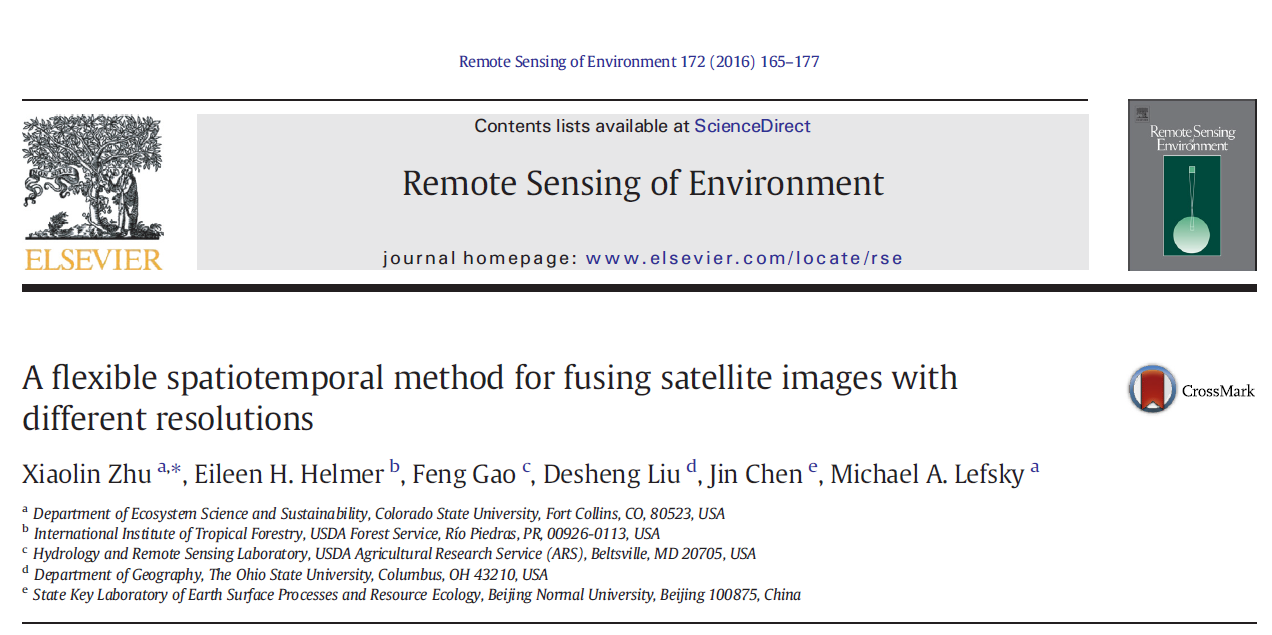
**Instruction of FSDAF Program（Python Code）**

Yue Sun ([sunyuemy@gmail.com](mailto:sunyuemy@gmail.com)), Xiaolin Zhu (zhuxiaolin55@gmail.com)

Department of Land Surveying and Geo-Informatics  
The Hong Kong Polytechnic University

**Reference**



**Download link:**

<https://www.sciencedirect.com/science/article/pii/S0034425715302042?casa_token=aa44zgANFZQAAAAA:NOTT6PyW8VFOh3CLWHW3s_F5CXi0yPHOE1KSjN3rppzspzNSspyFCl3eUSLDEyiY0Bd0sNlFVw>

**Input data preparation**

1) 1 Landsat image;

2) 1 MODIS images at the same time as Landsat image;

3) one MODIS image at the prediction time;

4)\* classification map of the Landsat image if using “FSDAF\_preclassification.pro” code

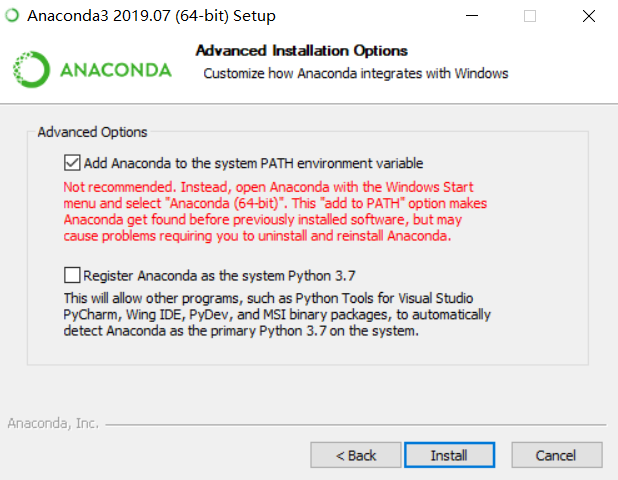
**Note: size of Landsat image should be k\*scale (e.g., scale=480/30, 480 is resampled MODIS resolution, 30 is Landsat resolution), k is an integer**

