PyRS: A Python package to process remotely sensed data for geomatics education

Xichen Meng[1], Yilong Han[1,2,*], Hanchao Zhang[3], Chaowei Huang[4], Wenhuan Yang[5], Chuanbin Liu[1], Jiaying Song[1], Zhanquan Liu[1]

1.College of Geodesy and Geomatics, Shandong University of Science and Technology, Qingdao, Shandong 266590, China;

2. Shandong University of Science and Technology Shandong Provincial 3S Engineering Technology Research Center, Qingdao;

3. Chinese Academy of Surveying & Mapping, Beijing 100036, China;

4.Shanghai Institute of Satellite Engineering, Shanghai 201109, China;

5.China Siwei Surveying and Mapping Technology Co.Ltd., Beijing 100036, China

KEY WORDS: PyRS, Remote Sensing, Image Processing, Data Fusion, Classification.

ABSTRACT:

PyRS is a Python package developed for processing remotely sensed data. It provides a user-friendly interface to handle various remote sensing tasks, including data reading, radiometric correction, image enhancement, image denoising, image segmentation, land cover classification, change detection, construction of remote sensing indices, quantitative inversion. PyRS includes a wide range of state-of-the-art algorithms and tools for processing optical data. PyRS is an open-source project that is continually updated and maintained by a group of developers. It is designed to be modular and extensible, allowing users to add new algorithms and functionality to the package. PyRS is widely used in the remote sensing community and has been applied to various applications, such as land cover mapping, crop monitoring, and disaster response. PyRS is a comprehensive and powerful Python package for processing remotely sensed data. Its user-friendly interface, broad range of functionalities, and open-source nature make it an attractive tool for researchers and practitioners in the remote sensing field. PyRS encapsulates the current mainstream remote sensing algorithms, which can greatly improve the scientific research efficiency of researchers in the field of remote sensing. The form of Python packages and the characteristics of open source make it more flexible and transparent than professional remote sensing processing software. PyRS will contribute to the progress and development of geomatics education.

1. INTRODUCTION

Earth resource monitoring is a vital topic in geoscience research because human living environment is significantly affected by the changes in Earth's biosphere biophysical characteristics. Artificial Earth satellites provide a stable and continuous data source for Earth observation by acquiring a large amount of Earth observation data (EO) with the advancement of space technology. Remote sensing technology is an extension of computer vision in geoscience field, which can apply many computer vision algorithms. New technologies in computer vision field promote remote sensing algorithms development. For instance, deep residual network (ResNet) (He et al., 2016) and ViT (Vison Transformer) (Dosovitskiy et al., 2020) have been effective in processing large-scale remote sensing images. Computer vision develops rapidly for many reasons, one important factor is various open source scientific computing, computer vision, machine learning, deep learning processing libraries. For example, Numpy (Harris et al., 2020), OpenCV, PIL, Scikit-image, Scikit-learn, Pytorch (Paszke et al., 2019) and TensorFlow. These libraries offer mature and highly optimized algorithms that make computer vision research easier and data processing more convenient and efficient. However unlike computer vision related libraries there are not many options for remote sensing image processing related libraries. Only a few libraries such as GDAL and PaddleRS exist which do not include common remote sensing image processing algorithms. Although various commercial remote sensing software are available but remote sensing researchers still need a remote sensing image processing library that has high-performance computing capabilities and contains remote sensing image processing algorithms to meet their deeper and wider needs.

Commercial remote sensing processing software, such as ENVI, ERDAS and PIE-Basic, incorporate mainstream remote sensing image processing algorithms internally, which can meet basic needs. However, as research progresses, using software alone may not satisfy the scientific research requirements. In addition, commercial remote sensing software is closed-source software, and users may not be able to view the internal algorithm implementation process, which belongs to a "black box" operation: for researchers, its workflow is opaque, non-reproducible and uncontrollable; for remote sensing professional students, they cannot understand the underlying principles of the algorithm. Moreover, other existing geospatial data processing libraries have limitations. GDAL focuses on the most basic geospatial data processing functions such as reading various types of raster data and vector data type conversion geometric correction etc. PaddleRS library is still in development stage and its core function is deep learning application in remote sensing. These two libraries do not involve high-level remote sensing image processing functions such as atmospheric correction image enhancement image denoising land cover classification etc. In contrast Google Earth Engine (GEE) is an online geospatial data processing platform developed by Google that has diverse functionalities. However, the use of cloud platform leads to its lack of flexibility as an algorithm library.

