Aguaje Detection – User Manual

Version 1

Table of Contents

Overview

Aguaje is a plugin that enables the identification of Aguaje palm trees in aerial images acquired by UAVs. This tool segment the Aguaje in RGB ortho-mosaics.

CHAPTER 1: QGIS CONFIGURATION

To install Aguaje Detection plugin, we must install python libraries in QGIS.

QGIS version required: QGIS 3.4 (or the latest stable version)

Required Libraries:

The following list of libraries have been used in the plugin development.

- fiona=1.8.11

- keras=2.2.4

- rasterio==1.0.28

- scikit-learn==0.21.3

- tensorflow=1.13.1

- tqdm==4.36.1

- opencv-python

- opencv-contrib-python

Libraries Installation:

1. Install 3.4 QGIS (https://qgis.org/en/site/forusers/download.html).
2. After QGIS installation, in the search bar look for 'OsGeo4W Shell', then right click and choose Open File Location, the location folder will be opened. (Fig.1)

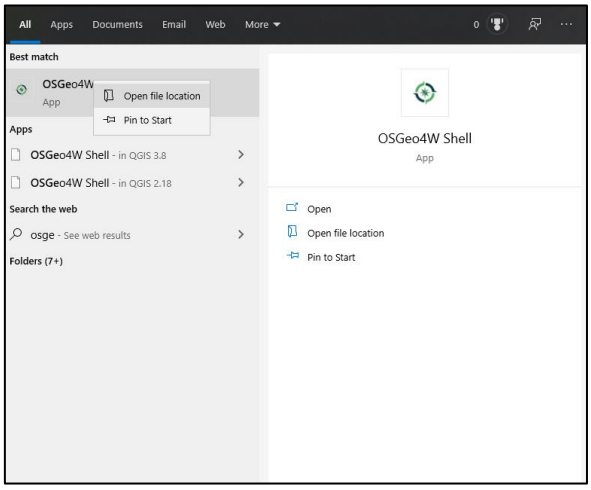
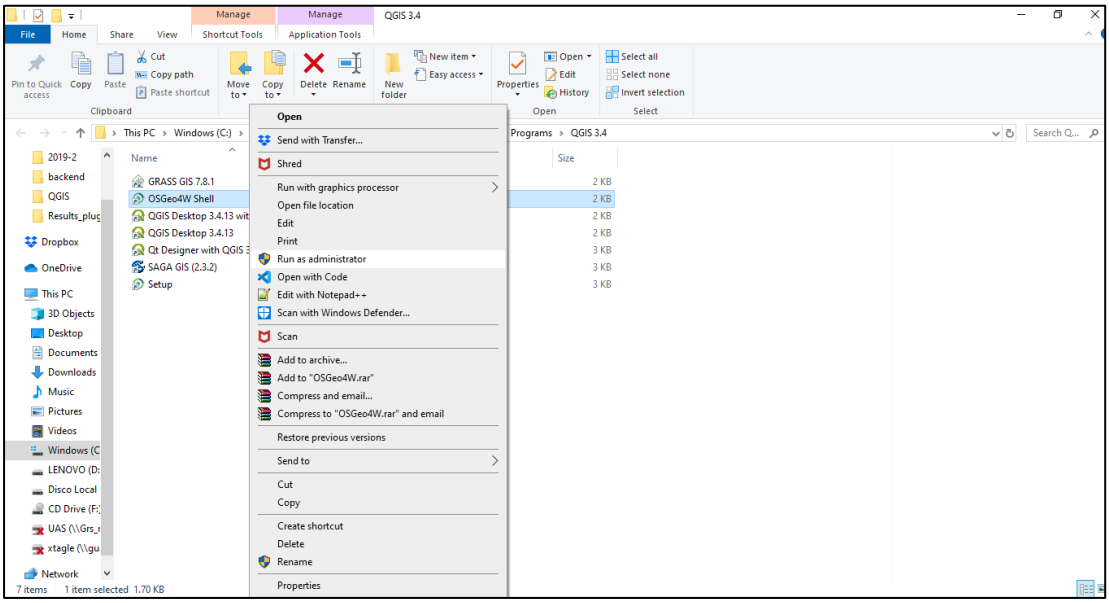


Fig. 1

3. Right click on the application ‘OsGeo4W Shell’ and, choose the option ‘run as administrator’. A windows console will be opened. (Fig. 2)



4. The environment Python 3 should be activated, therefore “py3_env” should be written in the command line.

```
C:\Windows\System32>py3_env
```

5. To check if the environment ‘Python 3’ is activated, we should write “python” in the command line, the version ‘3.X.X’ will appear.

```
C:\Windows\System32>python
Python 3.7.0 (v3.7.0:1bf9cc5093, Jun 27 2018, 04:59:51) [MSC v.1914 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()
```

6. To verify if the *pip* library is working, we should write “python -m ensurepip --default-pip”

```
C:\Windows\System32>python -m ensurepip --default-pip
```

7. All the library packets listed above should be installed. It should be added the following expressions:

- python -m pip install h5py==2.9.0
- python -m pip install tensorflow==1.13.1
- python -m pip install keras==2.2.4
- python -m pip install psutil==5.6.3
- python -m pip install scikit-learn==0.21.3

- python -m pip install tqdm==4.36.1
- python -m pip install opencv-python
- python -m pip install opencv-contrib-python

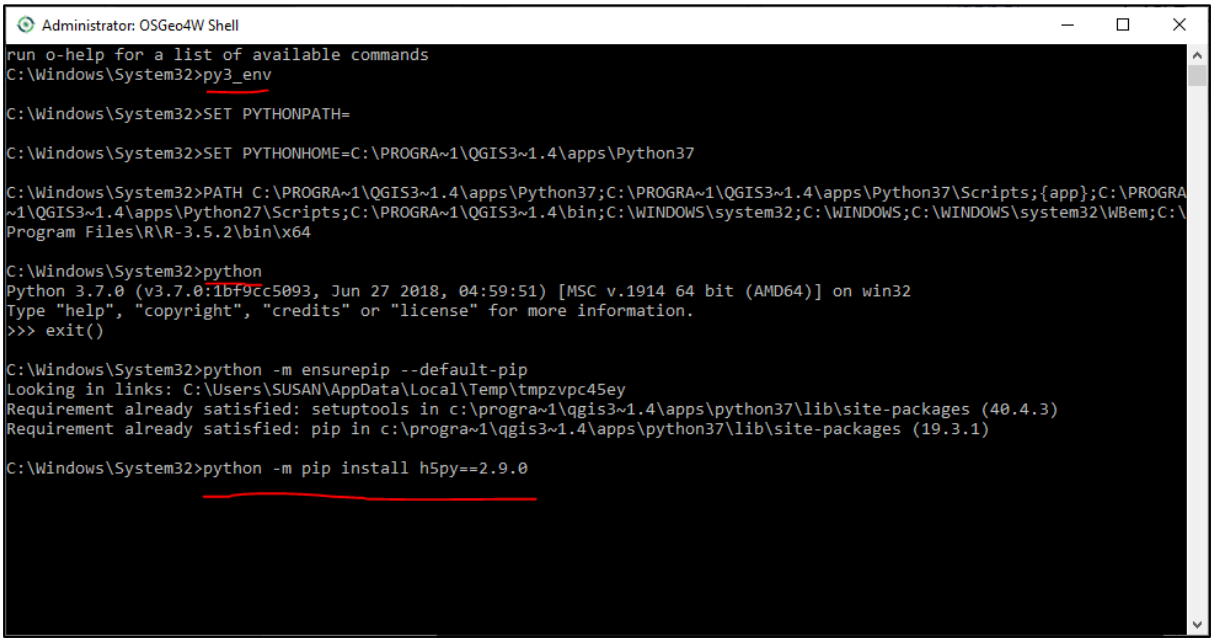


Fig. 3 Commands for installing

8. Fiona and Rasterio libraries should be downloaded to install. These libraries are attached to this document and are compatible with Python 3.7. To install the libraries the full path location should be specified.

