User's Manual for

EASRAPP v1.0

An Open-Source Semi-Automatic Python GUI-Based Application for Extraction and Analysis of Surface Ruptures in a large earthquake

Released in September, 2022

Dongchen, Li. and Junjie, Ren. (2022). EASRAPP: An Open-Source Semi-Automatic Python GUI-Based Application for Extraction and Analysis of Surface Ruptures in a large earthquake, Seismol. Res. Lett. (In submission)

Contact: **lidongchen20@mails.ucas.ac.cn**

EASRAPP is a graphical Python application that provides an interactive, user-friendly framework. It is designed to semi-automatically extract surface ruptures of a large earthquake and relevant quantitative parameters (including strikes, lengths, and widths) from remote sensing images in the GeoTIFF format (Ritter and Ruth, 1997); and semi-automatically analyze the width and strike of the extracted result. In addition, the part modules of EASRAPP(e.g., the RoughSeg_FineExt, EditVector and WidthAndStrike modules) require the input image to be a three-channel optical image with the projection coordinate system, a pixel type of 'unsigned integer' and a pixel depth of '8 Bit' (i.e., "0-255" for the pixel value extent of each channel). EASRAPP consists of four main modules for obtaining the region of interest for surface rupture in a remote sensing image, extracting surface ruptures, editing the vector extraction results, and analyzing the width of the surface rupture zone and strikes of all surface ruptures. Moreover, some auxiliary tools are available, including a data structure conversion toolset for vector and raster, a vector merging tool, a raster mosaicing tool, and a raster cropping tool for batch cropping multiple large images to many small images. EASRAPP can also be downloaded from https://github.com/GiserLi157/ProjectEASRAPP .

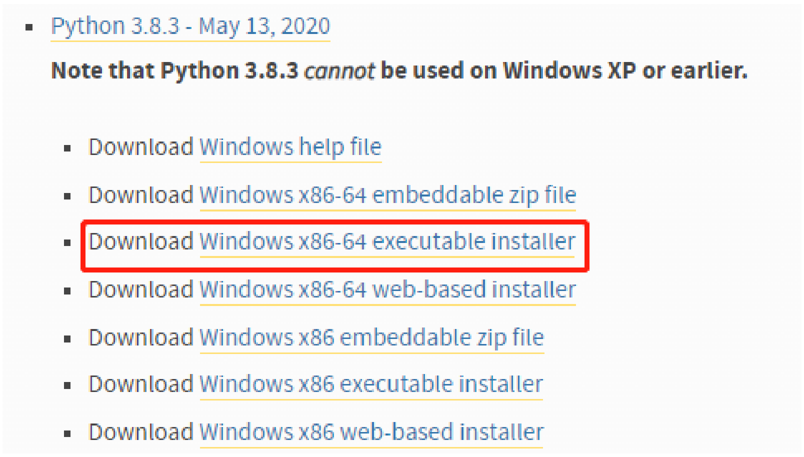
References
Ritter, N., and M. Ruth (1997). The GeoTiff data interchange standard for raster geographic images, Int. J. Remote Sens. 18, no. 7, 1637–1647, doi: 10.1080/014311697218340.

The demonstration for EASRAPP is based on the Python 3.8 platform. The user can also use other versions of Python, but the Python packages requested in the file named 'requirement.txt' must be installed.

A user manual is given below.

1 Deploy the program runtime environment

Step1 Download Python 3.8 from https://www.python.org/downloads/windows/ (Figure 1).



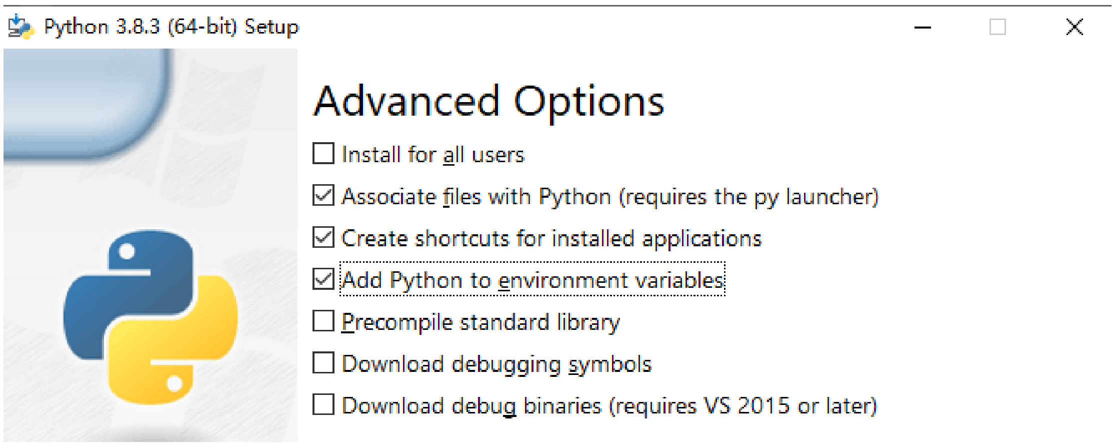
- Python 3.8.3 - May 13, 2020

  Note that Python 3.8.3 *cannot* be used on Windows XP or earlier.

  - Download Windows help file
  - Download Windows x86-64 embeddable zip file
  - Download Windows x86-64 executable installer
  - Download Windows x86-64 web-based installer
  - Download Windows x86 embeddable zip file
  - Download Windows x86 executable installer
  - Download Windows x86 web-based installer

Figure 1 Installer download interface

Step 2 Install Python and add Python to environment variables (Figure 2).



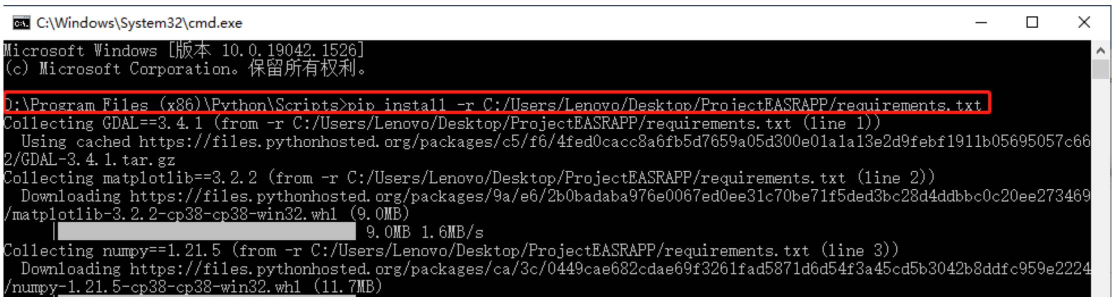
Python 3.8.3 (64-bit) Setup

Advanced Options

- [ ] Install for all users
- [x] Associate files with Python (requires the py launcher)
- [x] Create shortcuts for installed applications
- [x] Add Python to environment variables
- [ ] Precompile standard library
- [ ] Download debugging symbols
- [ ] Download debug binaries (requires VS 2015 or later)

Figure 2 Setup process: Advanced Options

Step 3 Open the windows command line under 'installation directory/script' and we will find that we use the command 'pip install -r 'download path/ProjectEASRAPP/ requirements.txt'' not to install the Python packages specified in the 'requirements.txt' file (Figure 3), so we need to install the Python packages manually.

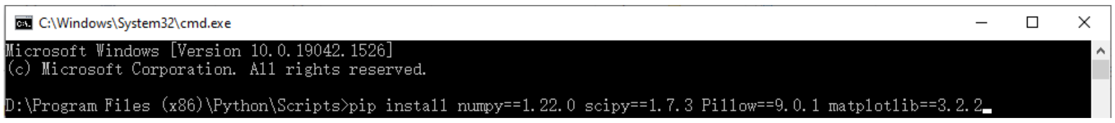


```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19042.1526]
(c) Microsoft Corporation。保留所有权利。

D:\Program Files (x86)\Python\Scripts>pip install -r C:/Users/Lenovo/Desktop/ProjectEASRAPP/requirements.txt
Collecting GDAL==3.4.1 (from -r C:/Users/Lenovo/Desktop/ProjectEASRAPP/requirements.txt (line 1))
  Using cached https://files.pythonhosted.org/packages/c5/f6/4fed0cacc8a6fb5d7659a05d300e01a1a13e2d9febf1911b05695057c66
2/GDAL-3.4.1.tar.gz
Collecting matplotlib==3.2.2 (from -r C:/Users/Lenovo/Desktop/ProjectEASRAPP/requirements.txt (line 2))
  Downloading https://files.pythonhosted.org/packages/9a/e6/2b0badaba976e0067ed0ee31c70be71f5ded3bc28d4ddbbc0c20ee273469
/matplotlib-3.2.2-cp38-cp38-win32.whl (9.0MB)
     |████████████████████████████████| 9.0MB 1.6MB/s
Collecting numpy==1.21.5 (from -r C:/Users/Lenovo/Desktop/ProjectEASRAPP/requirements.txt (line 3))
  Downloading https://files.pythonhosted.org/packages/ca/3c/0449cae682cdae69f3261fad5871d6d54f3a45cd5b3042b8ddfc959e2224
/numpy-1.21.5-cp38-cp38-win32.whl (11.7MB)
```

Figure 3 try to install the Python packages

Step 3.1 Install some Python packages at the windows command line with the command (Figure 4).

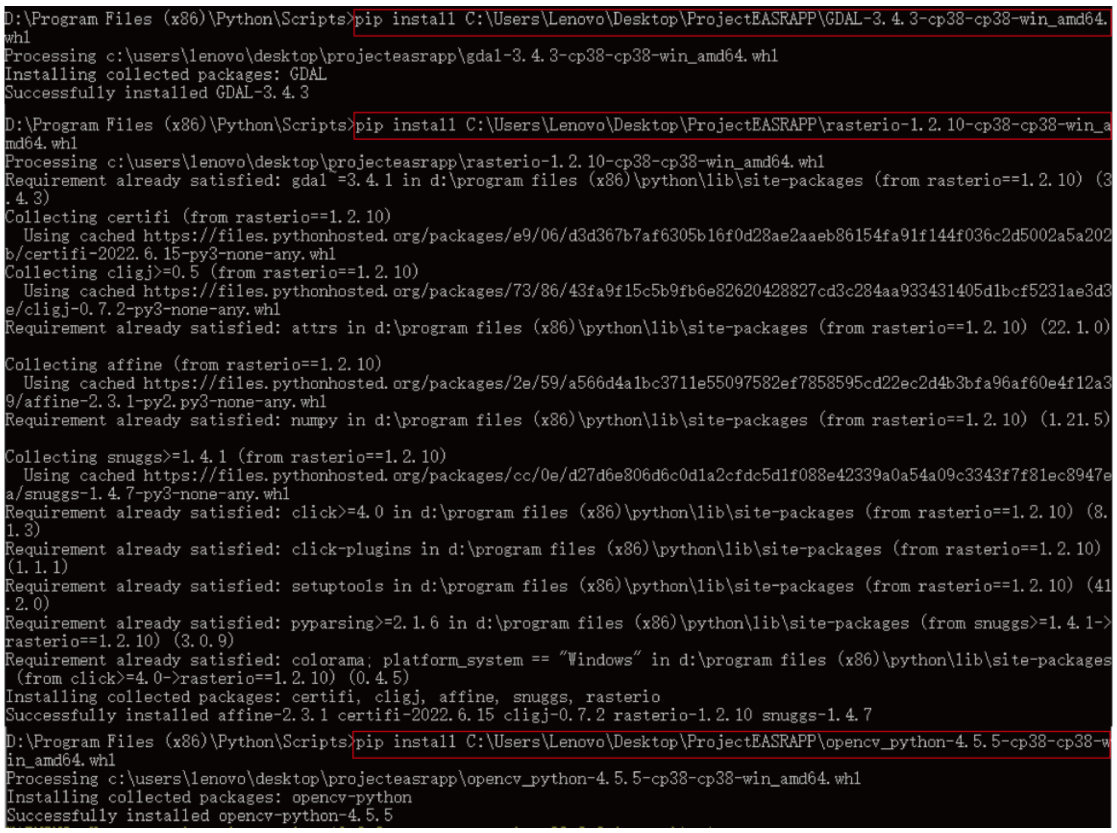


```
C:\Windows\System32\cmd.exe
Microsoft Windows [Version 10.0.19042.1526]
(c) Microsoft Corporation. All rights reserved.

D:\Program Files (x86)\Python\Scripts>pip install numpy==1.22.0 scipy==1.7.3 Pillow==9.0.1 matplotlib==3.2.2
```