This paper presents a novel remote sensing image processing library in Python language, which builds upon various well-established libraries for scientific computing, machine learning, and geospatial data processing. It implements advanced functionalities of remote sensing image processing and wraps them as APIs. Its fully open source characteristic enables a highly transparent and reproducible workflow. PyRS will greatly improve the scientific efficiency of those involved in remote sensing and advance geomatics education.

2. PYTHON-BASED SYSTEM SCIENTIFIC COMPUTATION

Python is a widely-used object-oriented interpreted language that boasts strong portability, a low learning threshold, and natural-language-like syntax with pseudo-code characteristics. Python's scientific computing ecosystem is particularly robust and includes libraries such as Numpy, Scikit-learn, OpenCV, PyTorch, TensorFlow, and GDAL. Numpy is a library that provides a multidimensional array object and functions for processing arrays, including high-performance array operations, Fourier transforms, graphics operations, and linear algebra operations. Scikit-learn offers machine learning algorithms and tools for efficient data mining and analysis, making it a valuable resource for complex environments. OpenCV efficiently implements computer vision algorithms for various applications, including image stitching, denoising, quality inspection, and face recognition. The library has significantly contributed to the development of computer vision, and it offers a Python interface for use with Numpy. PyTorch and TensorFlow are the two most popular deep learning frameworks in Python and provide users with easy access to common operators, automatic differentiation, and back-propagation functions while also supporting GPU parallel computing. GDAL is a library for processing geospatial data, providing a single raster abstract data model and vector abstract data model for all supported formats. Its Python interface makes it a valuable tool for remote sensing image processing.

Python's extensive scientific computing ecosystem provides crucial support for remote sensing image processing libraries, ensuring a more mature infrastructure and continuous stable service for users.

3. PYRS IMAGE CLASS

PyRS has developed the Image class as a data type for storing raster data. The Image class is based on the GDAL dataset class and extends its functionality, thereby enabling the reading of various types of raster data, including GeoTIFF, NetCDF, HDF, and ENVI. The GDAL dataset class comprises a collection of raster bands and associated information, including arrays of raster imagery, raster descriptors, the number of bands, raster width, raster height, spatial reference information, projection information, among other data. The Image class uses the GDAL dataset class as one of its attributes, thereby allowing users to access all of the functionality of the GDAL dataset class through the Image class. The Image class also reads raster descriptors, the number of bands, raster width, raster height, spatial reference information, projection information, and stores them as attributes of the class, enabling users to access the relevant information with greater ease. Fig 1 shows the data structure and functions of the Image class.

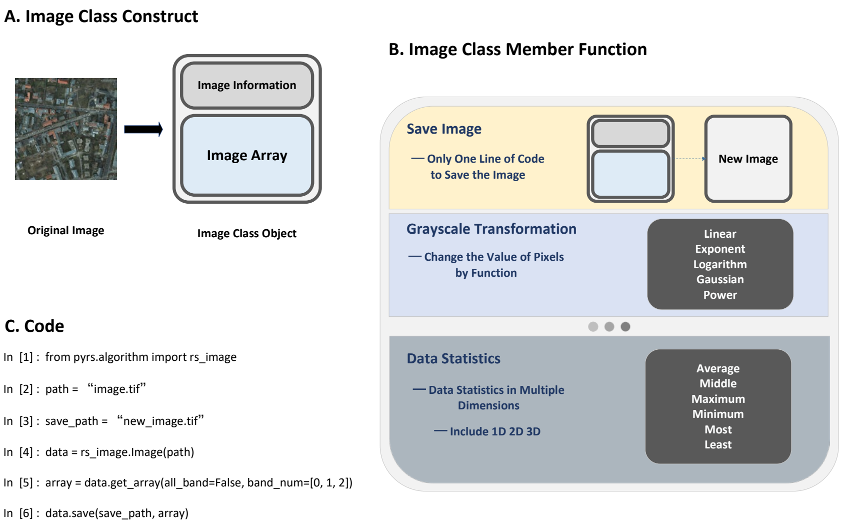


Figure 1. Image class (PyRS core class) data structure and function map.

Arrays constitute an essential component of raster imagery and serve as the core data structure for remote sensing image processing. GDAL can read two-dimensional arrays of each band of a dataset independently. The get_array function of the Image class can read Numpy two-dimensional arrays of shape (H, W) or three-dimensional arrays of shape (C, H, W) based on a user-specified single band index or a list of single band indices, thereby avoiding the need for users to manually combine two-dimensional arrays of single bands. The data type of an array is essential as every element of an array occupies the same amount of memory in RAM. The default data type of the Image class is float32, which users can change freely according to their needs and hardware memory restrictions. The array class based on Numpy encompasses all of Numpy's functionality and opens up channels for interaction with other Python packages.