   - python -m pip install C:\Users\SUSAN\qgispackages\Fiona-1.8.11-cp37-cp37m-win_amd64.whl
   - python -m pip install C:\Users\SUSAN\qgispackages\rasterio-1.0.28-cp37-cp37m-win_amd64.whl

9. If QGIS is working with a different python version 3.X, the packets should be downloaded from the link (https://www.lfd.uci.edu/~gohlke/pythonlibs/).
   - cp3X-cp3X points out the Python version.

Fiona, OGR's neater API.
Requires GDAL.
Fiona-1.8.13-cp27-cp27m-win32.whl
Fiona-1.8.13-cp27-cp27m-win_amd64.whl
Fiona-1.8.13-cp35-cp35m-win32.whl
Fiona-1.8.13-cp35-cp35m-win_amd64.whl
Fiona-1.8.13-cp36-cp36m-win32.whl
Fiona-1.8.13-cp36-cp36m-win_amd64.whl
Fiona-1.8.13-cp37-cp37m-win32.whl
Fiona-1.8.13-cp37-cp37m-win_amd64.whl
Fiona-1.8.13-cp38-cp38-win32.whl
Fiona-1.8.13-cp38-cp38-win_amd64.whl

Fig. 4

Rasterio provides clean and fast and geospatial raster I/O.
Requires gdal.

rasterio-1.0.24+gdal24-cp35-cp35m-win32.whl
rasterio-1.0.24+gdal24-cp35-cp35m-win_amd64.whl
rasterio-1.0.24+gdal24-cp36-cp36m-win32.whl
rasterio-1.0.24+gdal24-cp36-cp36m-win_amd64.whl
rasterio-1.0.24+gdal24-cp37-cp37m-win32.whl
rasterio-1.0.24+gdal24-cp37-cp37m-win_amd64.whl
rasterio-1.0.28-cp27-cp27m-win32.whl
rasterio-1.0.28-cp27-cp27m-win_amd64.whl
rasterio-1.0.28-cp35-cp35m-win32.whl
rasterio-1.0.28-cp35-cp35m-win_amd64.whl
rasterio-1.0.28-cp36-cp36m-win32.whl
rasterio-1.0.28-cp36-cp36m-win_amd64.whl
rasterio-1.0.28-cp37-cp37m-win32.whl
rasterio-1.0.28-cp37-cp37m-win_amd64.whl
rasterio-1.0.28-cp38-cp38-win32.whl
rasterio-1.0.28-cp38-cp38-win_amd64.whl

Fig. 5

CHAPTER 2: PLUGIN INSTALLATION

1. Download and decompress the file: **aguaje_detection_processing.**

2. Open QGIS and click *Settings* in the QGIS menu, select *User profiles* and select *Open Active Profile Folder* (Fig. 6). A folder will appear.

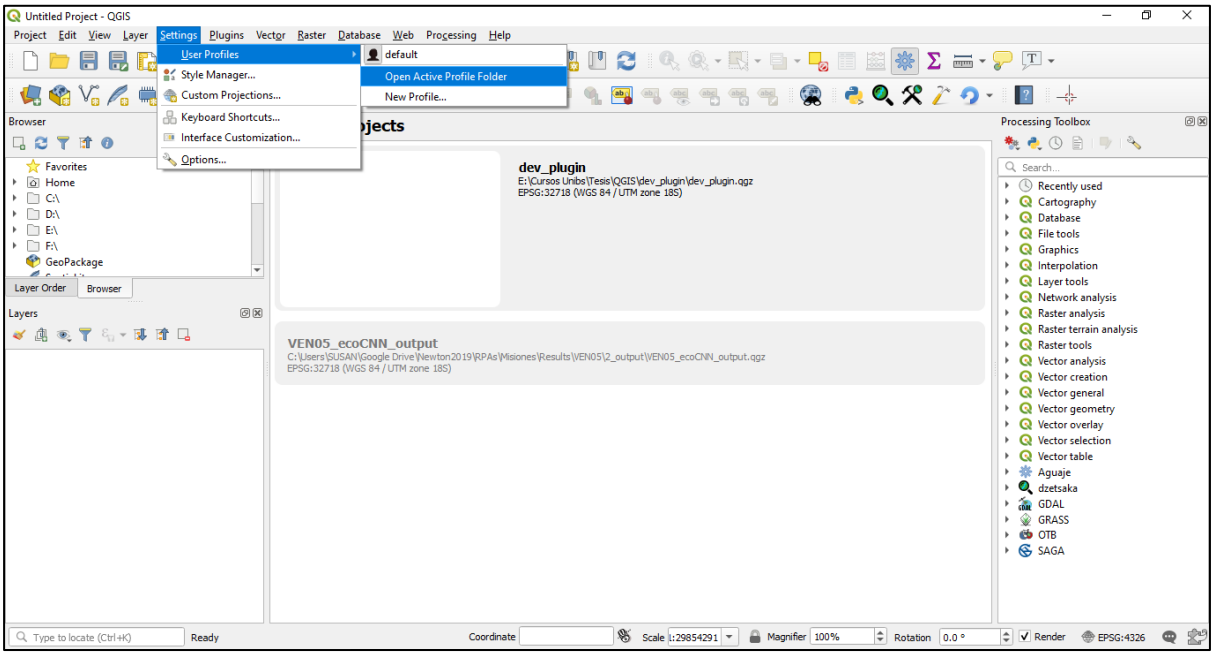


Fig. 6

3. Open *python* folder (Fig. 7), select *plugins* folder (Fig. 8), copy **aguaje_detection_processing** folder (Fig.9), and restart QGIS.