Figure 4 Install the part of the Python packages

Step 3.2 Install the GDAL and rasterio packages using the installers downloaded from https://www.lfd.uci.edu/~gohlke/pythonlibs/ (Figure 5).



```
D:\Program Files (x86)\Python\Scripts>pip install C:\Users\Lenovo\Desktop\ProjectEASRAPP\GDAL-3.4.3-cp38-cp38-win_amd64.
whl
Processing c:\users\lenovo\desktop\projecteasrapp\gdal-3.4.3-cp38-cp38-win_amd64.whl
Installing collected packages: GDAL
Successfully installed GDAL-3.4.3

D:\Program Files (x86)\Python\Scripts>pip install C:\Users\Lenovo\Desktop\ProjectEASRAPP\rasterio-1.2.10-cp38-cp38-win_a
md64.whl
Processing c:\users\lenovo\desktop\projecteasrapp\rasterio-1.2.10-cp38-cp38-win_amd64.whl
Requirement already satisfied: gdal~=3.4.1 in d:\program files (x86)\python\lib\site-packages (from rasterio==1.2.10) (3
.4.3)
Collecting certifi (from rasterio==1.2.10)
  Using cached https://files.pythonhosted.org/packages/e9/06/d3d367b7af6305b16f0d28ae2aaeb86154fa91f144f036c2d5002a5a202
b/certifi-2022.6.15-py3-none-any.whl
Collecting cligj>=0.5 (from rasterio==1.2.10)
  Using cached https://files.pythonhosted.org/packages/73/86/43fa9f15c5b9fb6e82620428827cd3c284aa933431405d1bcf5231ae3d3
e/cligj-0.7.2-py3-none-any.whl
Requirement already satisfied: attrs in d:\program files (x86)\python\lib\site-packages (from rasterio==1.2.10) (22.1.0)

Collecting affine (from rasterio==1.2.10)
  Using cached https://files.pythonhosted.org/packages/2e/59/a566d4a1bc3711e55097582ef7858595cd22ec2d4b3bfa96af60e4f12a3
9/affine-2.3.1-py2.py3-none-any.whl
Requirement already satisfied: numpy in d:\program files (x86)\python\lib\site-packages (from rasterio==1.2.10) (1.21.5)

Collecting snuggs>=1.4.1 (from rasterio==1.2.10)
  Using cached https://files.pythonhosted.org/packages/cc/0e/d27d6e806d6c0d1a2cfdc5d1f088e42339a0a54a09c3343f7f81ec8947e
a/snuggs-1.4.7-py3-none-any.whl
Requirement already satisfied: click>=4.0 in d:\program files (x86)\python\lib\site-packages (from rasterio==1.2.10) (8.
1.3)
Requirement already satisfied: click-plugins in d:\program files (x86)\python\lib\site-packages (from rasterio==1.2.10)
(1.1.1)
Requirement already satisfied: setuptools in d:\program files (x86)\python\lib\site-packages (from rasterio==1.2.10) (41
.2.0)
Requirement already satisfied: pyparsing>=2.1.6 in d:\program files (x86)\python\lib\site-packages (from snuggs>=1.4.1->
rasterio==1.2.10) (3.0.9)
Requirement already satisfied: colorama; platform_system == "Windows" in d:\program files (x86)\python\lib\site-packages
 (from click>=4.0->rasterio==1.2.10) (0.4.5)
Installing collected packages: certifi, cligj, affine, snuggs, rasterio
Successfully installed affine-2.3.1 certifi-2022.6.15 cligj-0.7.2 rasterio-1.2.10 snuggs-1.4.7
D:\Program Files (x86)\Python\Scripts>pip install C:\Users\Lenovo\Desktop\ProjectEASRAPP\opencv_python-4.5.5-cp38-cp38-w
in_amd64.whl
Processing c:\users\lenovo\desktop\projecteasrapp\opencv_python-4.5.5-cp38-cp38-win_amd64.whl
Installing collected packages: opencv-python
Successfully installed opencv-python-4.5.5
```

Figure 5 Install gdal, rasterio and OpenCV-python package

2 Main window

Run the main program 'EASRAPP.py' and the graphical user interface (GUI) of EASRAPP appears (see Figure 6). EASRAPP consists of four main modules and seven tools. The user can execute the corresponding functions by selecting the function buttons. As these functions are independent and called in multiple threads, the user can use different or several of the same processing programs simultaneously to avoid programs stuttering (e.g., running 'CustomCrop' while also running 'EditVector').

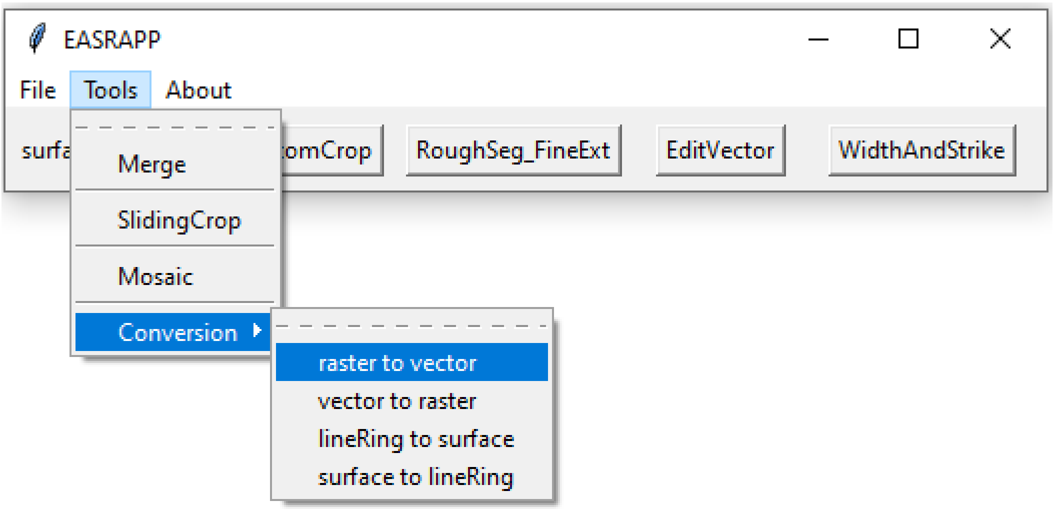


Figure 6 main window

3 CustomCrop module

The user can click the **'CustomCrop'** button from the main window (Figure 6). The remote sensing image in the GeoTIFF format can be fed into the application via the **'Ok'** button (Figure 7). The example data named '0525-0.01mDOM.tif' of this study is a three-channels UAV orthorectified image with a resolution of approximately 0.01m, and its projection coordinate system is 'WGS_1984_UTM_ Zone_47N'. **The module's crop function is also available for single channel images or floating point images in the GeoTIFF format.** Users can also use the single-band image 'binary.tif' from the 'CustomCrop&SlidingCrop' folder for testing.

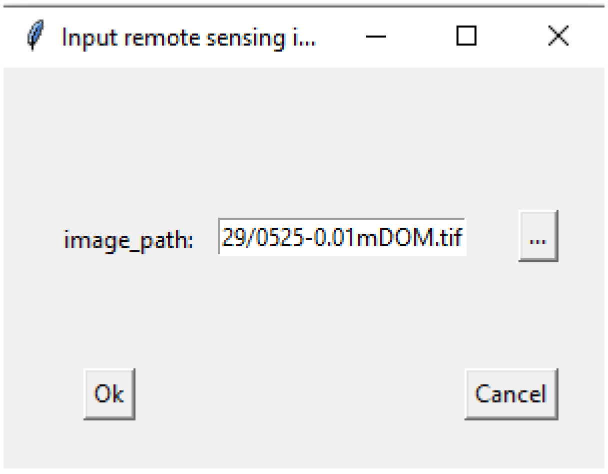
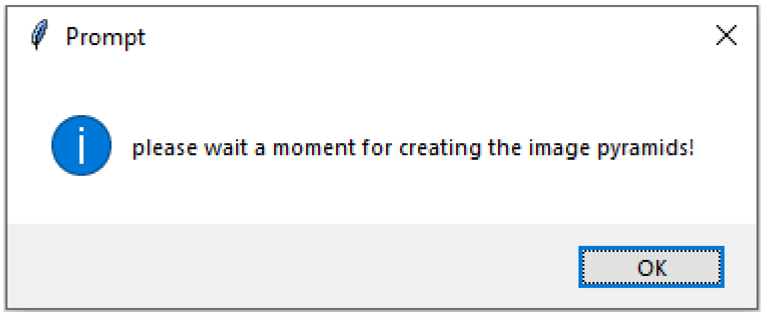


Figure 7 Input the remote sensing image

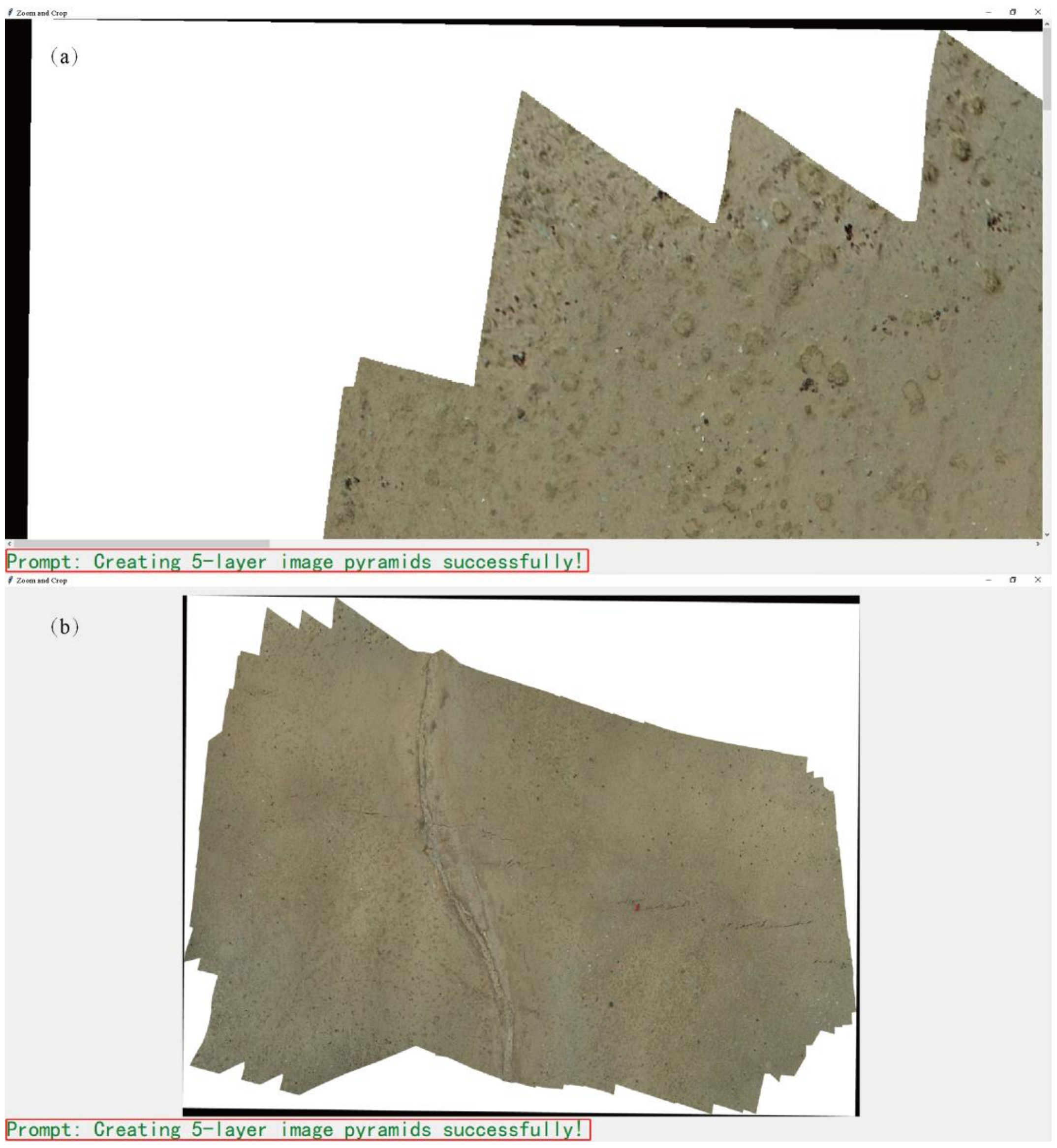
Step 1 After clicking the **'Ok'** button (Figure 8), the application will generate image pyramids to quickly achieve zoom-in or zoom-out for facilitating a large image to be viewed.