The Image class provides a fast and efficient method for saving raster data, which avoids the tedious and redundant task of creating a data driver and using multiple functions to set various parameters for the raster data, such as row and column numbers, band number, data type, spatial reference, and projection information. Instead, the Image class only requires the use of a single save function and the setting of relevant parameters to complete the saving process, reducing the amount of code required and enhancing code readability.

The Image class also offers the capability to perform data statistics on multiple dimensions. For example, it can calculate statistical measures such as mean, variance, maximum, median, and mode within a specified two-dimensional array for a given band, or within a specified window of a user-defined size. Moreover, it can also calculate the statistics of multiple bands at a specified location in a three-dimensional array projected onto a two-dimensional space, thereby facilitating profile analysis.

The Image class provides various methods for gray-scale transformations, such as linear transformation, segmented linear transformation, exponential transformation, and logarithmic transformation, which can enhance the contrast of remote sensing images and highlight regions of interest while suppressing areas that are not of interest. Furthermore, users can customize their own gray-scale transformation functions according to their specific needs. This feature is particularly useful when presenting physical parameters obtained from quantitative inversion as images, as such images may suffer from low contrast due to the small differences in physical parameters within different spatial ranges. Gray-scale transformation can improve the visual quality of such images.

Different sensors use different data storage formats, with GeoTIFF or TIFF formats being the most commonly used raster data formats. However, data obtained from some sensors may include additional non-image information stored in a different format. For example, WorldView satellite series use RPB files to store image geometric correction information. To separate the RPB file from the original satellite image in NTF format, one usually needs to use gdal_translate with additional command statements, which can be complex and time-consuming. The Image class simplifies this process by providing a second-level encapsulation of the gdal_translate function, with optimized organization of parameter settings and clear structure, thereby enhancing user-friendliness with a single line of code for various data types conversion and file separation.

4. THE PYRS REMOTE SENSING IMAGE PROCESSING MODULES

PyRS provides comprehensive remote sensing image processing modules that offer a range of functionalities such as radiometric calibration, image enhancement, noise reduction, image segmentation, change detection, land cover classification, remote sensing index construction, and quantitative inversion. The module is built upon the Image class, which serves as the base class for all other modules and enables efficient implementation of corresponding functionalities. Fig. 2 shows the functionality of PyRS.

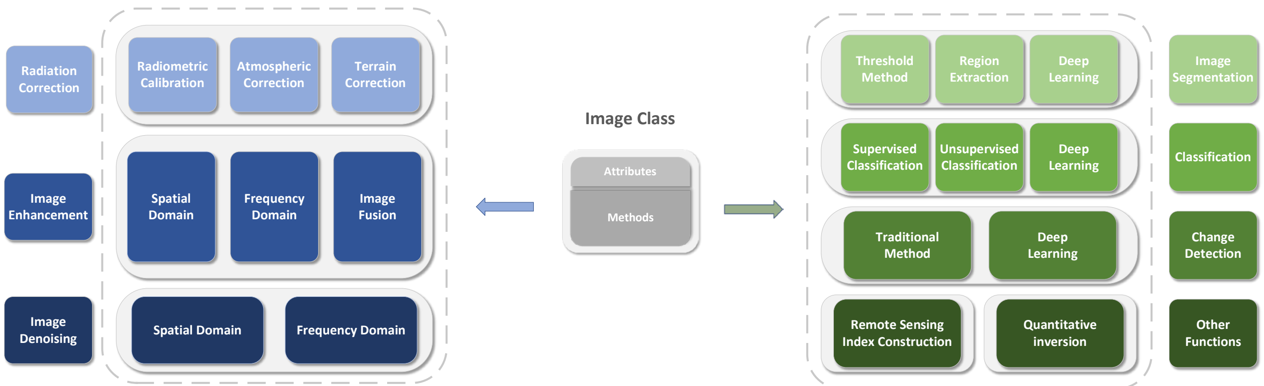


Figure 2. PyRS image processing function map.