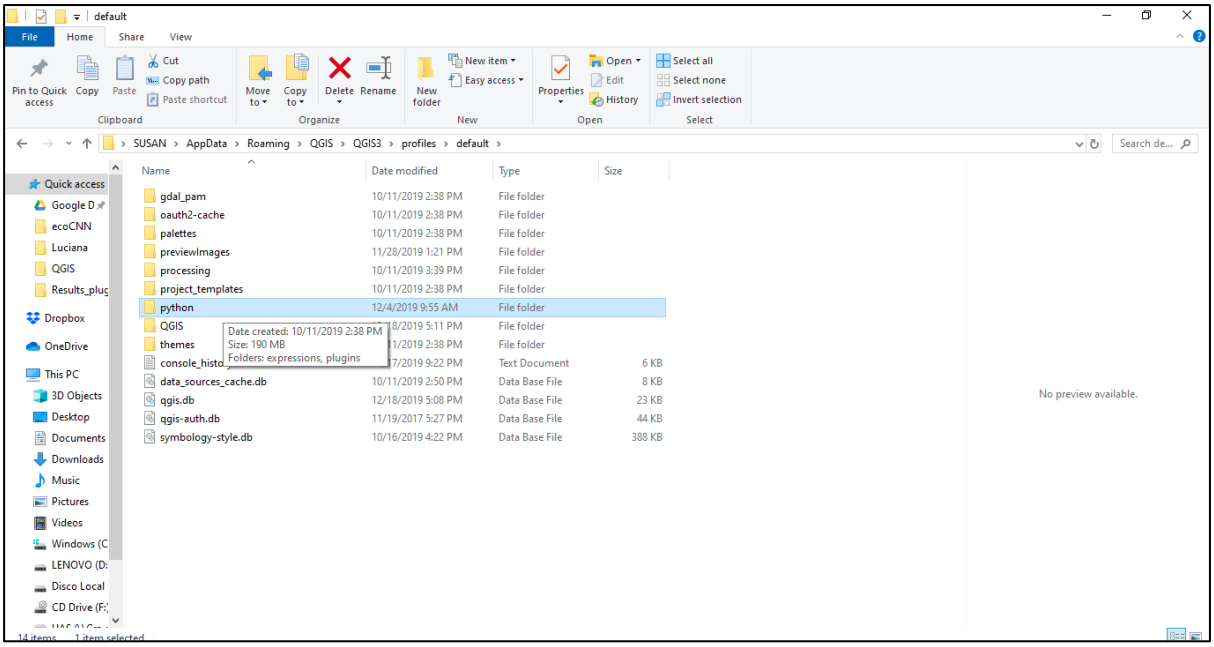


Fig. 7

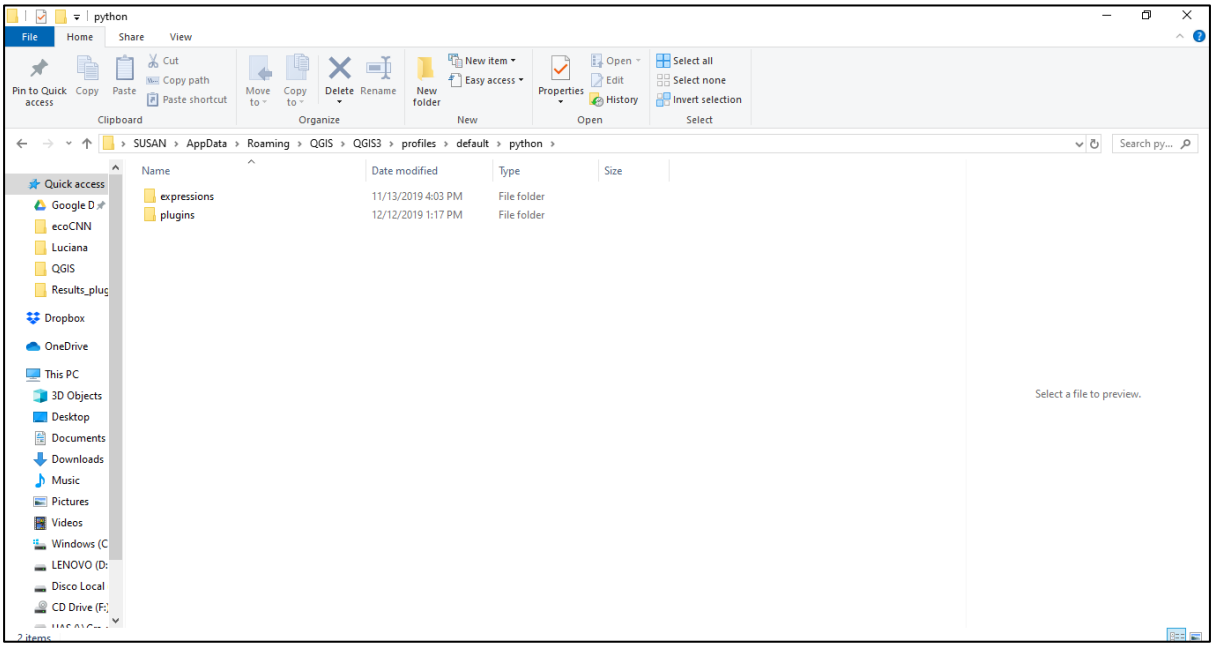


Fig. 8

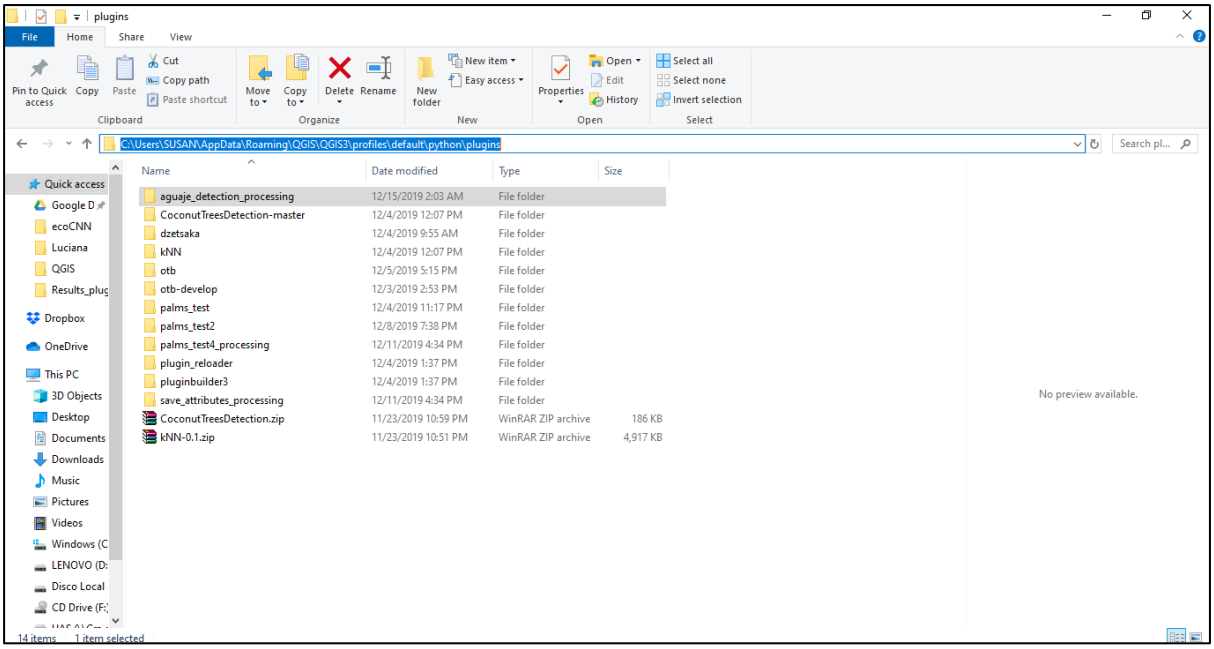


Fig. 9

4. Click *Processing* menu and choose *Toolbox* option (Fig. 10). The *Processing toolbox* window will appear on the right side in the QGIS environment.

