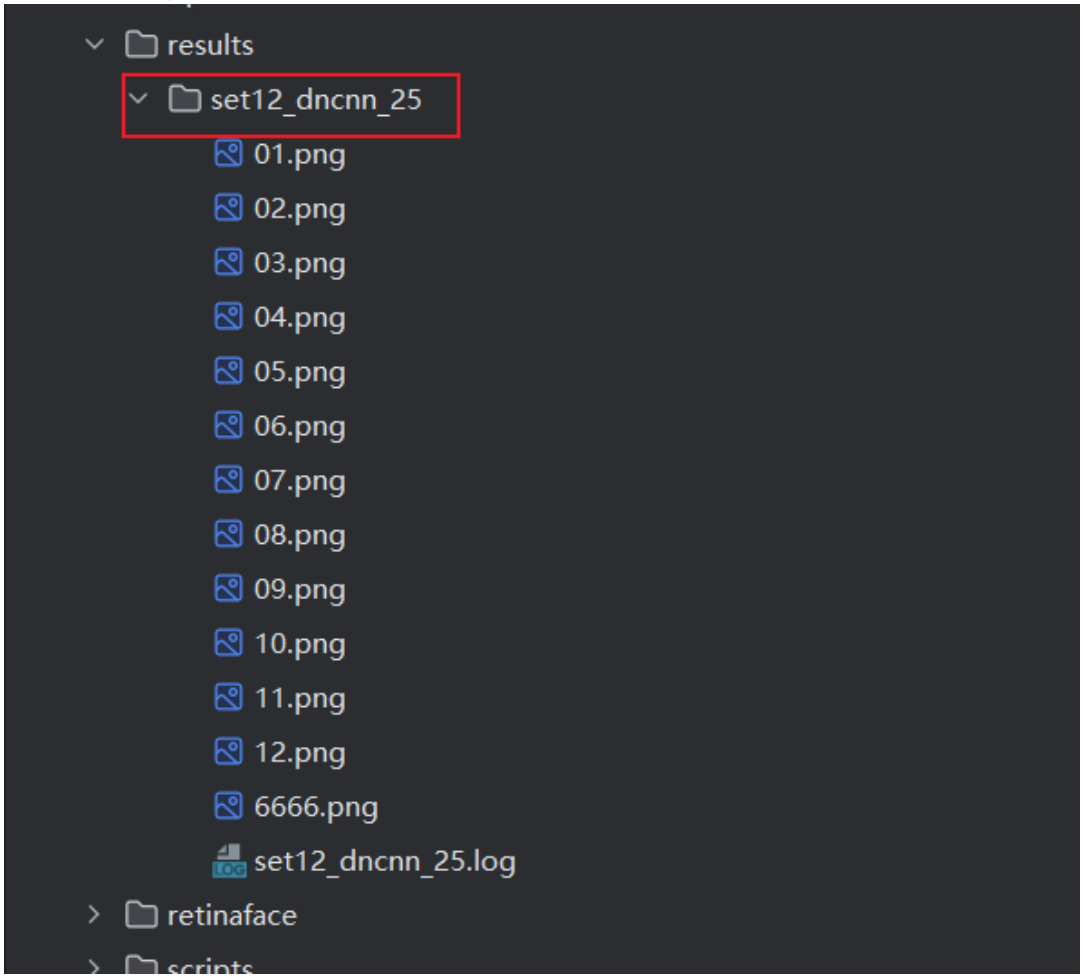
3. Set the pictures that need to be denoised: Open the folder under the KAIR project, then open set12, and place the pictures that need to be denoised inside。



4. Under the KAIR project, find main_test_dncnn.py, click on this file, scroll down to the bottom of the page, see the main method, click to run

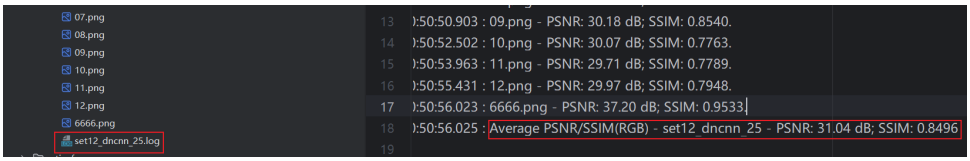


5. The denoised image results are placed in the results folder under the KAIR project. The first half of the generated folder (set12) represents the data source, and the second half (dnncn_25) represents the model used.