4.1 Radiometric correction

Radiometric correction is a necessary preprocessing step for remote sensing images, and PyRS implements three steps for preprocessing remote sensing images: radiometric calibration, atmospheric correction, and correction for solar zenith angle and terrain effects. Radiometric calibration is the process of establishing the relationship between the DN values of a sensor and the actual received radiance values. Users can create a list based on the radiometric calibration coefficients in the image-related files and use the radiometric_calibration function of PyRS to achieve radiometric calibration. Atmospheric correction can be achieved through physical models or statistical models. The underlying model of physical model-based atmospheric correction is the 6S (Second Simulation of satellite Signal in the Solar Spectrum) radiation transfer model. The function atmospheric_correction_physical_model requires users to set five parameters: geometric parameters at the time of imaging, aerosol model, atmospheric model, spectral data, and reflectance model. PyRS has built-in .sli files for the spectral response functions of various sensors, so users do not need to download them themselves. There are two methods for statistical model-based atmospheric correction: the dark target method and the internal average relative reflectance method. The dark target method is based on the principle that darker targets in remote sensing images (such as water bodies and dense vegetation) have low reflectance, and the received radiance values are mainly caused by atmospheric radiation. PyRS can automatically search for dark targets and subtract the radiance values of all pixels from the radiance values of dark targets to achieve relative radiometric correction. The internal average relative reflectance method divides all pixels by the average radiance value of the entire image to achieve relative atmospheric correction. The statistical model does not require any other parameters, which is convenient and fast, but only achieves relative atmospheric correction and cannot obtain the true surface reflectance value. The physical model-based atmospheric correction requires more input parameters and complex calculations, but can obtain the true surface reflectance value. Users can choose the appropriate model according to their own needs and data resources. The purpose of solar zenith angle correction and terrain correction is to reduce the difference in radiance values of land cover with similar reflectance characteristics due to terrain and solar zenith angle, and users need to input relevant angle parameters to achieve the corresponding function.

4.2 Image enhancement

The process of enhancing images involves three parts: spatial domain, transformation domain, and data fusion, with some of the functions implemented using OpenCV functions. Histogram equalization in the spatial domain can increase the range of gray levels in pixels with a higher count, making the histogram of the image approximately uniformly distributed and enhancing image contrast. The PyRS histogram equalization function requires the user to set reference and target images and adjusts the histogram recursively. The first- and second-order

differential operators' functions are based on encapsulating OpenCV operators and have high computational efficiency. Digital morphological gradient operations provide three ways: dilation-erosion gradient, dilation-original gradient, and original-erosion gradient. In transformation domain image enhancement, the PyRS frequency domain filtering high_pass_filter function first performs Fourier transform on the image and high-pass filters the frequency domain image. The shape and size of the filter can be set freely by the user. The high-frequency components of the filtered image are subjected to Fourier inverse transform, segmented, and added to the original image to obtain an enhanced image. The homomorphic filtering function first performs a logarithmic transformation on the original image, converting the illumination and reflection components from multiplication to addition. It then performs Fourier transform, only retaining the reflection component (high-frequency part of the image), and performs inverse Fourier and logarithmic transformations to obtain an enhanced image. PyRS uses PCA for image enhancement, performs PCA transformation on the input image, stretches the first component's contrast, and then performs PCA inverse transformation. For data fusion, PyRS focuses on data-level fusion and uses a pixel-by-pixel calculation method to fuse low-resolution multispectral images with high-resolution single-band images to obtain high-resolution multispectral images. There are two methods of data fusion available to the user: PCA-based data fusion and color space transformation-based (RGB to HSV) data fusion. PyRS includes the functionality of remote sensing image time-series synthesis, with four methods available to combine multiple images within a certain time range into a single image: median algorithm (Tran et al., 2022), maximum NDVI algorithm (Roy et al., 2010), Medoid algorithm (Flood et al., 2013), and weighted average reflectance algorithm (Griffiths et al., 2019). These algorithms are still evolving in this field, but the ones used in PyRS have passed accuracy validation. PyRS also adds the function of image dehazing in the image enhancement module, using the dark channel prior dehazing algorithm (He et al., 2010), which has been widely used in the computer vision field. The most significant feature of PyRS in image enhancement is that it provides the mainstream image enhancement processing with just one line of code by calling the corresponding API and setting parameters, while most current remote sensing processing software requires multiple steps and involves many process data, which is more cumbersome. PyRS maximizes user convenience, improves image processing efficiency, and enhances image quality.

4.3 Image denoising

Image denoising in PyRS involves spatial domain denoising and frequency domain denoising. Spatial domain denoising includes mean filtering and median filtering. Frequency domain denoising involves applying a low-pass filter to the frequency domain image obtained by performing Fourier transform on the spatial domain image. This function is relatively simple and is mainly implemented by wrapping OpenCV functions.