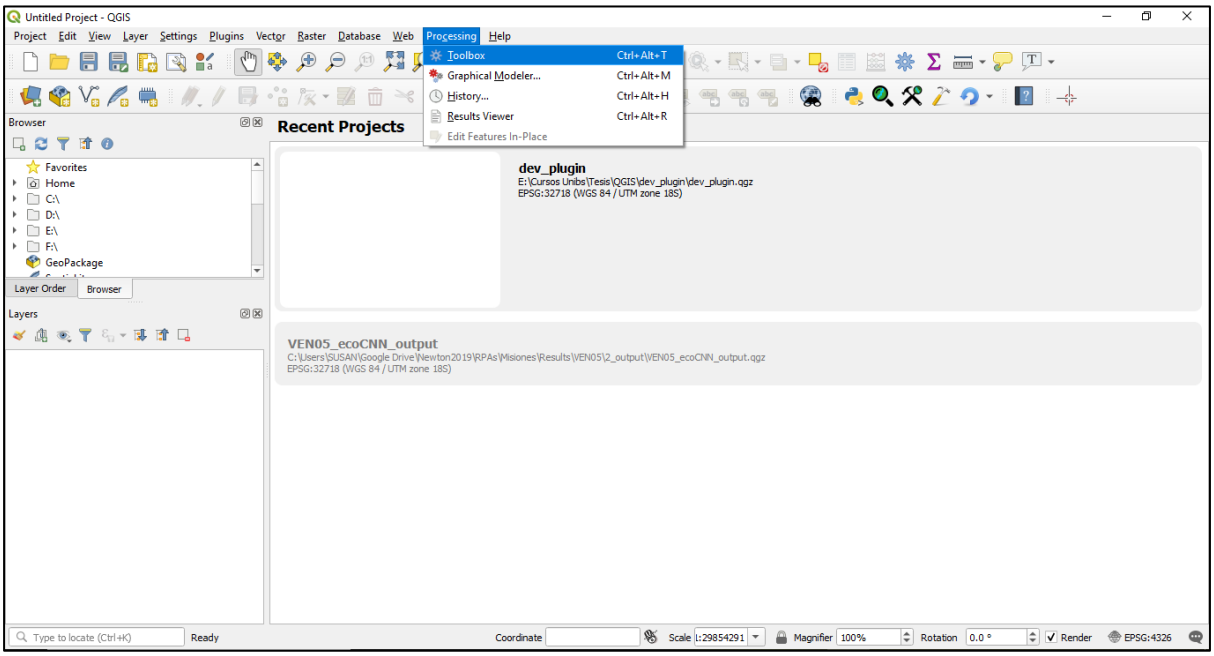


Fig. 10

5. Click on the *Plugin* menu in QGIS, select *Manage and install plugins* option (Fig. 11). A *Plugins* window should be opened.