Prompt

please wait a moment for creating the image pyramids!

Figure 8 Prompt: Creating the image pyramids takes some time

Step 2 Figure 9a shows the program interface with the loaded image. The user can move the image by pressing the '**mouse wheel**' and dragging the image or the **scrollbar** when the interface can not fully display the image, and then can also scroll with the **mouse wheel** to zoom in or out the image to the proper size (Figure 9b).

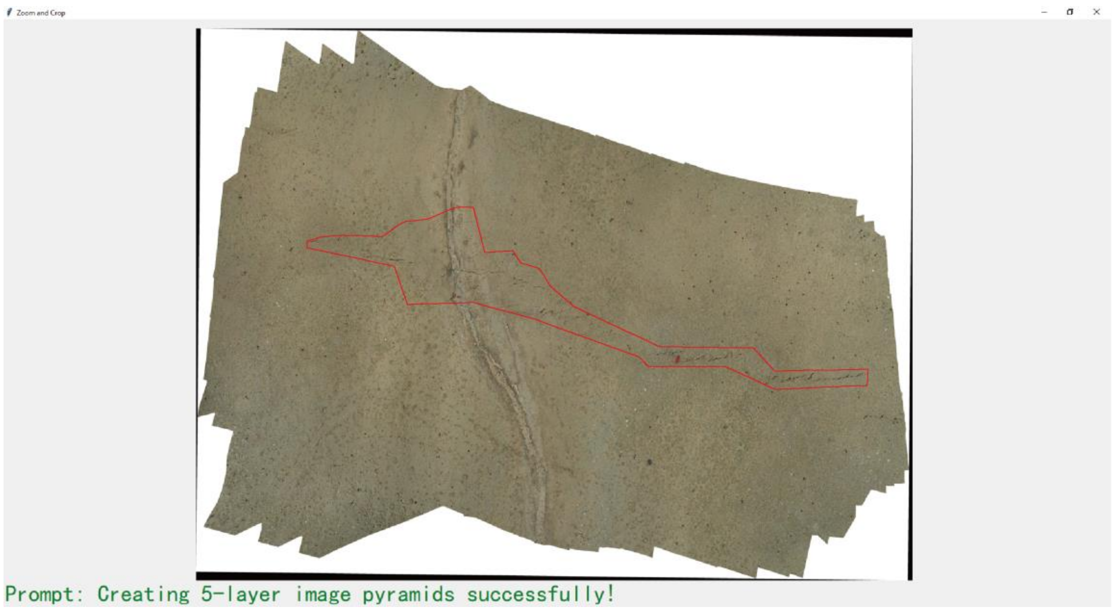


```
Prompt: Creating 5-layer image pyramids successfully!
```

```
Prompt: Creating 5-layer image pyramids successfully!
```

Figure 9 ‘Zoom and Crop’ interface, in which the green font in the red box is the prompt message. (a) Before moving and zooming the image; (b) After moving and zooming the image

Step 3 The user can select the inflection point of the area of interest (AOI) with a left mouse click and can zoom in or out and move at any time to determine the location of the inflection point. If the wrong inflection point is selected, it can be undone using the **‘ctrl+z’** shortcut. If only two points are chosen, the cropping operation cannot be completed because they cannot form an AOI; the last inflection point chosen does not have to coincide with the first one because the application will automatically generate a closed polygon to crop the image. After selecting the AOI (Figure 10), the user can save the result using the **‘ctrl+s’** shortcut. The cropped result can be saved as an image in the GeoTIFF format. The above operation can be repeated several times to crop multiple images of appropriate size from one large image.



Prompt: Creating 5-layer image pyramids successfully!

Figure 10 ‘Zoom and Crop’ interface in which the polygon in red is the area of interest surrounding surface ruptures

4 RoughSeg_FineExt module

Step 1 The user clicks the **‘RoughSeg_FineExt’** button (see Figure 6), and then an interface, including three buttons: **‘colorSeg’**, **‘fineExt’**, **‘colorSeg_fineExt’**, pops up (Figure 11).

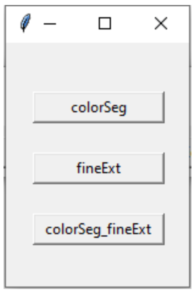
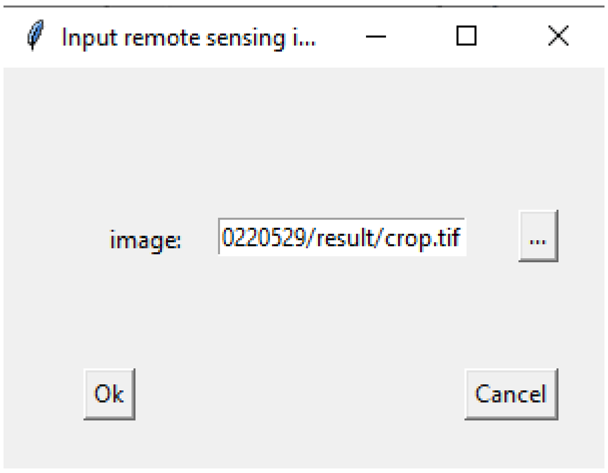


Figure 11 ‘RoughSeg_FineExt’ interface

The algorithms corresponding to the **‘coloring’** button and the **‘fineExt’** button are the front and back parts of the ‘rough segmentation - fine extraction’ algorithm corresponding to the **‘colorSeg_fineExt’** button, respectively. The user can choose to use the entire extraction algorithm or its part. Therefore, only the specific steps of the 'colorSeg_fineExt' functionality are described next.

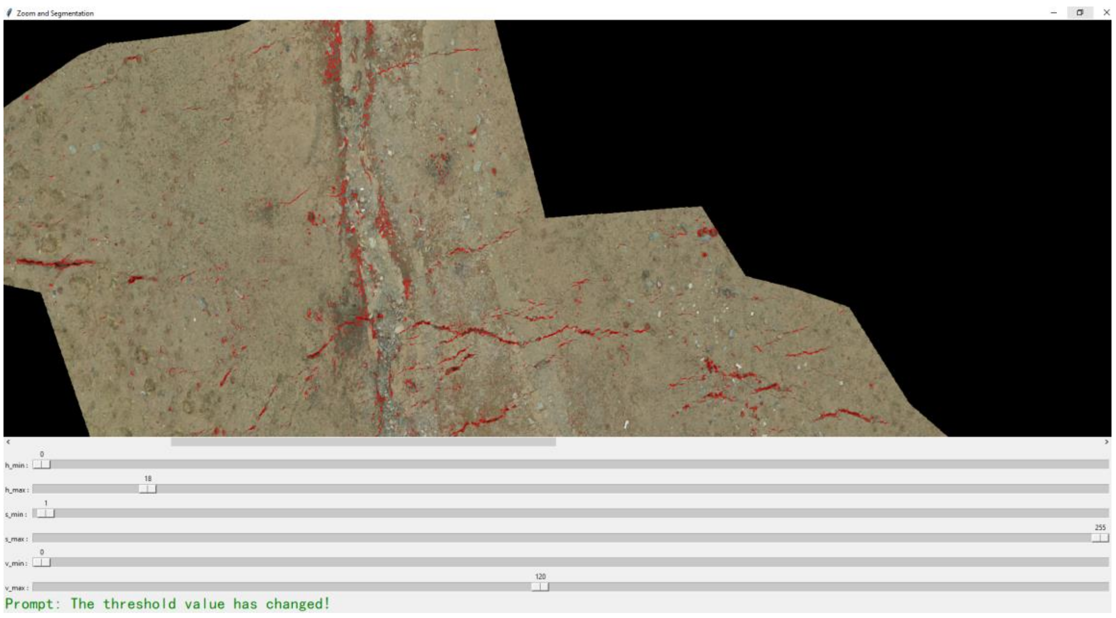
Step 2 After clicking the **‘colorSeg_fineExt’** button, the user can input the path of the image (Figure 12). The application will also generate image pyramids to achieve zoom-in or zoom-out quickly after clicking the **‘Ok’** button. The same operation as the 'CustomCrop' module allows for zooming and moving the image. While 'crop.tif' in Figure 12 is the output result of the 'Zoom and Crop' module.



0220529/result/crop.tif

Figure 12 Input the image.

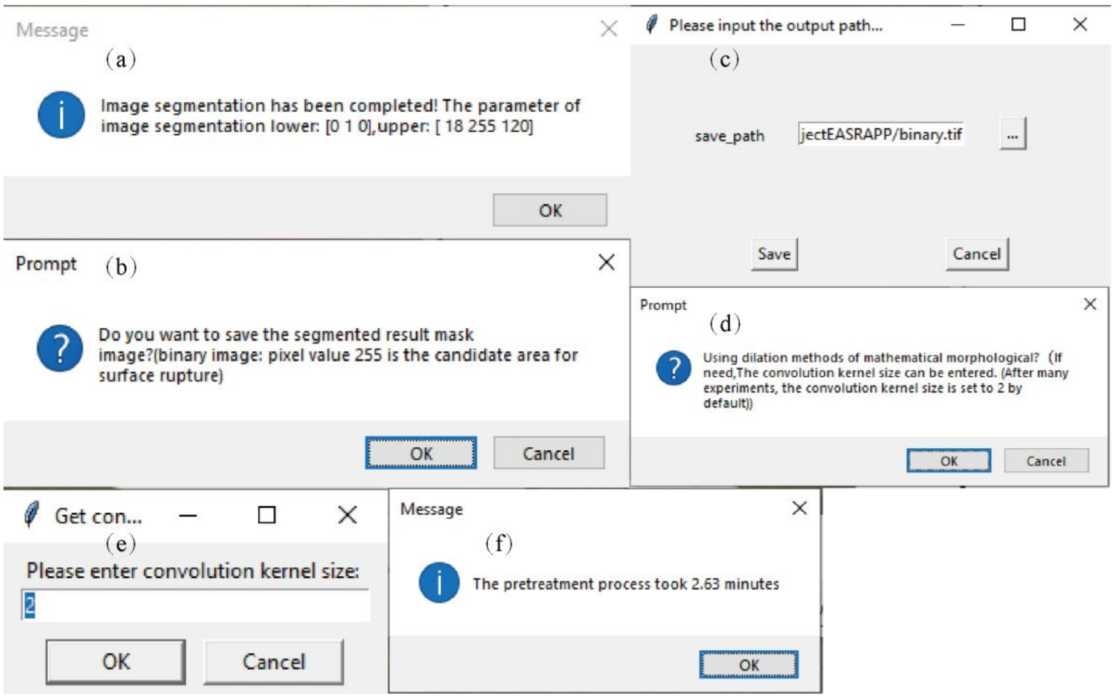
Step 3 After inputting the image, the 'Zoom and Segmentation' interface first pops up. The user can adjust the values of H, S, and V parameters via '**sliders**' to obtain the result of image color segmentation in real time (Figure 13). In addition, the user can move, zoom in, or out at any time to observe the segmentation effect. After adjusting to the appropriate threshold, the user may close the 'Zoom and Segmentation' interface by pressing the '**Q**' shortcut key or clicking the '**Close**' button.



```
h_min : 0
h_max : 18
s_min : 1
s_max : 255
v_min : 0
v_max : 120
Prompt: The threshold value has changed!
```

Figure 13 Adjust color parameter values via sliders.