PyRS integrates image noise type recognition, allowing users to set the window size. PyRS will traverse the image using this window, search for areas with high signal-to-noise ratio, calculate the histogram of this area, and match it with the histogram of common random noise types to determine the noise type.

4.4 Image segmentation

For image segmentation, PyRS uses thresholding and region extraction algorithms to separate the regions of interest from the background. The thresholding method includes three approaches for finding the optimal threshold: uniformity measurement method, maximum between-class variance, and maximum between-class variance (Otsu's method). Region extraction segmentation algorithms include region seed growth and region split-merge algorithms. The key to region seed growth algorithm is the selection of seed points and the establishment of pixel merging conditions.

PyRS not only integrates traditional image segmentation algorithms but also includes deep learning models for high-resolution remote sensing image semantic segmentation, including FCN (Dai et al., 2016) and DeeplabV3p (Chen et al., 2017). All deep learning models are implemented in Pytorch and are called in API form, eliminating the need for users to build their own networks. For specific datasets, PyRS provides the parameter weight files for each model. Users can import the model parameters into the corresponding model and apply the model to the segmentation scene, eliminating the time consumption and hardware limitations of model training.

4.5 Change detection

The change detection module of PyRS is divided into two categories: traditional remote sensing-based change detection and deep learning-based change detection. Traditional remote sensing-based change detection mainly uses the bi-temporal difference method and PCA transformation method, which are applicable to most remote sensing scenes, but with limited accuracy. Deep learning models are mainly used for building change detection, integrating the Transformer-based BIT model (Chen et al., 2021), which achieves SOTA performance on various building change detection datasets and is a lightweight model with a small number of parameters.

4.6 Land cover classification

Remote sensing image land cover classification can be divided into supervised and unsupervised classification. Supervised classification includes methods such as minimum distance, maximum likelihood, support vector machine, and random forest, while unsupervised classification includes K-Means and ISODATA classification methods.

For supervised classification, users need to pass the raster image or vector file used as classification labels as parameters to the function. Users can set the proportion of the number of pixels used for training and validation to the total original pixels. PyRS can randomly select training and validation pixels from various land cover types according to the set proportion. For high-resolution remote sensing images, if all bands are selected for training and validation at the same time, the required computer memory may exceed the user's memory limit. PyRS implements the function of selecting pixels by band and then splicing them, and users can choose different calculation methods according to their memory limitations.

Since some methods of land cover classification cannot be parallelized using vectors and can only be traversed pixel by pixel, implementing these functions using Python will consume a lot of computing time. Therefore, PyRS has implemented the corresponding functions using C++, which is efficient and fast.

4.7 Remote sensing index construction

Remote sensing indices are based on the unique spectral characteristics of land cover, used to highlight target land cover and reduce interference from other land cover types. PyRS integrates commonly used remote sensing indices, including various vegetation indices such as Normalized Difference Vegetation Index (NDVI), Ratio Vegetation Index (RVI), Difference Vegetation Index (DVI), Soil Adjusted Vegetation Index (SAVI), Greenness Vegetation Index (GVI), Perpendicular Vegetation Index (PVI), and Leaf Area Index (LAI). Users need to upload multispectral images and specify the band index list required for each vegetation index to obtain vegetation index calculation results. The Temperature Vegetation Dryness Index (TVDI) is a commonly used remote sensing index that characterizes vegetation water stress. By constructing the LST-NDVI feature space and using the maximum and minimum values of LST for pixels within each NDVI range, the "dry edge" and "wet edge" lines are fitted, and the soil moisture content is measured by the relative distance between the LST value of the pixel and the two lines. The Normalized Difference Building Index (NDBI) and Normalized Difference Water Index (NDWI) can be used to extract building and water pixels, respectively. Users can use the PyRS remote sensing index module to obtain relevant indices without calculating them on their own, thereby avoiding errors in remote sensing index calculation caused by data format setting errors.