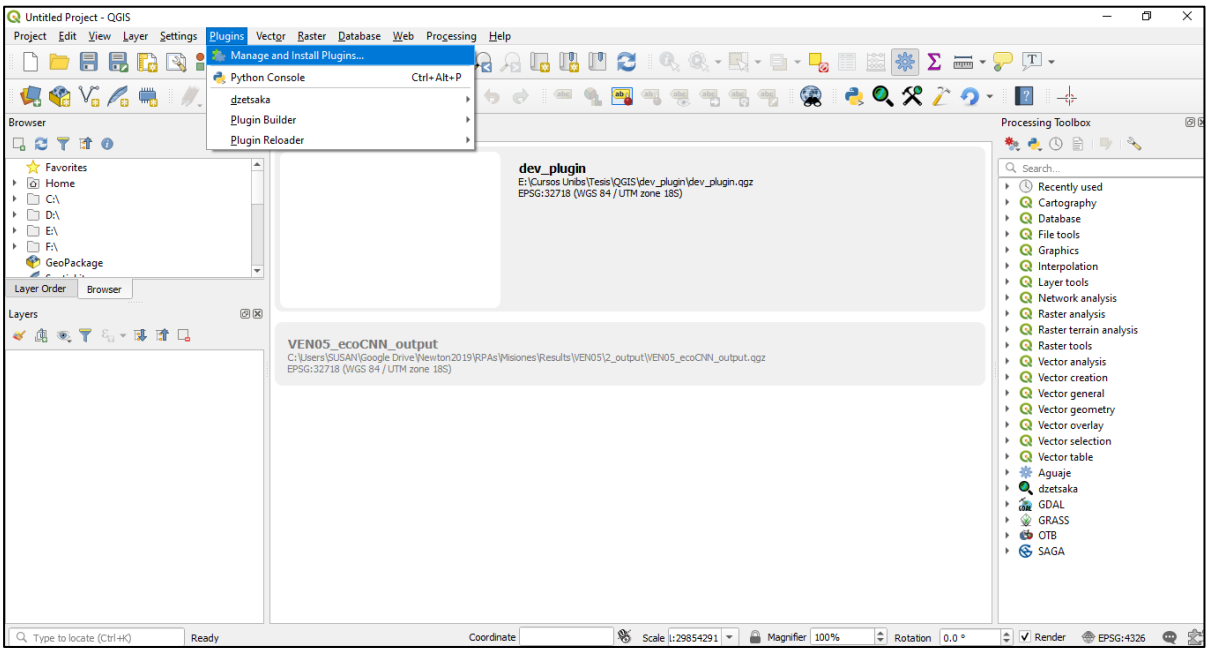


Fig. 11

6. Select on the right side of the Plugins window *Installed* option (Fig. 12).

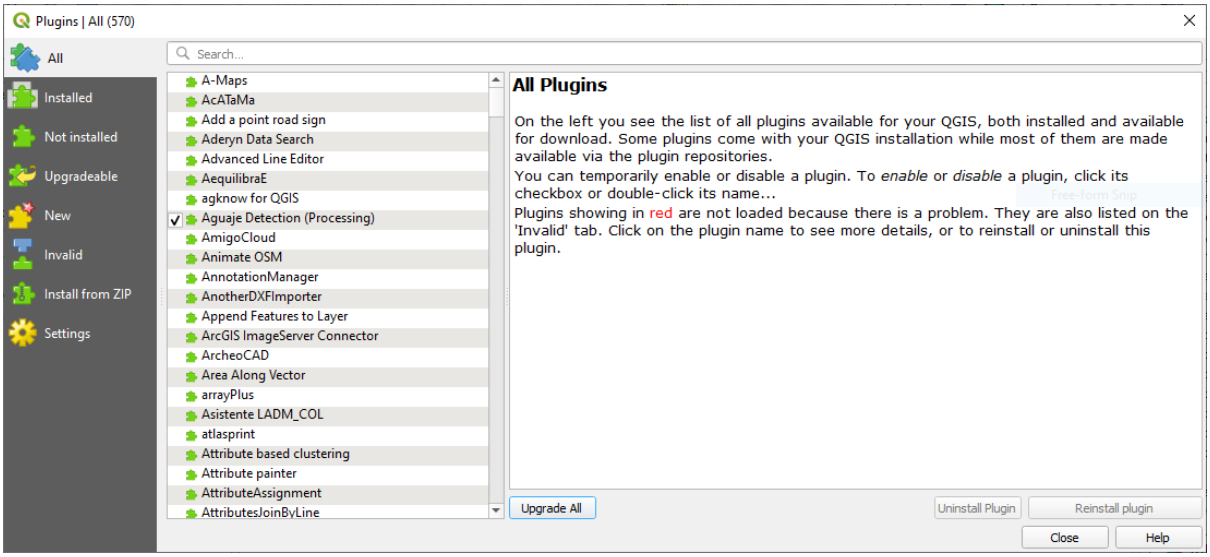


Fig. 12

7. Check the *Aguaje Detection (Processing)* box (Fig. 13) and close the window.

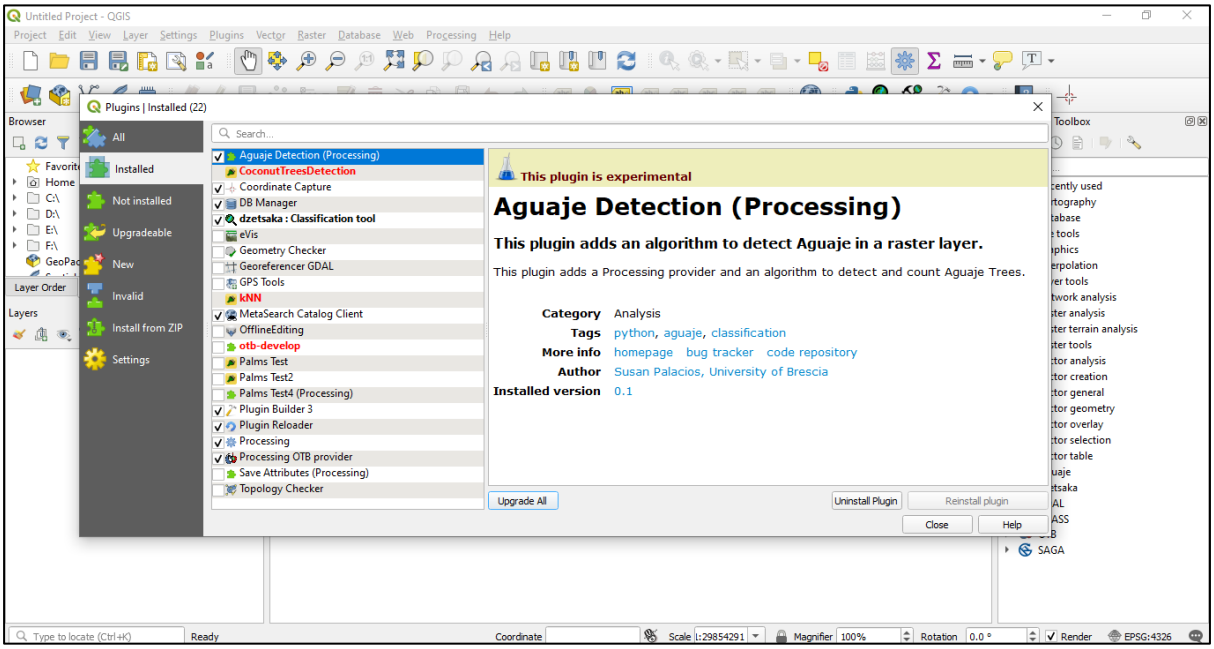


Fig. 13

8. The plugin will appear in the Processing Toolbox. (Fig. 14)

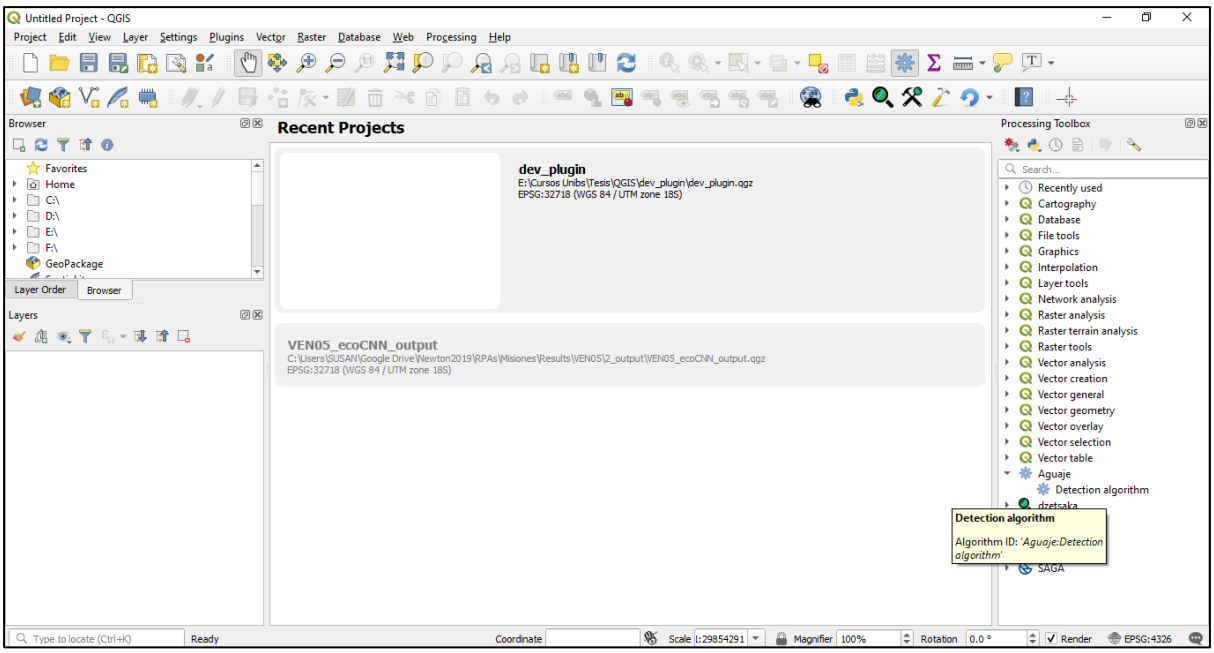


Fig. 14

CHAPTER 3: PLUGIN PARAMETERS

The plugin parameters used on the detection procedure.

- **Input raster**
  It allows to select an image in RGB color.
- **Output raster**
  It allows to save the result images.
- **Log**
  It shows processing information and the status.