Step 4 Figure 14 shows a series of prompts and requests. Figure 14a shows the prompt message for the segmentation parameters. The user can choose whether or not to output a black and white binary image (Figure 14b, c). The user can also decide whether to inflate the segmentation results to obtain a smoother and more accurate segmentation result. After several experiments, the best result is obtained by setting the convolution kernel to 2 in the expansion method (Figure 14d, e). Figure 14f shows the time taken to dilate the segmentation result and compute the parameters used in the 'fineExt' extraction process.


Message

(a)

Image segmentation has been completed! The parameter of image segmentation lower: [0 1 0],upper: [18 255 120]

OK

Please input the output path...

(c)

save_path jectEASRAPP/binary.tif

Save Cancel

Prompt (b)

Do you want to save the segmented result mask image?(binary image: pixel value 255 is the candidate area for surface rupture)

OK Cancel

Prompt

(d)

Using dilation methods of mathematical morphological? (If need,The convolution kernel size can be entered. (After many experiments, the convolution kernel size is set to 2 by default))

OK Cancel

Get con...

(e)

Please enter convolution kernel size:

2

OK Cancel

Message

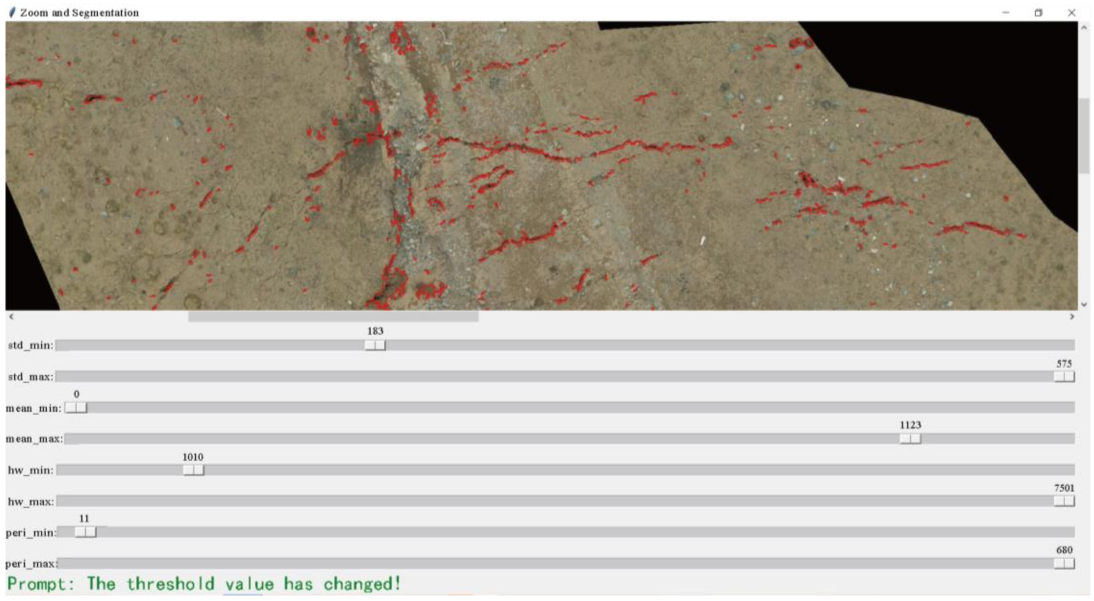
(f)

The pretreatment process took 2.63 minutes

OK

Figure 14 A series of prompts and requests

Step 5 The user can adjust the shape parameters of the extracted feature via '**sliders**' to screen non-extracted objects. Please see my paper for the definition of shape parameters (Figure 15). Likewise, the user can move and zoom the image at any time to see the extraction effect.



```
std_min: 183
std_max: 575
mean_min: 0
mean_max: 1123
hw_min: 1010
hw_max: 7501
peri_min: 11
peri_max: 680
Prompt: The threshold value has changed!
```

Figure 15 Adjust shape parameters values via sliders

Step 6 Similarly, the results may be saved by the save interface (Figure 16) after pressing the '**Q**' shortcut or clicking the '**Close**' button. Additionally, the user may choose to output both or one of these results.

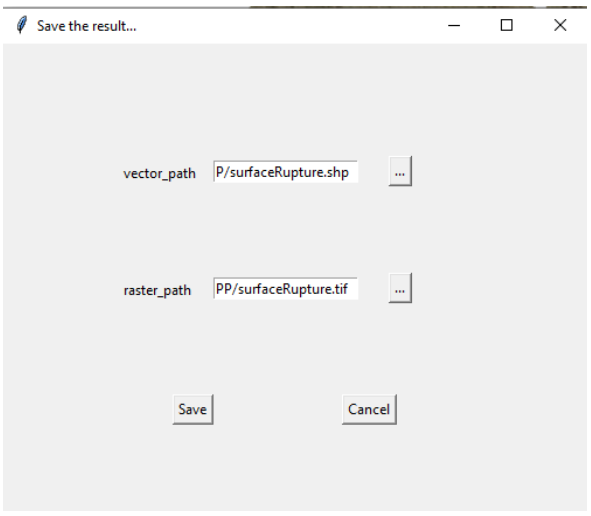


Figure 16 result save interface

5 EditVector module

After clicking the **'EditVector'** button, the user can input the vector in .shp format and the corresponding remote sensing image(the original image named '0525-0.01mDOM.tif' was input for this test, but you can actually also input the cropping result named 'crop.tif ' generated by the 'CustomCrop' module) in .tif format through the input data interface (Figure 17). The vector data also needs to contain three floating-point fields (named 'length', 'width' and 'angle'). If the values of one of these three fields is '-1', the equivalent length, equivalent width and the dominant orientation of each surface rupture will be recalculated and written into these three fields. Finally, the user can save the results after pressing the **'Q'** shortcut key or the **'Close'** button. The user can use the vector line file named 'cal_fieldVal.shp' in the 'EditVector' folder to test the module's functionality which is calculating geometric features of surface ruptures.

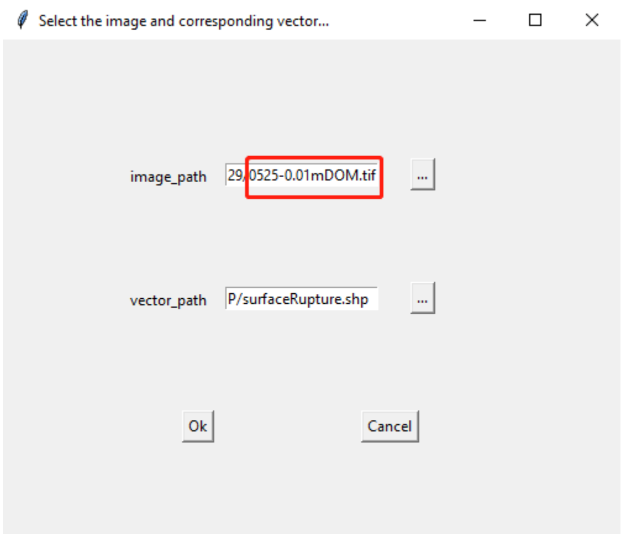
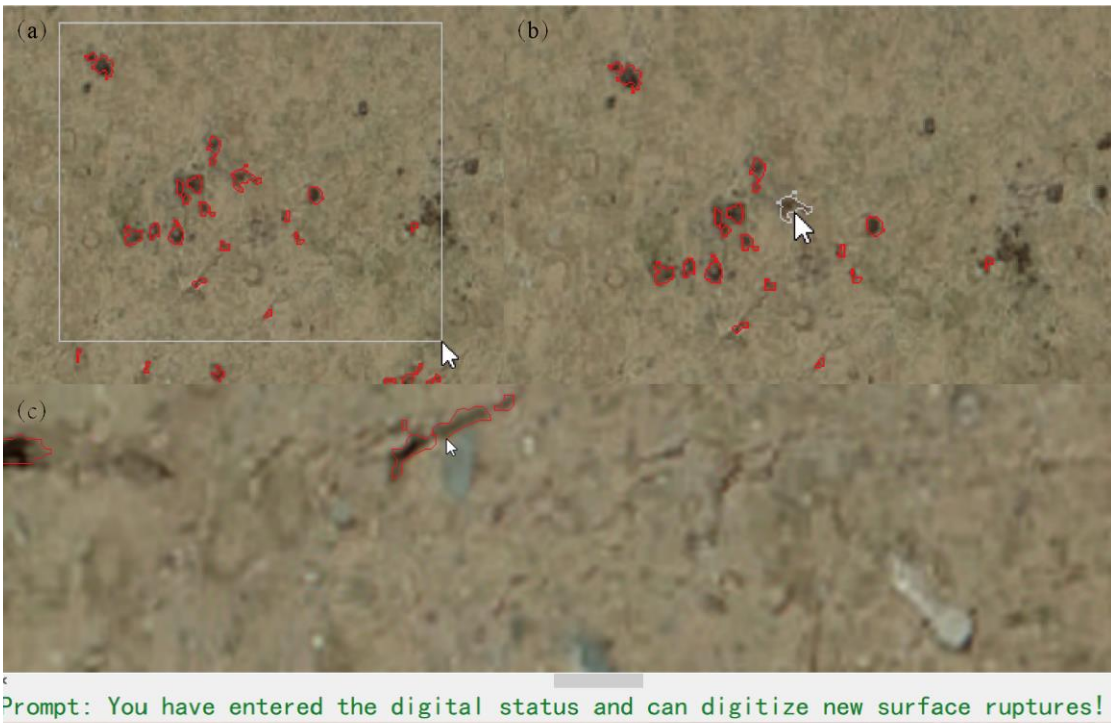