4.8 Quantitative Inversion

PyRS incorporates mature physical parameter inversion algorithms, many of which have been widely used. These include surface temperature inversion algorithms, which include single-window algorithms and split-window algorithms. The split-window algorithm mainly uses the difference in water vapor absorption in different thermal infrared bands to eliminate atmospheric effects. The split-window algorithm uses a linear combination of brightness temperatures in two bands to solve for surface temperature. The advantage of the split-window algorithm is that it can avoid using a large number of atmospheric parameters and can use its own information for atmospheric correction, but the inversion accuracy has certain errors. The single-window algorithm directly solves the surface temperature through the radiation transfer equation of the thermal infrared band and is related to information such as the radiation of the atmosphere in the up and down directions, the temperature and content of the atmosphere itself, and the water vapor content. It requires real-time atmospheric profile data, but obtaining real-time atmospheric profile data is difficult, which makes the surface temperature inversion process based on physical processes complicated. PyRS provides both of these algorithms for users to choose from. Net Primary Productivity (NPP) of vegetation reflects the total amount of organic matter accumulated by green plants in a unit area and unit time through photosynthesis. Common methods for NPP inversion include establishing a linear model based on measured NPP data, meteorological data, and vegetation data, and the CASA model based on the vegetation growth process. PyRS performs NPP inversion based on the CASA model by calculating the Absorbed Photosynthetically Active Radiation (APAR) and the vegetation light use efficiency (ε). Because the CASA model requires users to input all NDVI data, meteorological data, and vegetation type parameters for the study area for one year, there are many parameters. PyRS proposes a simplified mode of the CASA model, but the accuracy is unsatisfied.

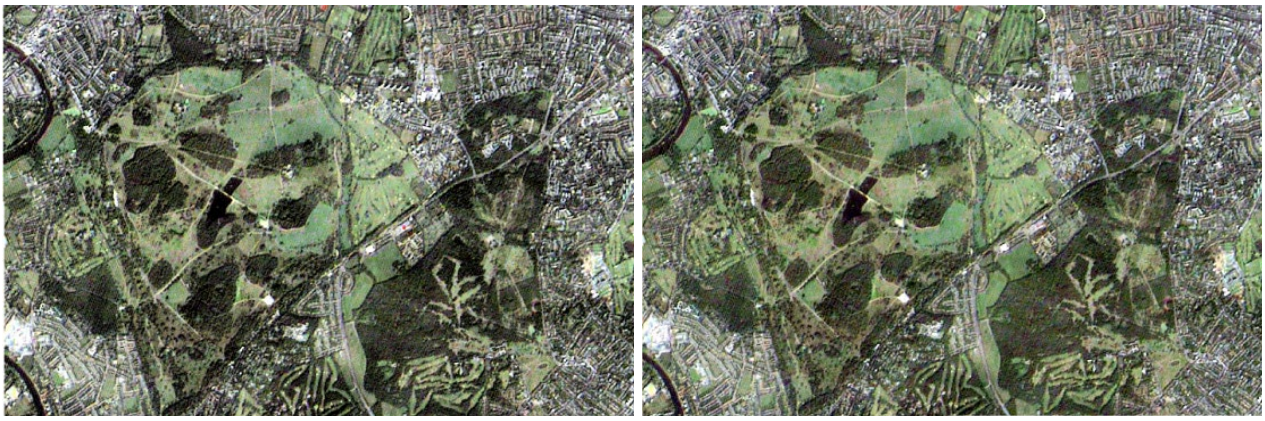
4.9 Performance comparison

In the following, we will use PyRS and ENVI to implement image fusion and minimum distance classification, respectively; PyRS encapsulates the complex remote sensing image processing process into an API, which can be implemented with only one line of code. On the contrary, the implementation of image fusion with ENVI requires tedious operations, including image resampling, color space conversion, component replacement, color space inverse transformation, etc. The comparison can show the convenience of PyRS. In addition, image fusion and minimum distance classification are more complicated than image denoising, change detection, remote sensing index construction and other functions.

This image fusion method first converts multispectral images from RGB format to HSV format, replaces the V component using high-resolution images, and then converts them to RGB format. There may be differences between PyRS and ENVI in the format conversion process, which is the reason for the differences in the resultant images. Therefore, we randomly selected 1% of all image pixels and used a t-test to compare the differences in the H, S, and V components of the PyRS and ENVI results. As can be seen from the Table 1, there is no difference in the H component, the S and V components are more different, the brightness of the PyRS results is higher than that of the ENVI results, and the saturation is lower than that of the ENVI results (Fig. 3).

Table 1. HSV component t-test and mean result

Components	probability	PyRS Mean	ENVI Mean
Hue	0.54	67.49	**67.65**
Saturation	2.41E-76	41.48	**44.5**
Value	1.03E-38	**125.8**	120.89



ENVI PyRS

Figure 3. Image fusion results of ENVI and PyRS.