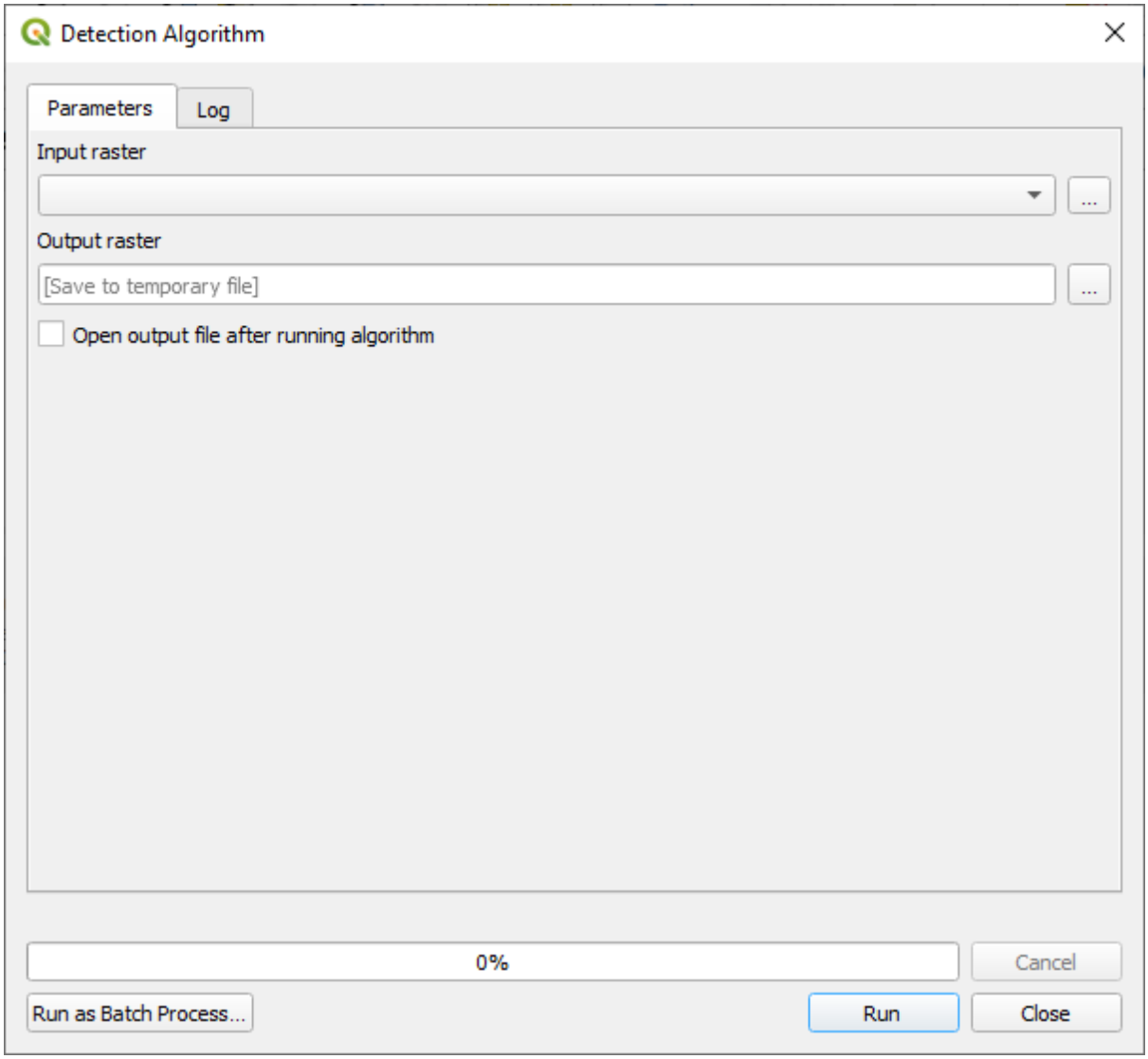


Fig. 15

CHAPTER 4: WORKFLOW

1. Load the RGB image to segment.

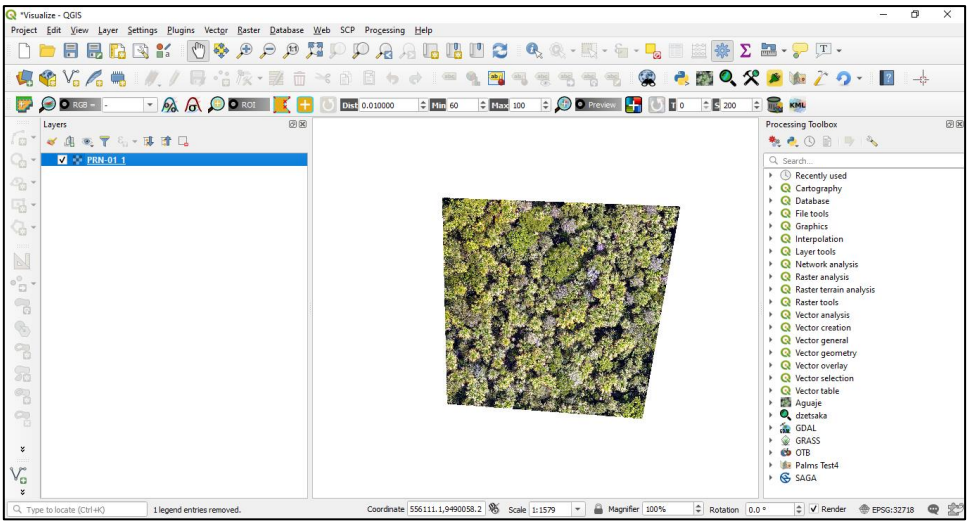


Fig. 16

2. Click  *Python console* icon on the QGIS Toolbars. A Python Console panel will be opened (Fig. 17)

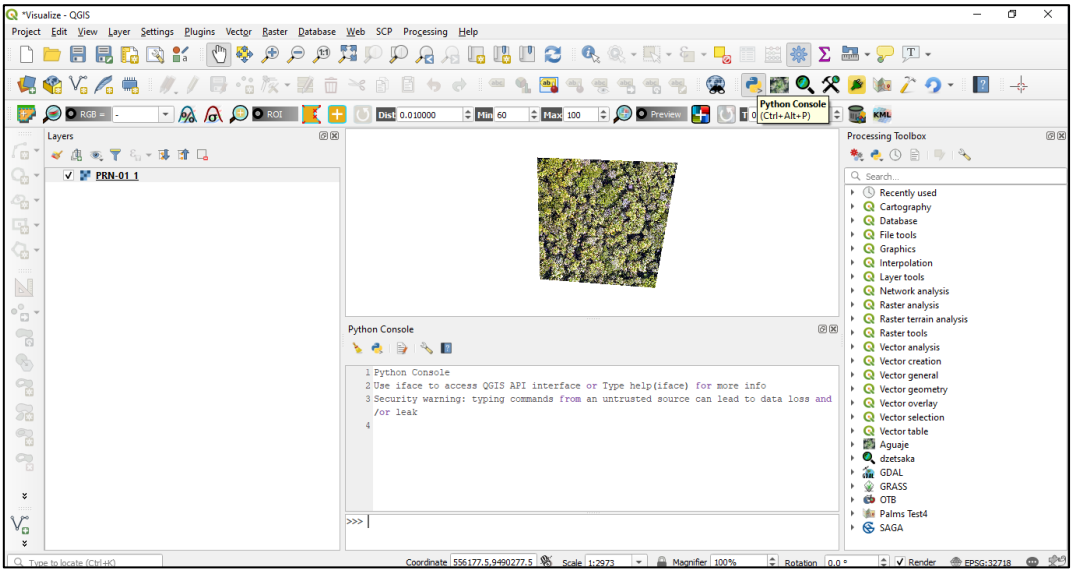
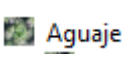