Figure 17 input data interface

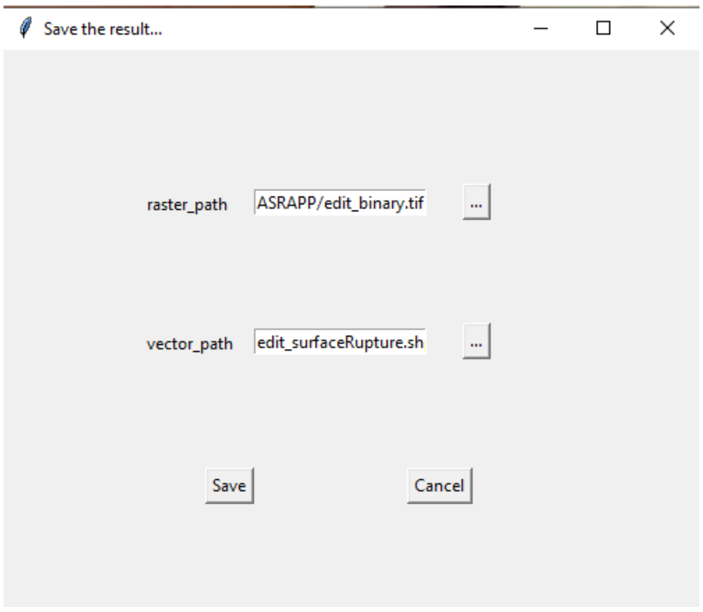
This module includes the following functions: (1) the user can move and zoom the image and vector objects at any time to see each vector object; (2) the user presses the '**left mouse**' button and drags to get the gray rectangle box in which the features are deleted when the '**left mouse**' button is lifted (Figure 18a); (3) Its color turns gray when the user moves the mouse over the feature, and then the user can press the '**right mouse**' button to delete it (Figure 18b); (4) the program will enter the digital state after the user presses the **'ctrl+d'** shortcut key. Then the user can use the **'left mouse'** button to draw the feature, and then press the **'enter'** shortcut key to complete the drawing of a feature. And the user can exit the digitizing state via the **'ctrl+alt+d'** shortcut key (Figure 18c); (5) In the process of (1), (2), (3) and (4), you can undo the operation by **'ctrl+z'** shortcut key, such as undoing the mistaken deletion, undeleting the wrong point in the digitizing process, etc.



Prompt: You have entered the digital status and can digitize new surface ruptures!

Figure 18 the functionalities of the 'EditVector' module

At the end of editing, press the **'Q'** shortcut or click the **'Close'** button to enter the result save window (Figure 19). Each edit can be saved to the vector layer and binary image. The binary image and the vector result are output in the .shp and .tif formats, respectively.


Save the result...

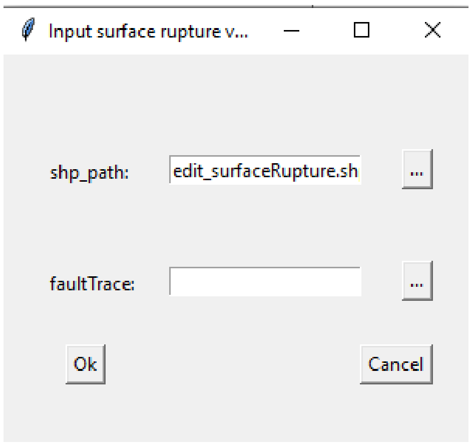
raster_path ASRAPP/edit_binary.tif

vector_path edit_surfaceRupture.sh

Figure 19 result save window

6 WidthAndStrike module

Step 1 The user clicks on the **'WidthAndStrike'** button in the main window to bring up the data input window (Figure 20). The user can choose whether or not to enter fault trace data (i.e., the dominant direction of the fault affecting that section of surface rupture). The fault trace data must be **linear vector data** in .shp format consistent with the original image coordinate system. We do not input fault trace data in this demo, but the uploaded data contains the data named 'fault_trace.shp', which the user can select for testing the program.

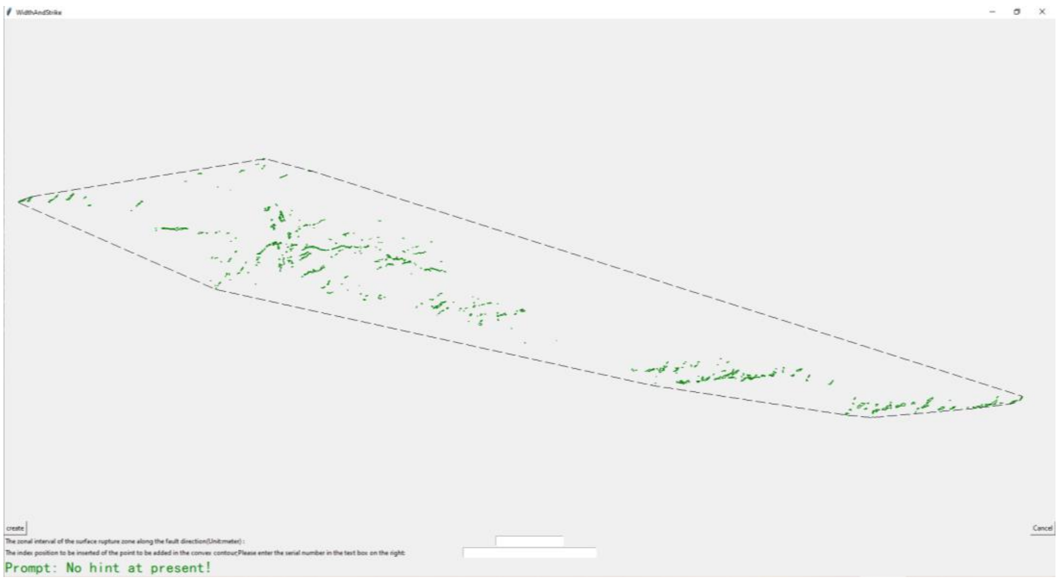

Input surface rupture v...

shp_path: edit_surfaceRupture.sh

faultTrace:

Figure 20 data input window

After clicking the **'Ok'** button, the application will automatically generate the minimum convex polygon surrounding the surface ruptures (Figure 21).



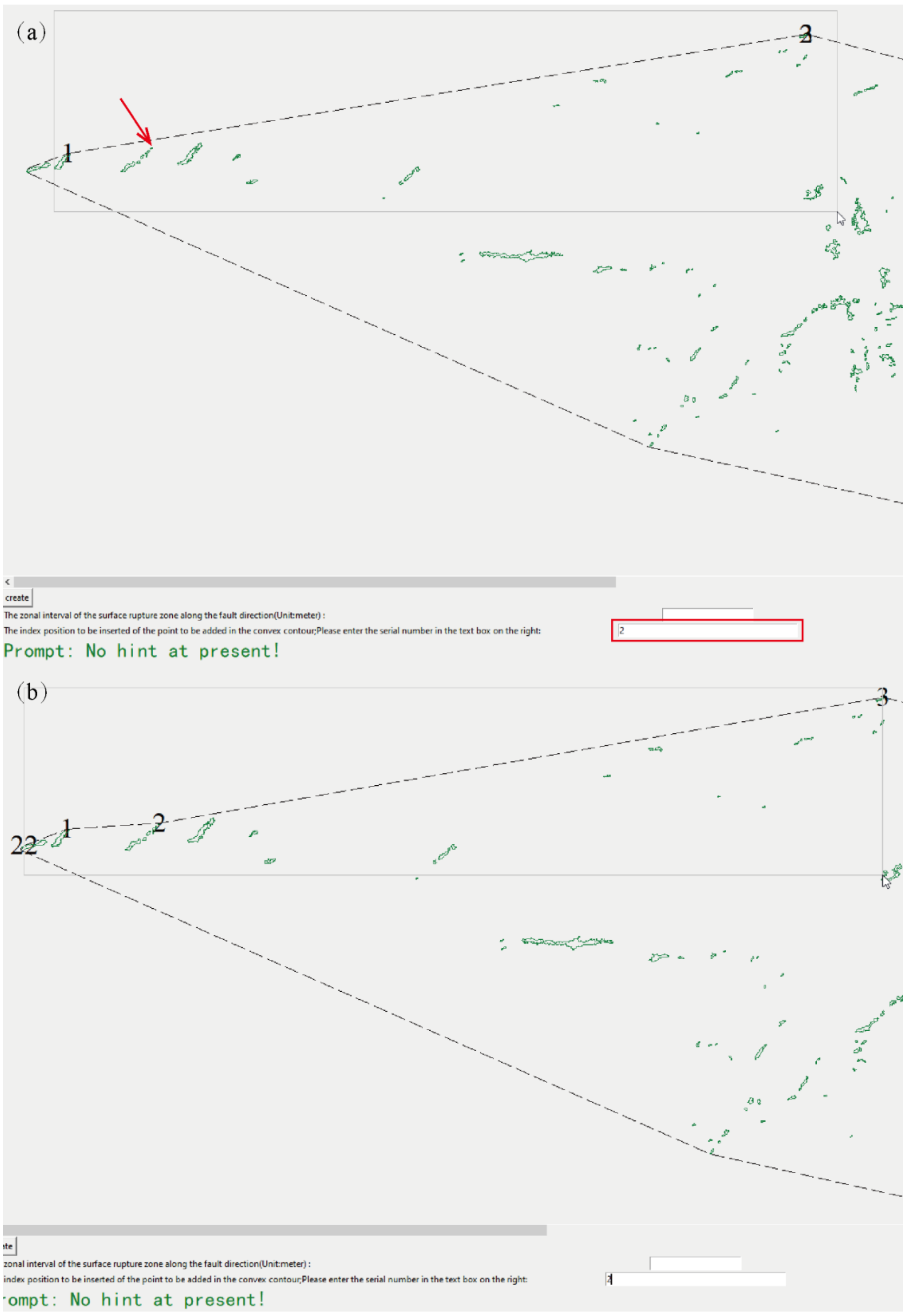
The zonal interval of the surface rupture zone along the fault direction(Unit:meter) :

The index position to be inserted of the point to be added in the convex contour,Please enter the serial number in the text box on the right:

Prompt: No hint at present!

Figure 21 surface ruptures and the convex polygon

Step 2 As in Figure 22a, the user presses the ‘**left mouse’** button and drags it to form a gray rectangular box. The order number of the inflection points of the outer contour line inside the rectangular box will be displayed. To insert a point at the red arrow position to form a new outsourced contour line, the user firstly needs to input '2' in the text box inside the red box (because the inflection points' serial numbers are clockwise), then presses the ‘**right mouse’** button and insert a new inflection point when the mouse approaches the surface rupture, and the program will automatically form a new outsourced contour line (Figure 22b). If the wrong inflection point is inserted, the operation can be undone by the **'ctrl+z'** shortcut key.

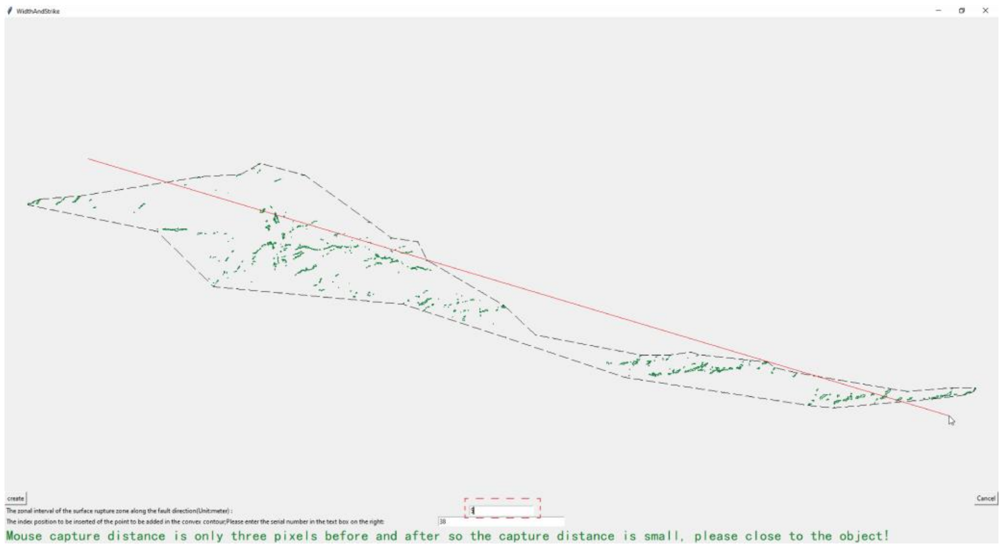


create
The zonal interval of the surface rupture zone along the fault direction(Unit:meter) :
The index position to be inserted of the point to be added in the convex contour;Please enter the serial number in the text box on the right: 2
Prompt: No hint at present!

te
zonal interval of the surface rupture zone along the fault direction(Unit:meter) :
index position to be inserted of the point to be added in the convex contour;Please enter the serial number in the text box on the right: 2
ompt: No hint at present!