For the classification results, the classification accuracy of the two is also similar, when considering the overall accuracy, the minimum distance classification algorithm implemented in PyRS outperform the one in ENVI, and the average F1 score of the classification results of ENVI is higher (Table 2). Smoother classification results with PyRS (Fig. 4). The F1 scores of the Grassland/Pasture and Sorghum categories of the PyRS classification results are higher than ENVI, and the rest are lower than ENVI (Table 3)

Table 2. The overall accuracy and mean F1 score of the classiffication result.

Method	PyRS	ENVI
OA	**0.867**	0.8609
Mean F1	0.8471	**0.8518**

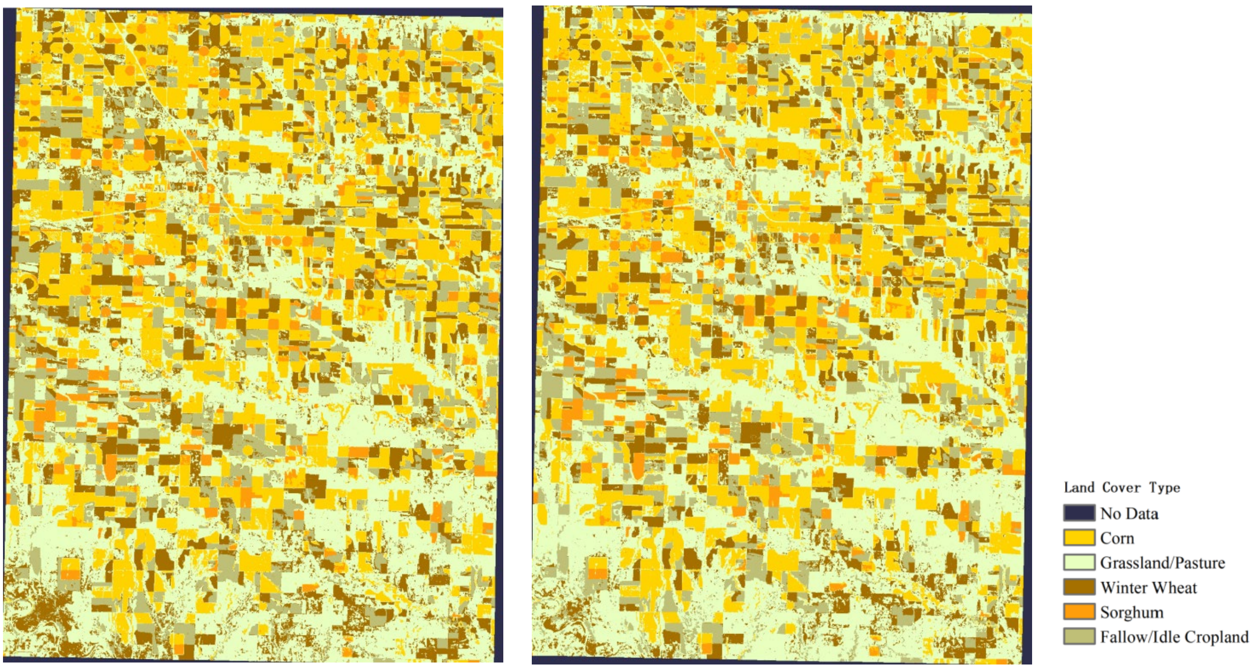
Table 3. F1 scores in all categories.

Method	PyRS	ENVI
Corn	**0.9305**	0.9245
Grassland/Pasture	0.8023	0.8226
Winter Wheat	**0.7523**	0.7436
Sorghum	0.8781	0.9135
Fallow/Idle Cropland	**0.8725**	0.8549

5. PYRS SPECIAL MECHANISMS

5.1 The intermediate process data retention mechanism

The intermediate process data retention mechanism is commonly employed in the implementation of remote sensing related functions, which often generate a plethora of process data. For instance, during the inversion of land surface temperature using the single-window algorithm, raster images of surface emissivity in the study area are produced. The computation of leaf area index (LAI) also leads to the generation of normalized difference vegetation index (NDVI) data. PyRS encapsulates various functionalities in functions, unlike traditional remote sensing image processing software, which do not require users to operate step by step. However, in specific cases, retaining some intermediate process data is necessary. Therefore, PyRS has set the save_prodata parameter in each



ENVI PyRS

Figure 4. Classification result graph of ENVI and PyRS.

API. If a user sets this parameter to True, PyRS will automatically save the necessary intermediate process data in the same directory as the specified result file path, thereby making the image processing process more transparent and verifiable.