**Operation Steps of FSDAF**

**Step1: Configure Anaconda and PyCharm**

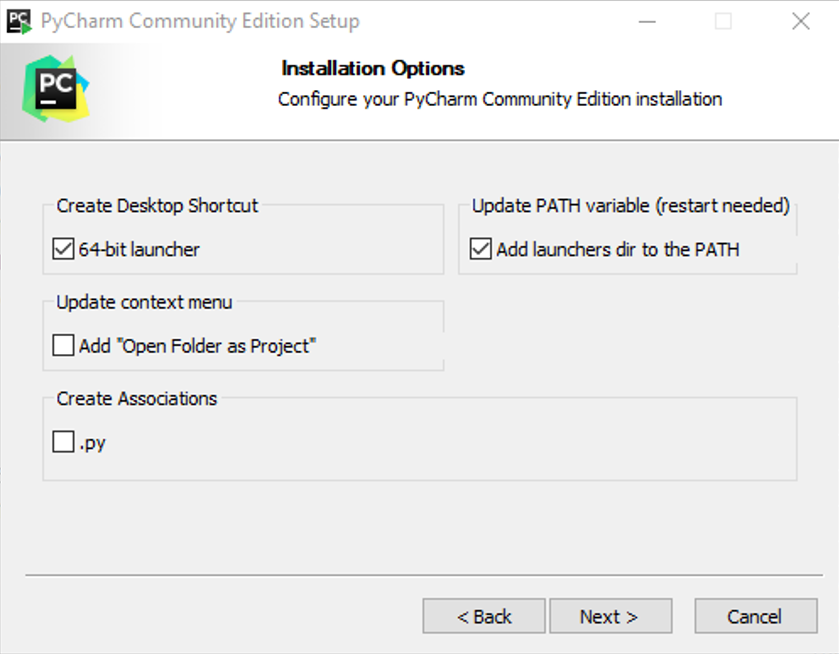
**Anaconda is selected as a virtual environment**

1. Download Anaconda from <https://www.anaconda.com/distribution/> for your operating system and install (Please check the box of “Add Anaconda to the system PATH environment variable”)



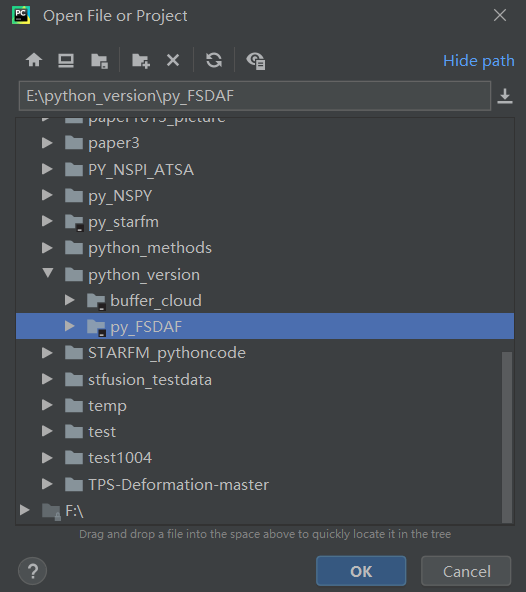
**PyCharm**

1. PyCharm Community Edition is used as an integrated development environment (IDE) for development and debugging. Download the latest version of PyCharm from <https://www.jetbrains.com/pycharm/download/#section=windows>. The free Community Edition is recommended.
2. Install PyCharm (Please check the box of “Add launchers dir to the PATH”)