Fig. 17

3. Click  *Aguaje* icon in the Processing Toolbox and select *Detection algorithm* (Fig. 18)

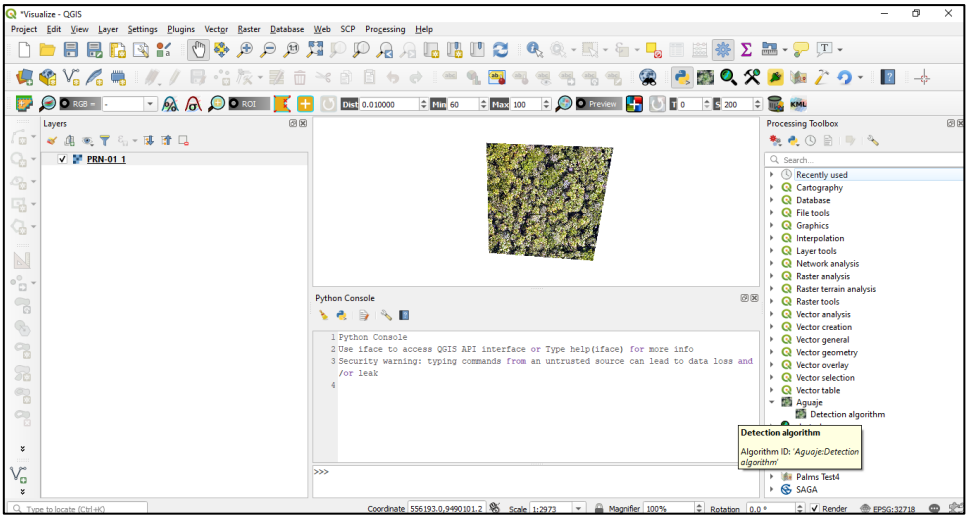
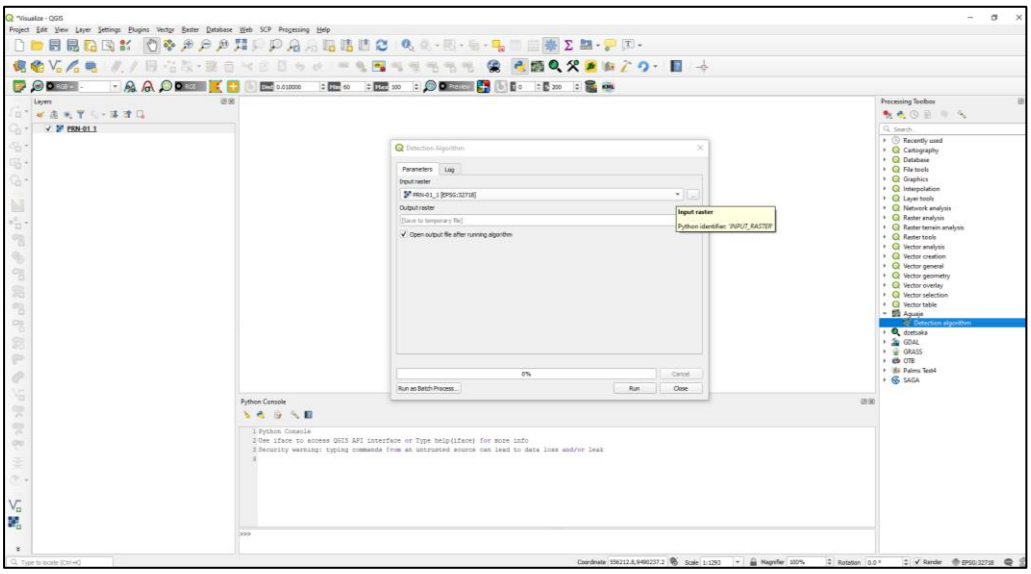


Fig. 18

4. Select the image in the Input Raster.



5. Click  to Save the *Output Raster.* It is recommended use the *Save to file* option, because the algorithm will create data in the location given that will be removed later. (Fig.20).

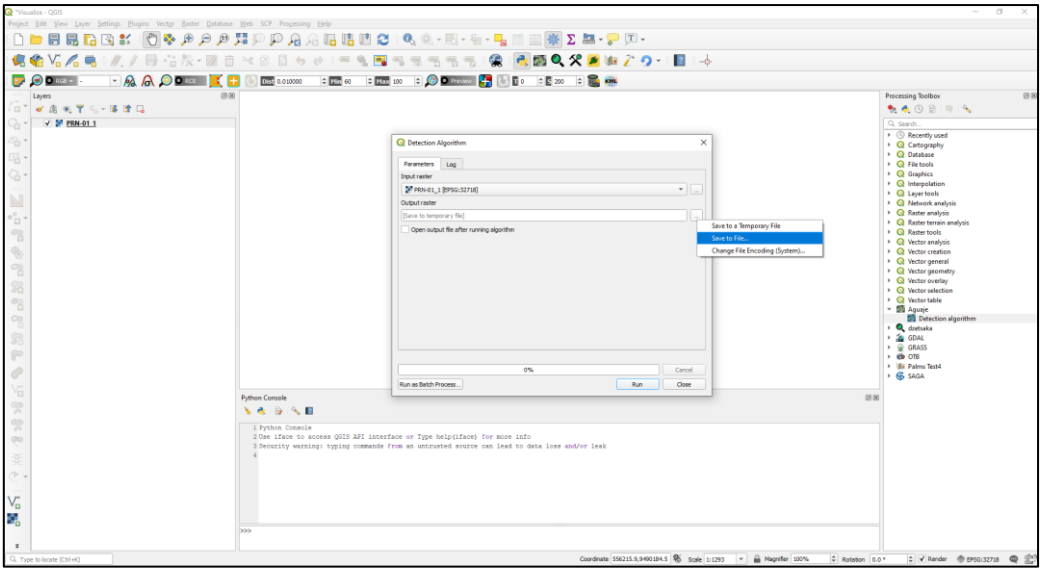


Fig. 20

6. Browse the location and write a name to save the result. Click *Save* button, the result will be saved in .TIF format (Fig.21).