5.2 Band Segmentation Mechanism

In the context of image processing, the high resolution images often occupy a large amount of memory, and their processing imposes significant memory requirements on the computer system. In particular, when processing multiple bands in parallel, the memory constraints may cause errors. To address this challenge, PyRS offers two processing modes: multi-band parallel processing and per-band processing. The latter mode enables users to perform calculations on individual bands one at a time, thereby avoiding the memory issue that arises when processing all bands in parallel.

In implementing remote sensing functionalities, a large amount of process data is often generated. For example, using the single-window algorithm to retrieve land surface temperature produces a raster image of surface emissivity in the study area. Similarly, calculating leaf area index produces NDVI data. As high-resolution imagery requires large memory and computing power, PyRS provides two processing modes: parallel processing for multiple bands and segmented processing.

6. DISCUSSION

The PyRS community has divided its development process into four stages, namely version control, unit testing, code review, and issue tracking, which has compensated for the inadequacies of scientific researchers in building software ecosystems. Additionally, PyRS currently utilizes CPU for computations, which may consume a considerable amount of time when processing large-scale data. In the future, PyRS will support GPU parallel computing, which is expected to significantly enhance algorithmic implementation efficiency. Currently, PyRS only supports the processing of multi-spectral images ranging from visible light to thermal infrared bands. However, in the future, it will add processing capabilities for hyperspectral and synthetic aperture radar (SAR) images.

7. CONCLUSION

We proposed a Python remote sensing image processing library, introduced its main functions and characteristics, selected two functions to implement them with PyRS and ENVI, and compared the results. PyRS makes up for the missing gap in the image processing algorithm library in the field of remote sensing, and provides users with a highly transparent and repeatable workflow. The community will continue to add new algorithms to PyRS. PyRS will contribute to remote sensing technology and geomatics education.

ACKNOWLEDGEMENT

Thanks to the supports from Natural Science Foundation of Shandong Province (No.ZR2022QD141), National Natural Science Foundation of China (No.42271273) and the China Aerospace Science and Technology Corporation 2021 Qian Xuesen Innovative Youth Fund Project. Thanks to anonymous reviewers for their hard work.

REFERENCES

Chen, H., Qi, Z., & Shi, Z, 2021. Remote sensing image change detection with transformers. IEEE Transactions on Geoscience and Remote Sensing, 60, 1-14

Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A.L., 2017. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE transactions on pattern analysis and machine intelligence, 40, 834-848

Dai, J., Li, Y., He, K., & Sun, J., 2016. R-fcn: Object detection via region-based fully convolutional networks. Advances in neural information processing systems, 29

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., & Gelly, S., 2020. An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929

Flood, N. 2013., Seasonal composite Landsat TM/ETM+ images using the medoid (a multi-dimensional median). Remote Sensing, 5, 6481-6500

Griffiths, P., Nendel, C., & Hostert, P., 2019. Intra-annual reflectance composites from Sentinel-2 and Landsat for national-scale crop and land cover mapping. Remote Sensing of Environment, 220, 135-151

Harris, C., Millman, K., van der Walt, S., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., & Berg, S., 2020. Smith 474 nj. Kern R, Picus M, Hoyer S, van Kerkwijk MH, Brett M, Haldane A, del R'ıo JF, Wiebe M, Peterson P, G'erard-475 Marchant P, et al. Array programming with NumPy. Nature, 585, 357-362

He, K., Sun, J., & Tang, X., 2010. Single image haze removal using dark channel prior. IEEE transactions on pattern analysis and machine intelligence, 33, 2341-2353

He, K., Zhang, X., Ren, S., & Sun, J., 2016. Deep residual learning for image recognition. In, Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778)

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., & Antiga, L., 2019. Pytorch: An imperative style, high-performance deep learning library. Advances in neural information processing systems, 32

Roy, D.P., Ju, J., Kline, K., Scaramuzza, P.L., Kovalskyy, V., Hansen, M., Loveland, T.R., Vermote, E., & Zhang, C., 2010. Web-enabled Landsat Data (WELD): Landsat ETM+ composited mosaics of the conterminous United States. Remote Sensing of Environment, 114, 35-49

Tran, K.H., Zhang, H.K., McMaine, J.T., Zhang, X., & Luo, D., 2022. 10 m crop type mapping using Sentinel-2 reflectance and 30 m cropland data layer product. International Journal of Applied Earth Observation and Geoinformation, 107, 102