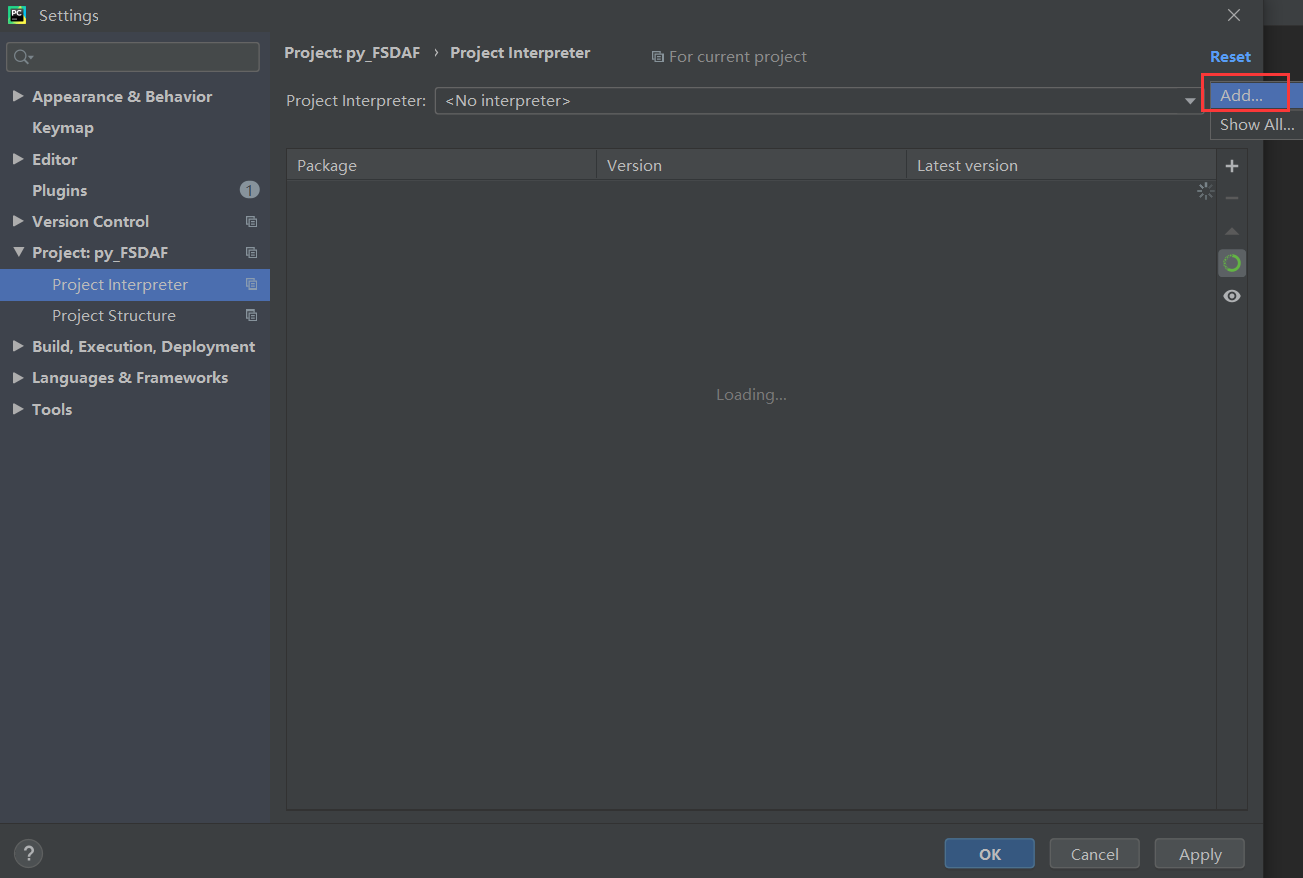


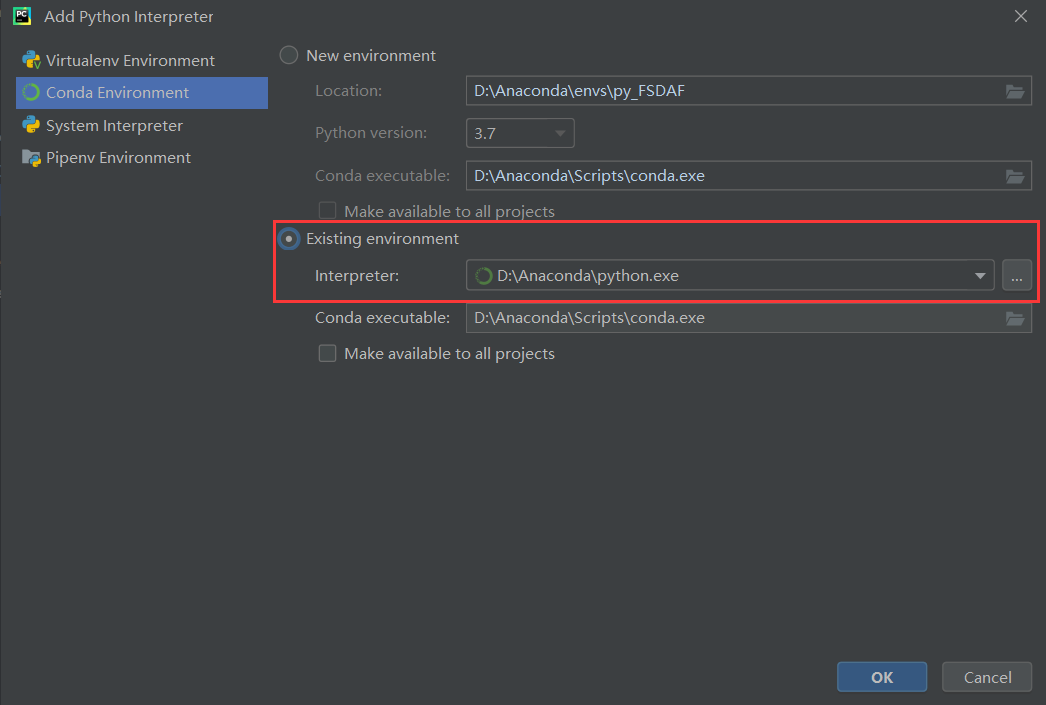
**Configure PyCharm with Anaconda**

1. Launch JetBrains PyCharm and click “File -> Open” to Open py\_FSDAF project (i.e. select the folder “py\_FSDAF” and then click “OK”)