6. The generated log file displays key information about the image data used.。

    1. **Average PSNR/SSIM(RGB)**: This means that what is recorded next is the average PSNR and SSIM values based on the RGB image. PSNR (Peak Signal-to-Noise Ratio) and SSIM (Structural Similarity Index) are common indicators for image quality evaluation.
    2. **set12_dncnn_25**: This means that the evaluation is based on the `set12` dataset and uses the `DnCNN` algorithm with a noise level of 25 (usually meaning Gaussian noise with a standard deviation of 25). `DnCNN` is a deep convolutional neural network commonly used for image denoising tasks.
    3. **PSNR: 31.04 dB**: This is the average Peak Signal-to-Noise Ratio value. PSNR is an important indicator of image quality after denoising. The higher the value, the better the image quality. 31.04 dB indicates that the average quality of the denoised image is relatively high.
    4. **SSIM: 0.8496**: This is the average Structural Similarity Index value. SSIM is used to measure the similarity of two images in terms of structure, brightness and contrast. The value ranges from 0 to 1. The closer to 1, the more similar the image structure is. 0.8496 indicates that the structural similarity between the denoised image and the original image is high.



补充，The model file can be changed. For example, dncnn_25 is a model file trained for average noise 25. Can be replaced with 3, 15, 50. **dncnn_gray_blind_blind** removes gray image noise, **dncnn_color_blind** removes color image noise

- Enter in the terminal. This allows you to use another model dncnn_50

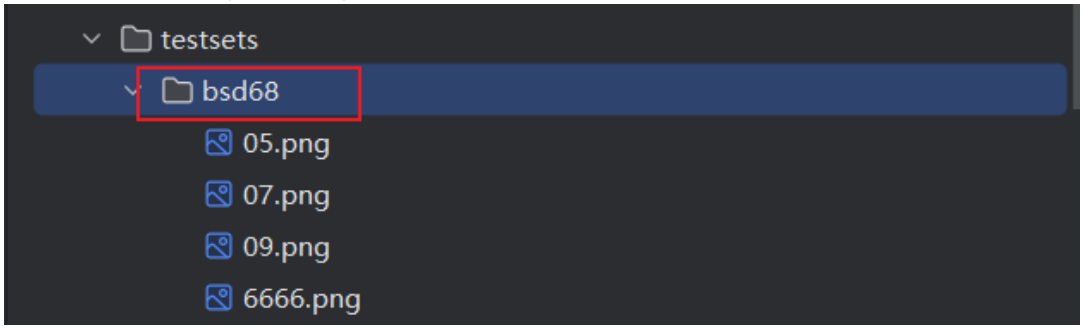
```
python main_test_dncnn.py --model_name dncnn_50
```

二.FFDET

Fast and flexible denoising convolutional neural network, providing a fast and flexible solution for CNN-based image denoising.

Environment configuration, for using FFDET, it is feasible to use python version 3.8.10.

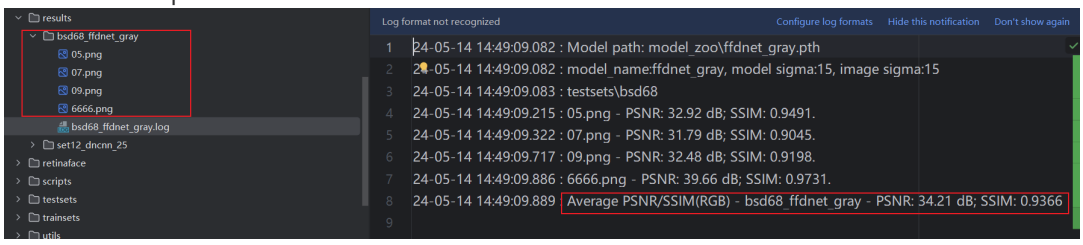
1. Same as above
2. Same as above，Just download two models ffdnet_color.pth and ffdnet_gray.pth, the former is for color and the latter is for grayscale. A single FFDET model can target the noise of [0, 75].
3. Set the pictures that need to be denoised: Open the folder under the KAIR project, create the file bsd68, and place the pictures that need to be denoised in it.



4. Under the KAIR project, find main_test_ffdnet.py, click on this file, scroll down to the bottom of the page, see the main method, click to run (grayscale is used by default), you can use the code below to switch (console execution)

```
python main_test_ffdnet.py --model_name ffdnet_color.pth
```

5. The result output is as follows



三.DPIR

Go to the project address to research and use. The other three can be used directly in KAIR.

四.IRCNN

Environment configuration, it is feasible to use python version 3.8.10.

Dependency version changes

- To lower the numpy version, first uninstall the associated dependencies, then install the 1.19.5 version of numpy, and finally install the uninstalled dependencies.

```
pip uninstall matplotlib
pip uninstall scikit-image
pip uninstall scipy
pip install numpy==1.19.5

pip install matplotlib==3.3.3
pip install scikit-image==0.17.2
pip install scipy==1.5.4
```

1. Same as above
2. Same as above，Just download two models ircnn_gray.pth and ircnn_color.pth, the former is for grayscale and the latter is for color.
3. Set the pictures that need to be denoised: Open the folder under the KAIR project, then open set12, and place the pictures that need to be denoised in it.
4. Under the KAIR project, find main_test_ircnn_denoiser.py, click on this file, scroll down to the bottom of the page, see the main method, click to run (grayscale is used by default), you can use the code below to switch (console execution)

```
python main_test_ircnn_denoiser.py --model_name ircnn_color.pth
```

5.Output results