Figure 22 Generate more accurate minimum envelope contours of surface ruptures.

Step 3 After several operations of inserting the inflection points, the gray convex outline of the package in Figure 23 was finally formed. The user draws the dominant direction (as shown by the red line in Figure 23) of the fault affecting surface ruptures at that location via the **'ctrl + left mouse'** event (Figure 23). The user can undelete the wrong selection point via the **'ctrl+alt+z'** shortcut key. Please note that: If the fault trace data is input in Step 1, it is no longer necessary to draw the dominant direction of the fault here.



The zonal interval of the surface rupture zone along the fault direction(Unit:meter) : 5

The index position to be inserted of the point to be added in the convex contour;Please enter the serial number in the text box on the right: 38

Mouse capture distance is only three pixels before and after so the capture distance is small, please close to the object!

Figure 23 Input the parameter and the dominant direction of the fault

Step 4 Then, input the distance interval (unit: meter) along the dominant direction of the fault in the red dashed box and click on the 'Create' button (Figure 23) to generate a bar graph of the width of the surface rupture zone (Figure 24a), a rose diagram of angles (Figure 24b) and a schematic diagram of the widths (Figure 24c). Small surface ruptures due to ground disturbance do not reflect geomechanical features. To obtain a geomechanically meaningful rose diagram of angles, the user may adjust the equivalent length threshold via a slider to remove the influence of small ruptures (Figure 24c). The bar graph and the rose map may be saved in the needed format. In addition, the user can modify the distance interval value several times to generate new results. The user can also enter different main directions of the fault multiple times to get new results. After pressing the ‘**ctrl+s**’ shortcut key, the schematic diagram of widths of the surface rupture is determined by the screenshot range, which is obtained by dragging the ‘**left mouse**’ button. Finally, the user can save the schematic diagram in the desired format.

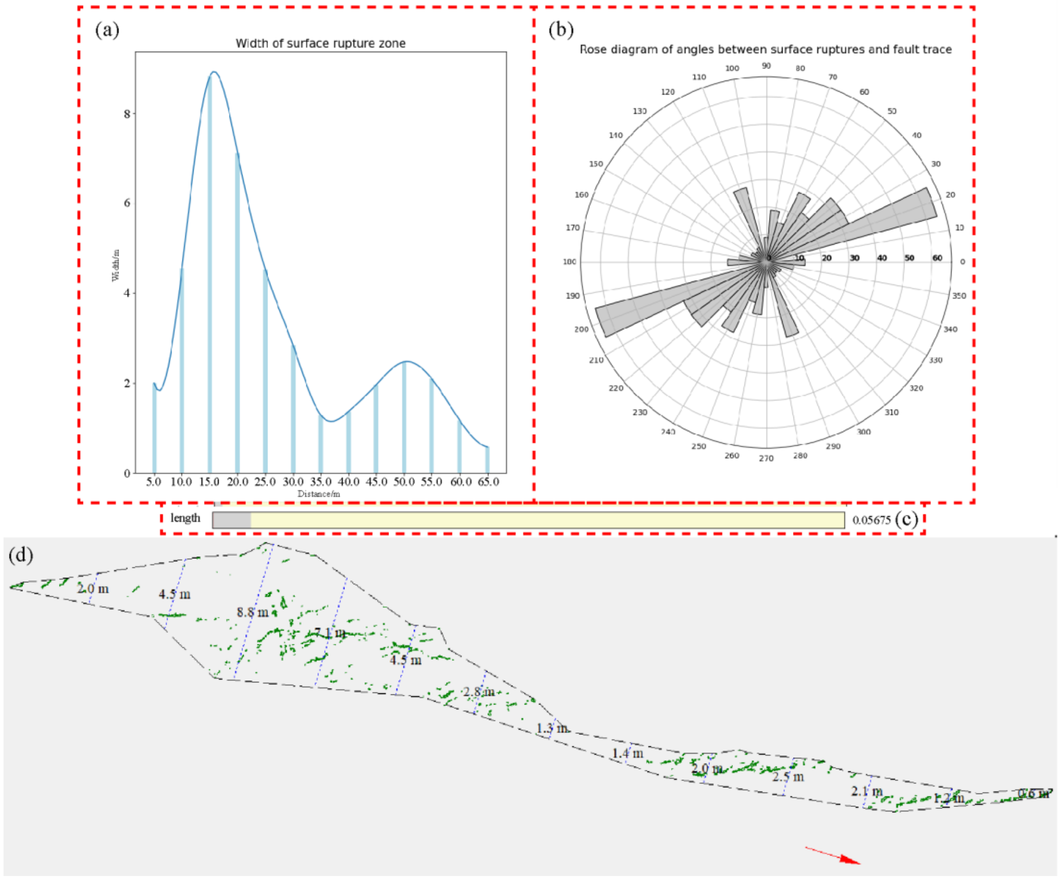


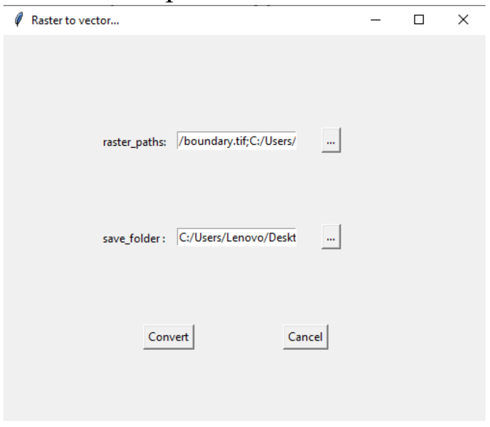
Figure 24 the generated results

7 Auxiliary tools available

7.1 Data structure conversion

The user can access data structure conversion tools via the **'raster to vector'** button, **'vector to raster'** button, **'lineRing to surface'** button, and '**surface to lineRing'** button from the main window.

(a) After clicking the **'raster to vector'** button, the user can choose to input one or multiple raster images and input the output folder (Figure 25). After clicking the **'Convert'** button, the input raster will be converted to the vector surfaces and the message "Conversion Complete" will be displayed at the end of the run. In this example, "boundary.tif" and "editout.tif" are input.


Raster to vector...

raster_paths: /boundary.tif;C:/Users/

save_folder : C:/Users/Lenovo/Deskt

Figure 25 input data and the output folder

The result of the conversion of the 'boundary.shp' file is shown in Figure 26.

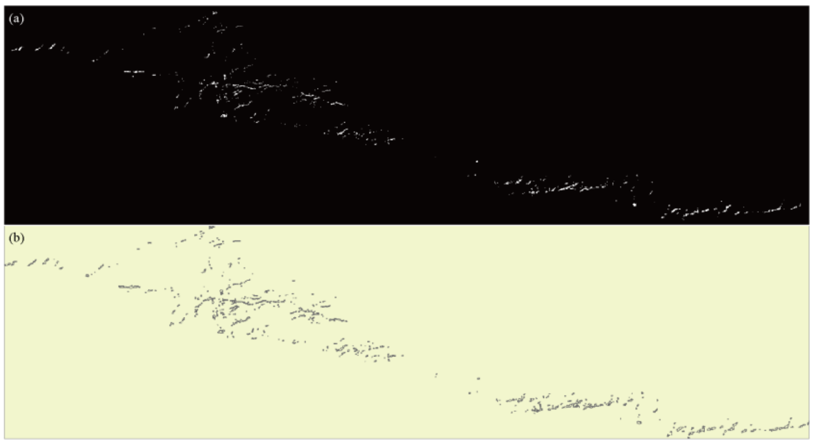
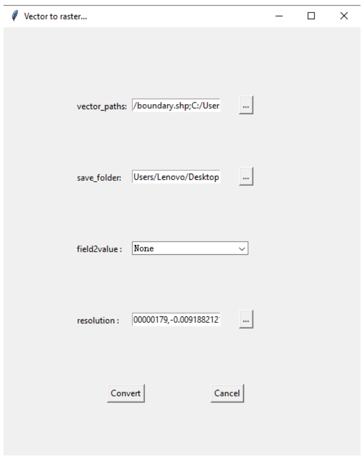


Figure 26 (a) the raster named "editout.tif"; (b) the vector result.

(b) After clicking the **'vector to raster'** button, the user can choose to input one or multiple raster images and input the output folder (Figure 27). In particular, the user can get the resolution in the form of 'ew, ns' ('ew' means east-west resolution and 'ns' means north-south resolution) by selecting the raster data of '.tif', or the user can fill in the text box with the resolution in the form of 'ew, ns'. In this example, the line vectors named 'boundary.shp' and 'editout.shp' are selected.



vector_paths: /boundary.shp;C:/User
save_folder: Users/Lenovo/Desktop
field2value : None
resolution : 00000179,-0.009188212

Figure 27 Input the parameters in the 'vector to raster' interface

The result of the conversion of the 'boundary.shp' file is shown in Figure 28.

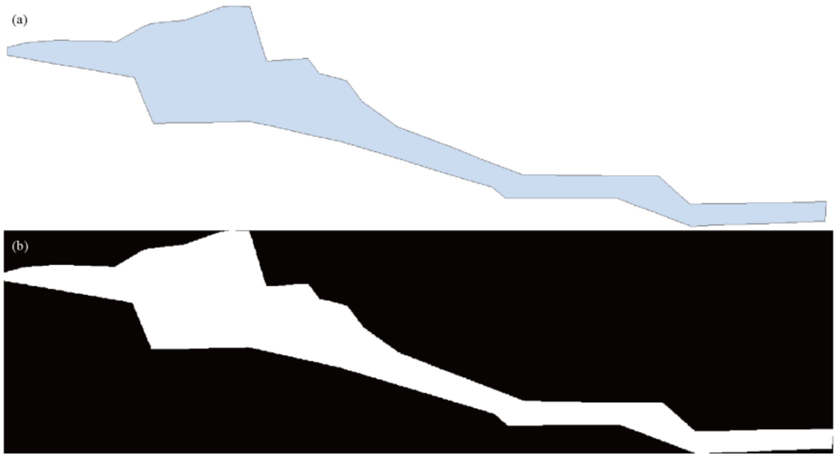
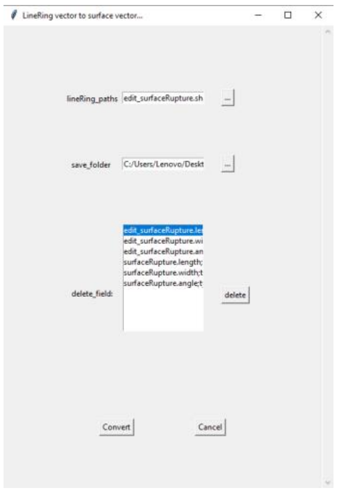


Figure 28 (a) the vector named 'boundary.shp'; (b) the raster result.