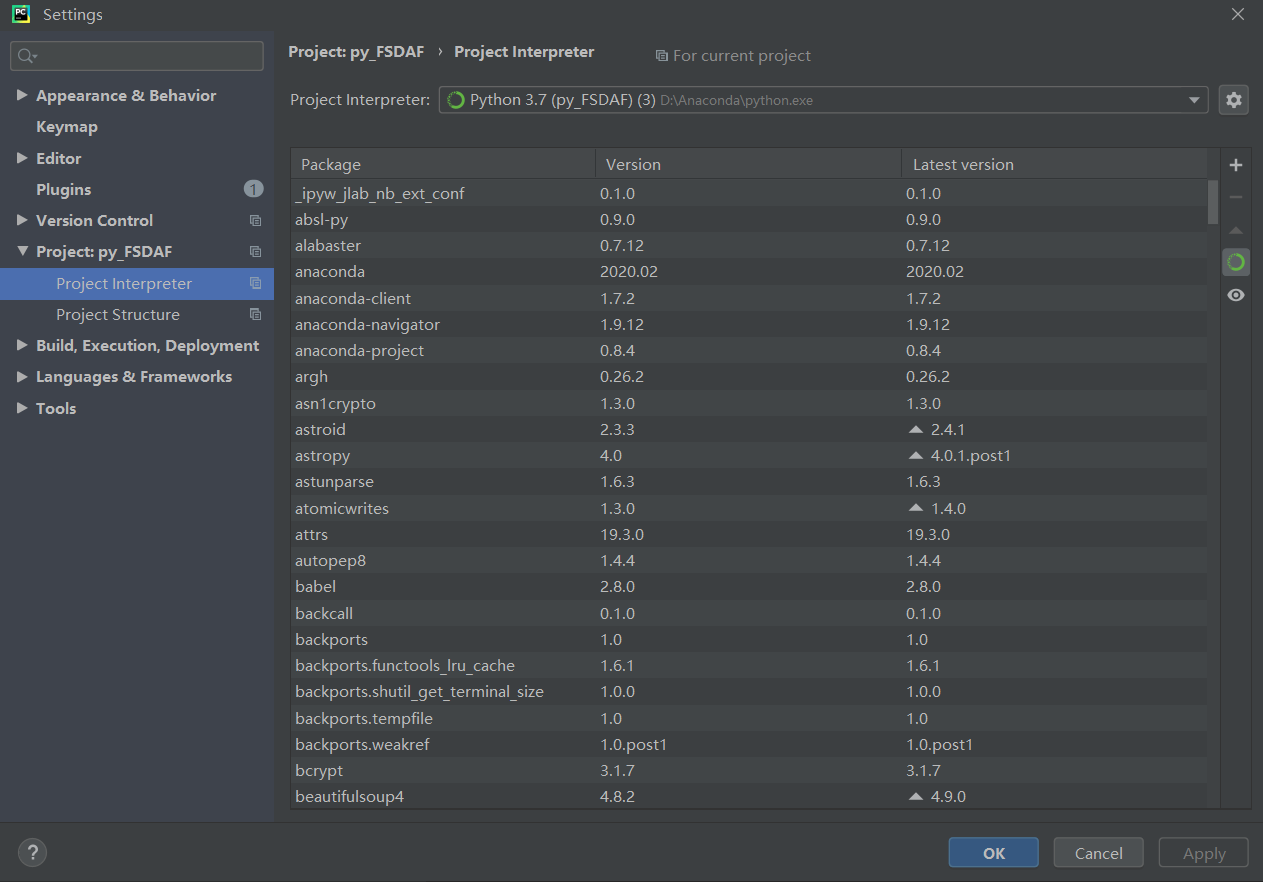


1. Select File -> Settings -> Project -> Project Interpreter
2. Click a settings button  for adding the python.exe interpreter that you just installed in Conda Environment to the “Existing environment”. In my case, the path is “D:\Anaconda\python.exe”, then click “OK”.



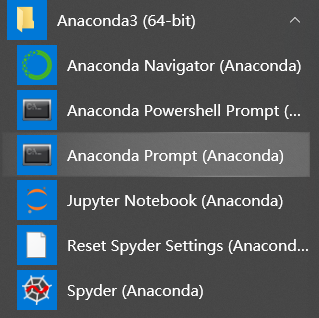


Then click “OK” in the following interface window:

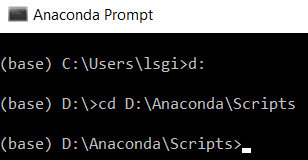


**Install the required packages**

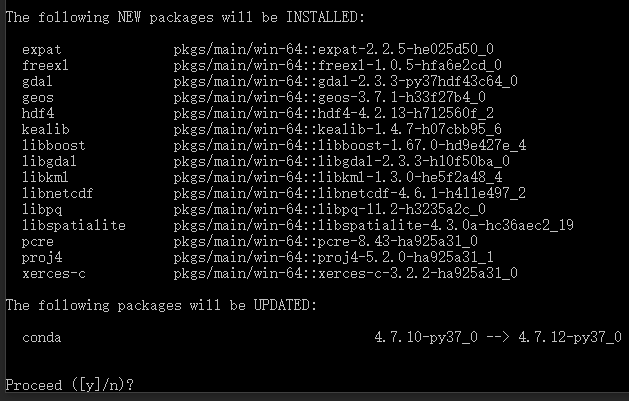
1. Open **Anaconda Prompt** in the window starter



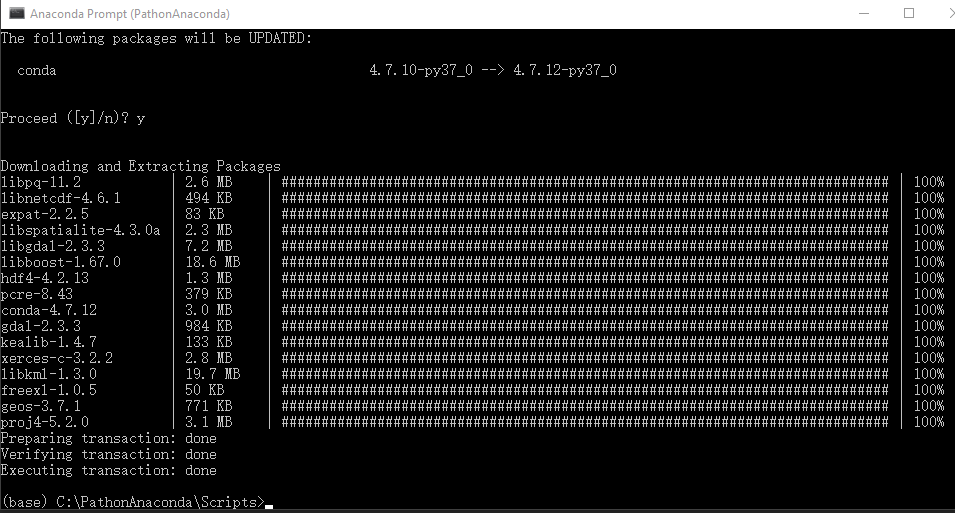
1. Locate your Python **Scripts** path (can be found within the Anaconda installation folder) and press **Enter**. In my case, the Python Scripts path is: D:\Anaconda\Scripts, so in the pop-out window, type “cd D:\Anaconda\Scripts” and the press “enter”



1. Type “**conda install gdal” and press “enter”** toinstall the required Python package **gdal** (other required packages include os, numpy, yaml, statsmodels, scipy, tkinter idlwrap, many of which have been included in Anaconda, if not, please use **pip install packagename** to install required packages). Type “y” and press “enter” when the window shows the following message:

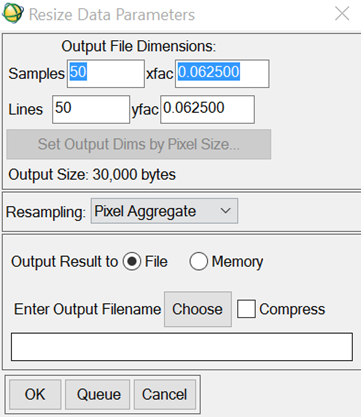


1. It is done **when you see the following text** (if no errors appear, then the package was successfully installed)



**Step 2 Data Prepocessing**

* In ENVI or MODS tool: resample all MODIS images to 480 m (250 m data to 240 m) use nearest neighbor method and reproject MODIS to Landsat’s projection;
* Resample Landsat image to the same resolution of MODIS (480 m or 240m) using pixel aggregate method



**Step 3**

* Geo-rectified one MODIS image to the resampled Landsat: Choose resampled Landsat as reference, correct MODIS.
* Envi/map-registration/ select GCPs: image to image

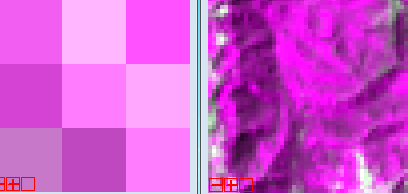
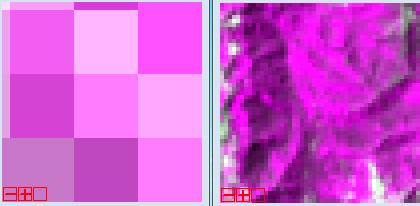
**The rule derived from GCPs applied to all MODIS images**

**Step 4**

* Crop all MODIS images by sampled Landsat, make sure they are with exactly same size:
* Envi/File/save file as envi standard/import file-select MODIS image/spatial subset/file-select Landsat/ok
* Resample MODIS to 30m by nearest neighbor method

**Note: For the test data, we have done step2-step4!**

* **Make sure all the input data with same size and all MODIS pixels are complete pixels (have 16\*16 Landsat pixels inside)**