(c) After clicking the **'lineRing to surface'** button, the user can choose to input one or multiple line vector data and input the output folder (Figure 29). In addition, the user can delete unwanted fields via the **'delete'** button in the 'lineRing to surface' interface. In this example, 'edit_surfaceRupture.shp' and 'surfaceRupture.shp' are input.


LineRing vector to surface vector...

lineRing_paths: edit_surfaceRupture.sh

save_folder: C:/Users/Lenovo/Deskt

delete_field:
edit_surfaceRupture.le
edit_surfaceRupture.wi
edit_surfaceRupture.an
surfaceRupture.length;
surfaceRupture.width;t
surfaceRupture.angle;t

Figure 29 input desired parameters in the 'lineRing to surface' interface

As in Figure 30, the input is a line vector file and the output is a surface vector result.

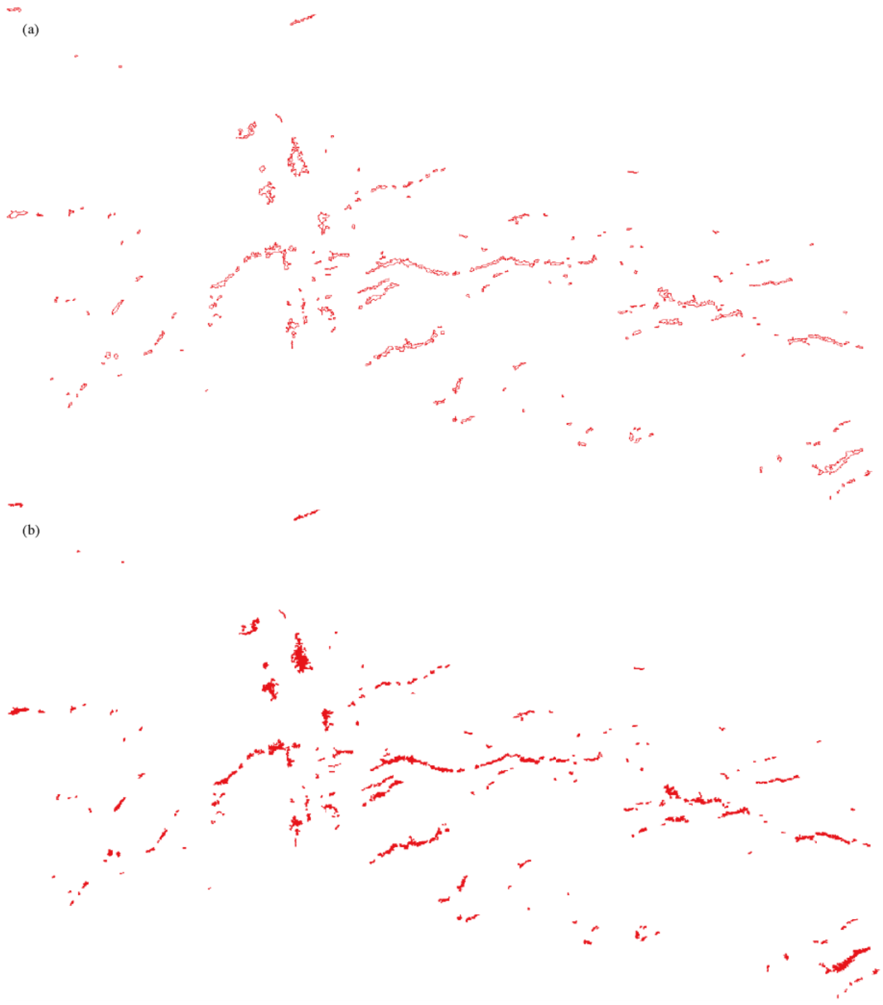
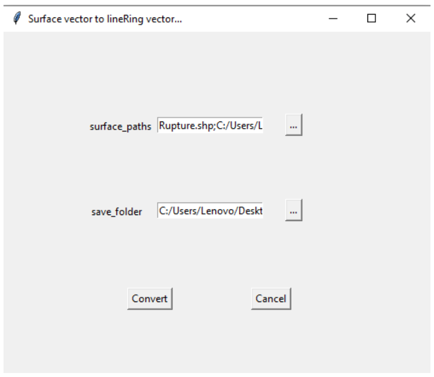


Figure 30 (a) the local graph of the line vector named 'edit_surfaceRupture.shp'; (b) the local graph of the generated results.

(d) After clicking the **'surface to lineRing'** button, the user can choose to input one or multiple surface vector data and input the output folder (Figure 31). In this example, the surface vector files named 'edit_surfaceRupture.shp' and 'surfaceRupture.shp' are input.



surface_paths Rupture.shp;C:/Users/L

save_folder C:/Users/Lenovo/Deskt

Figure 31 the input interface of the 'surface to lineRing' tool

As in Figure 32, the input is a surface vector file and the output is a line vector result.

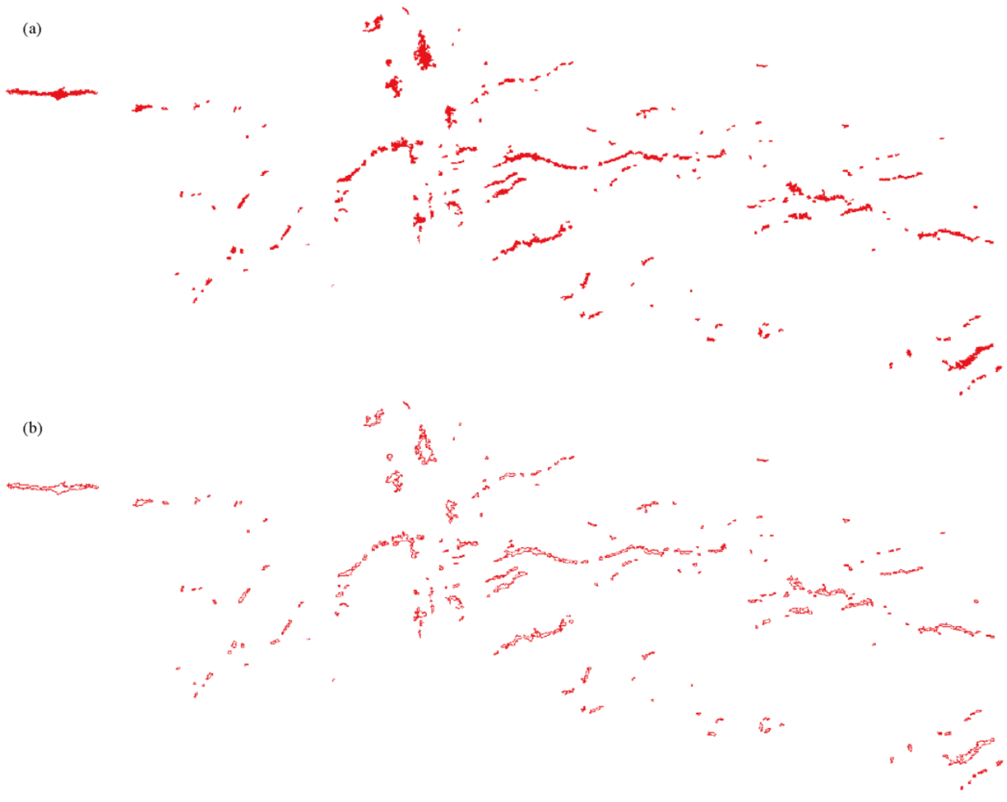
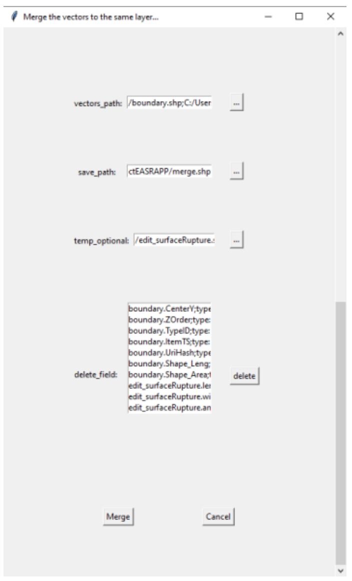


Figure 32 (a) the local graph of the surface vector named 'edit_surfaceRupture.shp'; (b) the local graph of the generated results.

7.2 Merge

This tool can be accessed via the **'Merge'** button. Figure 33 shows the GUI of this tool in which the user can choose to input the vector template to obtain a projection coordinate system for the result and remove unwanted fields. The tool acquires the projection coordinate system of the first vector of the input vectors as the projection coordinate system of the result if no vector template is selected. In addition, the same fields of input vectors will be merged, and fields that do not exist in some vectors will be filled with prescribed values in which only three prescribed values are defined currently (Please see Figure 8b in the paper).


Merge the vectors to the same layer...

vectors_path: /boundary.shp;C:/User

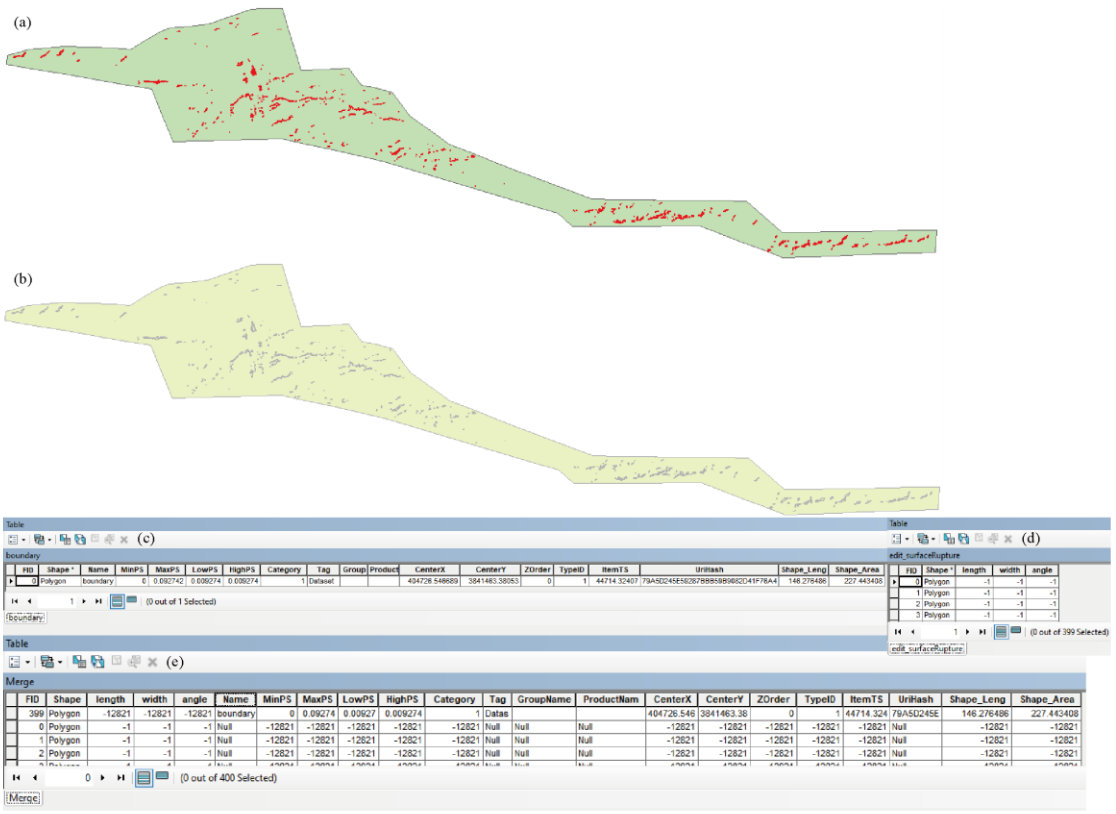
save_path: ctEASRAPP/merge.shp

temp_optional: /edit_surfaceRupture.