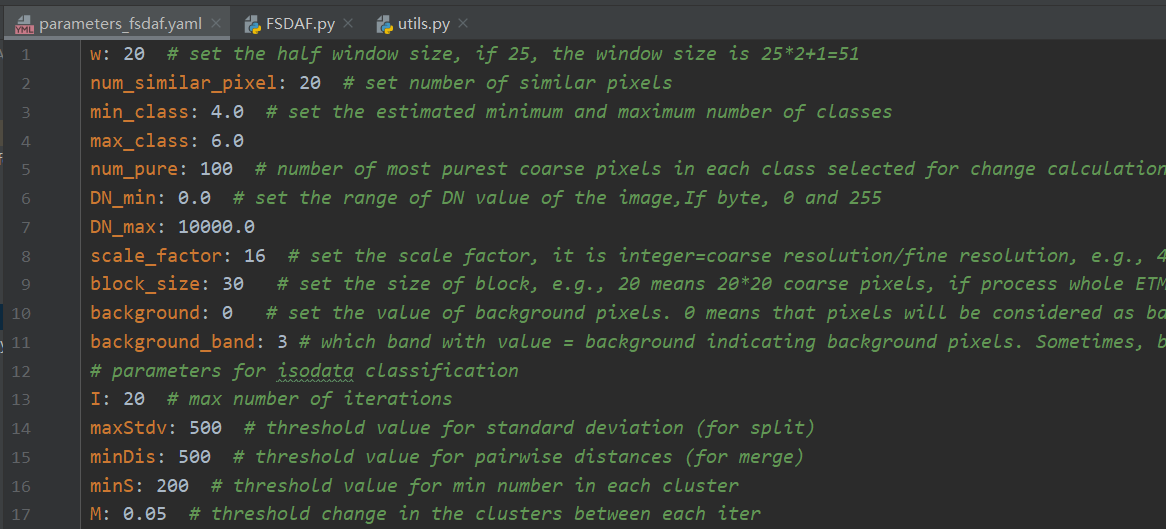
**Step 5: Set parameters and pat**

BAD!

GOOD

**Step 5 Set the parameters**

1. Please follow the explanation or the paper to set the parameters, which can be set in **parameters\_fsdaf.yaml (and parameters\_fsdaf\_preclassification.yaml)**. The default parameters are used for test data, you can change these parameters according to your own data before using this parameter file.



**NOTE: if the input data is with only one band, please set the parameter background\_band to 1.**

1. Since the ISODATA function used here is not the same as that in ENVI, as for the parameters for ISODATA classification, if you cannot acquire appropriate classification results, please:

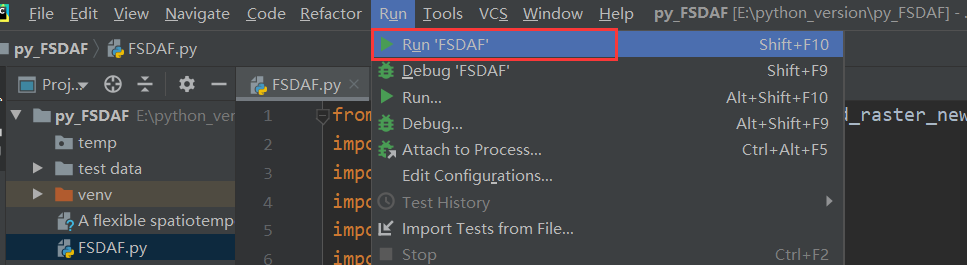
* Adjust the input classification parameters into a proper range (for example, you can decrease or increase the value of maxStdx and minDis)
* Or classify input image using ENVI or other ways to get preclassification results and using FSDAF\_preclassification.py to do spatio-temporal fusion work

**Step 6: Run**

**Note:**

* **FSDAF: use isodata to do classification automatically. There is no need to input a classification map**
* **FSDAF\_preclassification: users need to input a classification map from the Landsat image (for the test data, “Landsat20041126\_SVM\_classification” is the classification map)**

Click Run -> Run ‘FSDAF’



Open files according to the name of pop-up windows

* Open the parameter settings file (**parameters\_fsdaf.yaml or parameters\_fsdaf\_preclassification.yaml if using FSDAF\_Preclassification**)
* Set the temporary folder
* open the fine image of the first pair
* open the coarse image of the first pair
* open the coarse image of the prediction time
* open the class image of fine image in the 1st pair (if using **FSDAF\_Preclassification**)

Output: a fused image of the original MODIS file name with a suffix “\_FSDAF”

Thanks for using FSDAF algorithm. If you meet any problems, please feel free to contact Miss Yue Sun or Dr. Xiaolin Zhu.