delete_field:
boundary.CenterY;type
boundary.ZOrder;type:
boundary.TypeID;type:
boundary.ItemTS;type:
boundary.UriHash;type
boundary.Shape_Leng;
boundary.Shape_Area;t
edit_surfaceRupture.le
edit_surfaceRupture.wi
edit_surfaceRupture.an

Figure 33 the GUI of the 'Merge' tool

In this example, 'boundary.shp' and 'edit_surfaceRupture.shp' are input, and the output is a merged surface vector (Figure 34a, b). In addition, Figure 34c, d are the fields of the input data, and Figure 34e is the attribute information of the result. Observing Figure 34c, d, e, the user can find that the values that exist in the input data still exist in the output, while the values that do not exist in the input are filled with the prescribed values in the output (text type field is filled with 'Null', and floating point field is filled with '-12821').



boundary

FID	Shape *	Name	MinPS	MaxPS	LowPS	HighPS	Category	Tag	Group	Product	CenterX	CenterY	ZOrder	TypeID	ItemTS	UriHash	Shape_Leng	Shape_Area
0	Polygon	boundary	0	0.092742	0.009274	0.009274	1	Dataset			404726.546689	3841463.38053	0	1	44714.32407	79A5D245E58287BBB59B9082D41F76A4	146.276486	227.443408

1 (0 out of 1 Selected)

boundary

edit_surfaceRupture

FID	Shape *	length	width	angle
0	Polygon	-1	-1	-1
1	Polygon	-1	-1	-1
2	Polygon	-1	-1	-1
3	Polygon	-1	-1	-1

1 (0 out of 399 Selected)

edit_surfaceRupture

Merge

FID	Shape	length	width	angle	Name	MinPS	MaxPS	LowPS	HighPS	Category	Tag	GroupName	ProductNam	CenterX	CenterY	ZOrder	TypeID	ItemTS	UriHash	Shape_Leng	Shape_Area
399	Polygon	-12821	-12821	-12821	boundary	0	0.09274	0.00927	0.009274	1	Datas			404726.546	3841463.38	0	1	44714.324	79A5D245E	146.276486	227.443408
0	Polygon	-1	-1	-1	Null	-12821	-12821	-12821	-12821	-12821	Null	Null	Null	-12821	-12821	-12821	-12821	-12821	Null	-12821	-12821
1	Polygon	-1	-1	-1	Null	-12821	-12821	-12821	-12821	-12821	Null	Null	Null	-12821	-12821	-12821	-12821	-12821	Null	-12821	-12821
2	Polygon	-1	-1	-1	Null	-12821	-12821	-12821	-12821	-12821	Null	Null	Null	-12821	-12821	-12821	-12821	-12821	Null	-12821	-12821

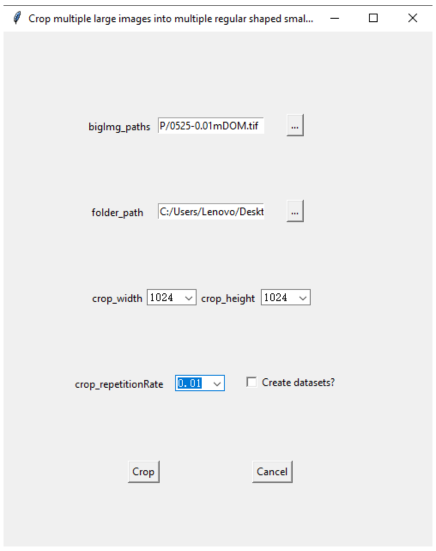
0 (0 out of 400 Selected)

Merge

Figure 34 the merged result

7.3 SlidingCrop

This tool can be accessed via the **‘SlidingCrop’** button. As in Figure 35, the user may set the crop repetition rate via the input box and select whether to create a sample dataset for machine learning or deep learning via the checkbox. If the checkbox is not checked, the size of the cropping results at the border may be different from the sliding window size (i.e. 'crop_width' and 'crop_height' in Figure 35), caused by the fact that the size of the cropped image is not necessarily an integer multiple of the sliding window. If it is checked, the small image of equal size is cut from the large image. In this example, the original image named ‘0525-0.01mDOM.tif’ is input.


Crop multiple large images into multiple regular shaped smal...

bigImg_paths P/0525-0.01mDOM.tif

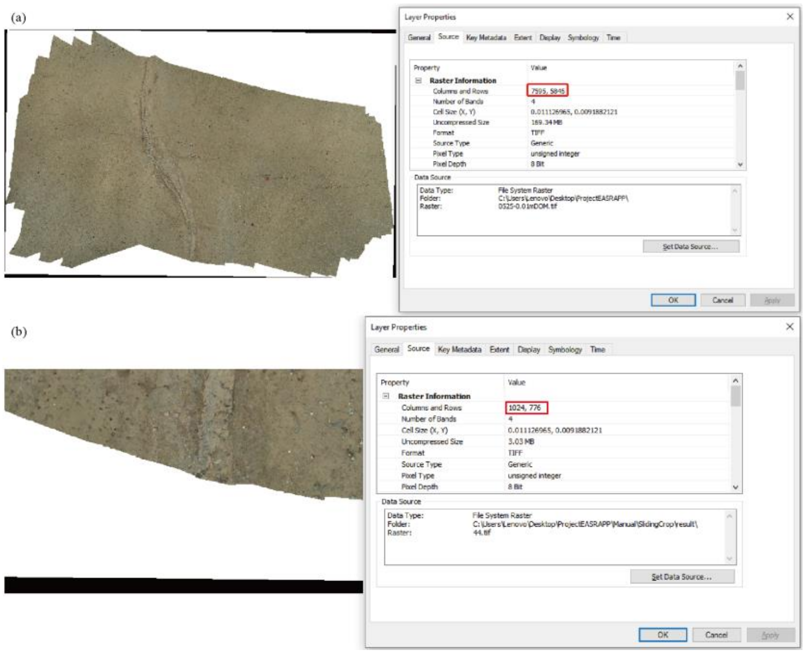
folder_path C:/Users/Lenovo/Deskt

crop_width 1024 crop_height 1024

crop_repetitionRate 0.01 ☐ Create datasets?

Figure 35 the GUI of the ‘slidingCrop’ tool.

Because the "Create dataset?" checkbox is not selected (Figure 35), some of the resulting cropped images are not 1024×1024 in size (the image size in Figure 36b is 1024×776).



(a)

Property	Value
Raster Information	
Columns and Rows	7595, 5845
Number of Bands	4
Cell Size (X, Y)	0.011126965, 0.0091882121
Uncompressed Size	169.34 MB
Format	TIFF
Source Type	Generic
Pixel Type	unsigned integer
Pixel Depth	8 Bit

Data Source

Data Type:	File System Raster
Folder:	C:\Users\Lenovo\Desktop\ProjectEASRAPP\
Raster:	0525-0.01mDOM.tif

(b)

Property	Value
Raster Information	
Columns and Rows	1024, 776
Number of Bands	4
Cell Size (X, Y)	0.011126965, 0.0091882121
Uncompressed Size	3.03 MB
Format	TIFF
Source Type	Generic
Pixel Type	unsigned integer
Pixel Depth	8 Bit

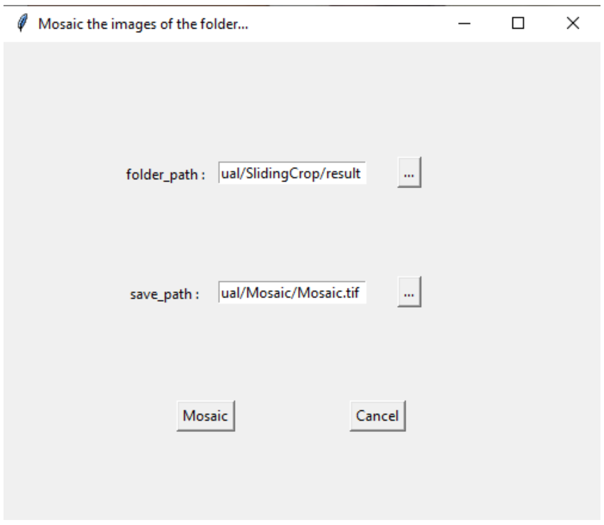
Data Source

Data Type:	File System Raster
Folder:	C:\Users\Lenovo\Desktop\ProjectEASRAPP\Manual\SlidingCrop\result\
Raster:	44.tif

Figure 36 (a) the input image; (2) one of the results.

7.4 Mosaic

This tool can be accessed via the **'Mosaic'** button in the main window (Figure 6). The user inputs the path of the image folder to be mosaicked and the resulting path in the input interface of the **'Mosaic'** tool (Figure 37). In this example, the test data input is the folder path of the result of the **'SlidingCrop'** tool cropping. Figure 38 shows the final mosaicing result.


Mosaic the images of the folder...

folder_path : ual/SlidingCrop/result

save_path : ual/Mosaic/Mosaic.tif

Figure 37 the input interface of the 'Mosaic' tool

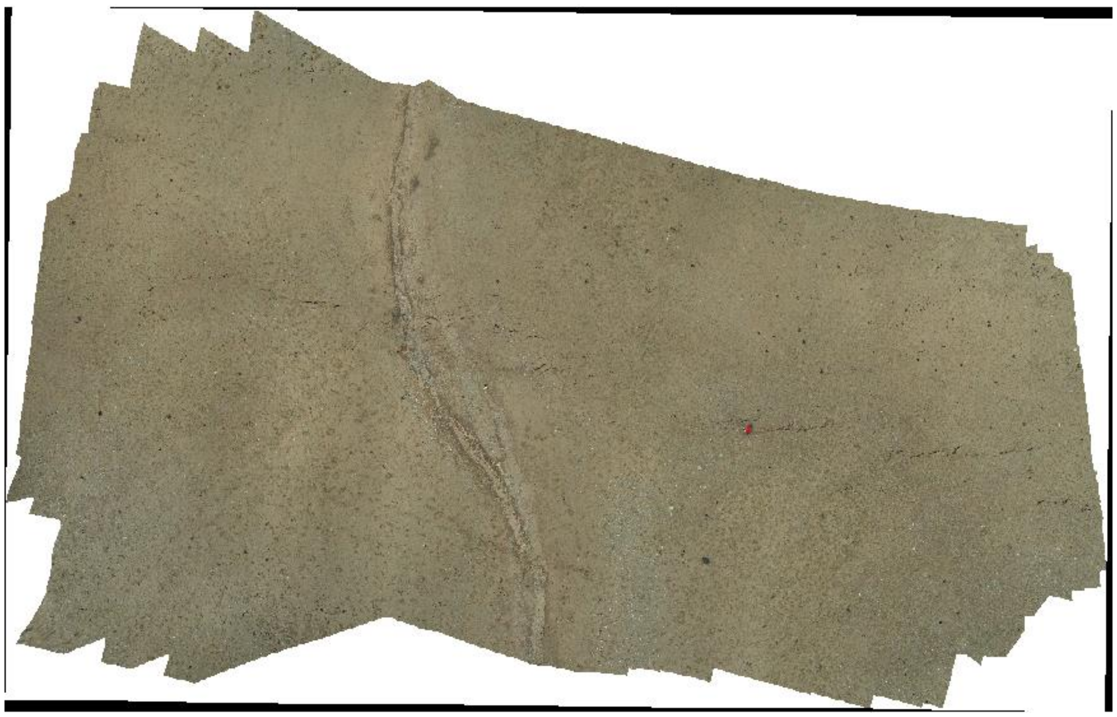


Figure 38 the mosaic result