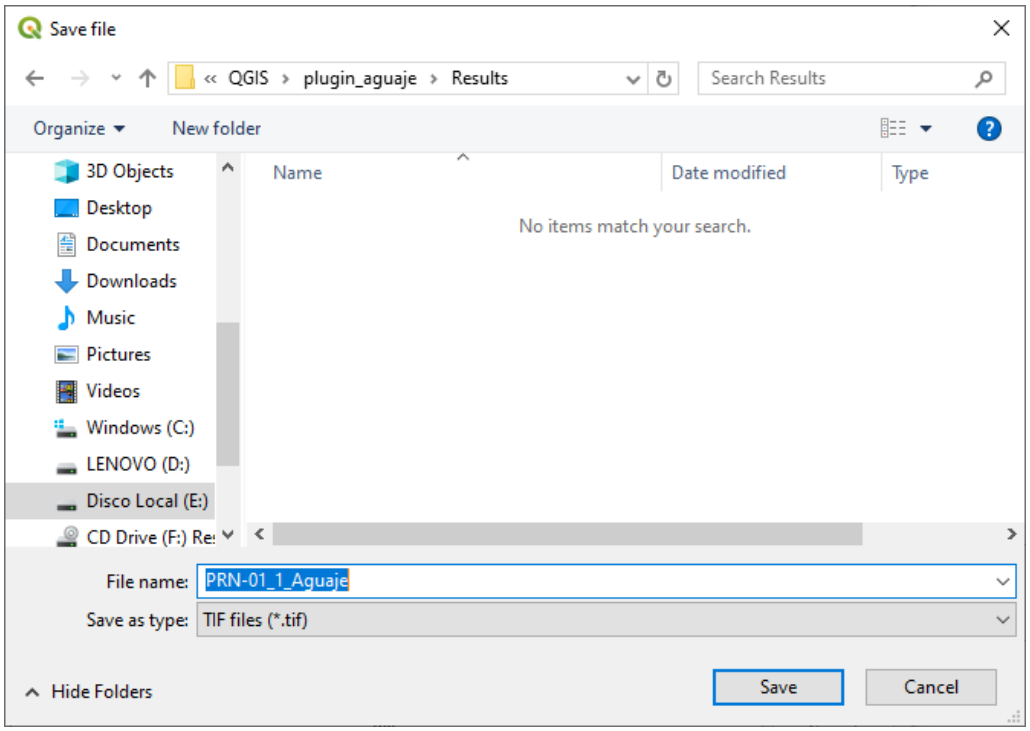


Fig. 21

7. In the plugin window it can be observed that all the fields are filled. The option *Open output file after running algorithm* should be checked to load the image segmented in the QGIS environment (Fig.22).

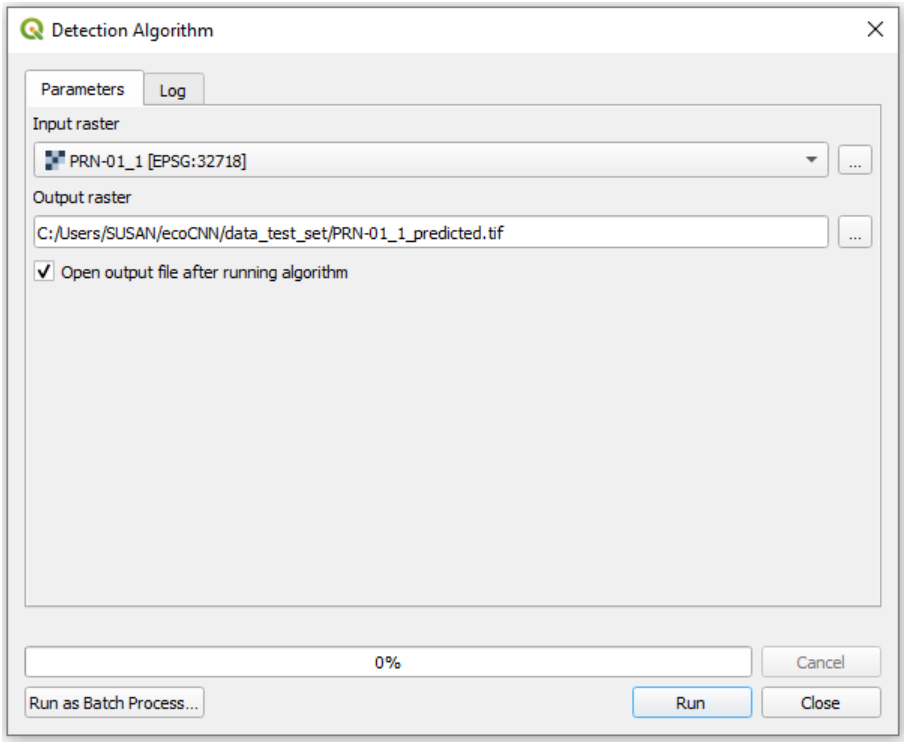


Fig. 22

8. Click the *Run* button, the Log windows will show the processing status (Fig. 23) and if the algorithm finished (Fig. 24) .

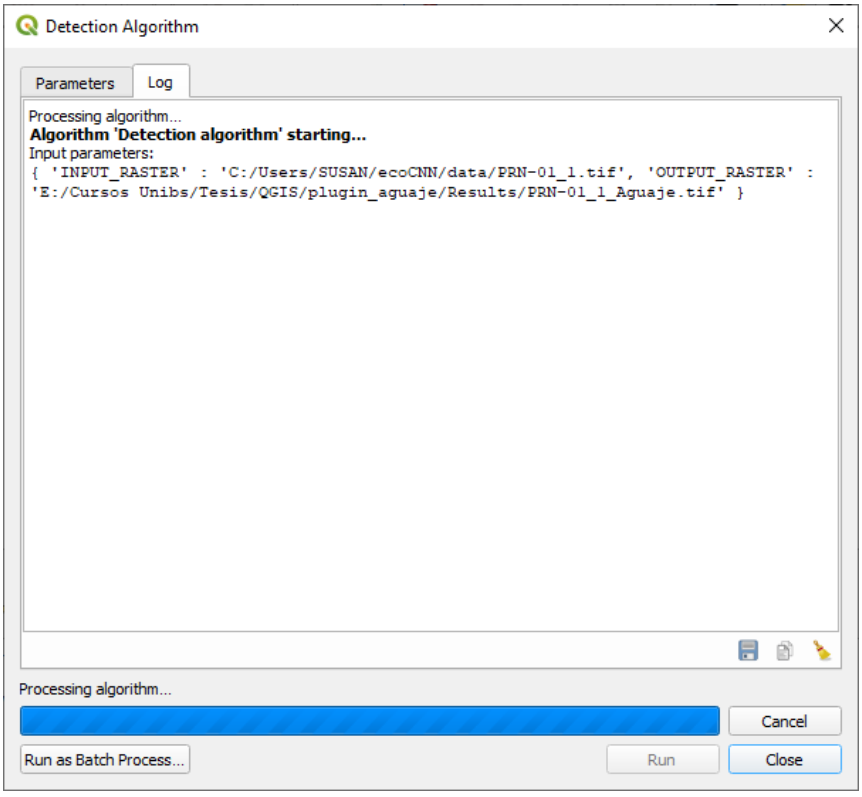


Fig. 23

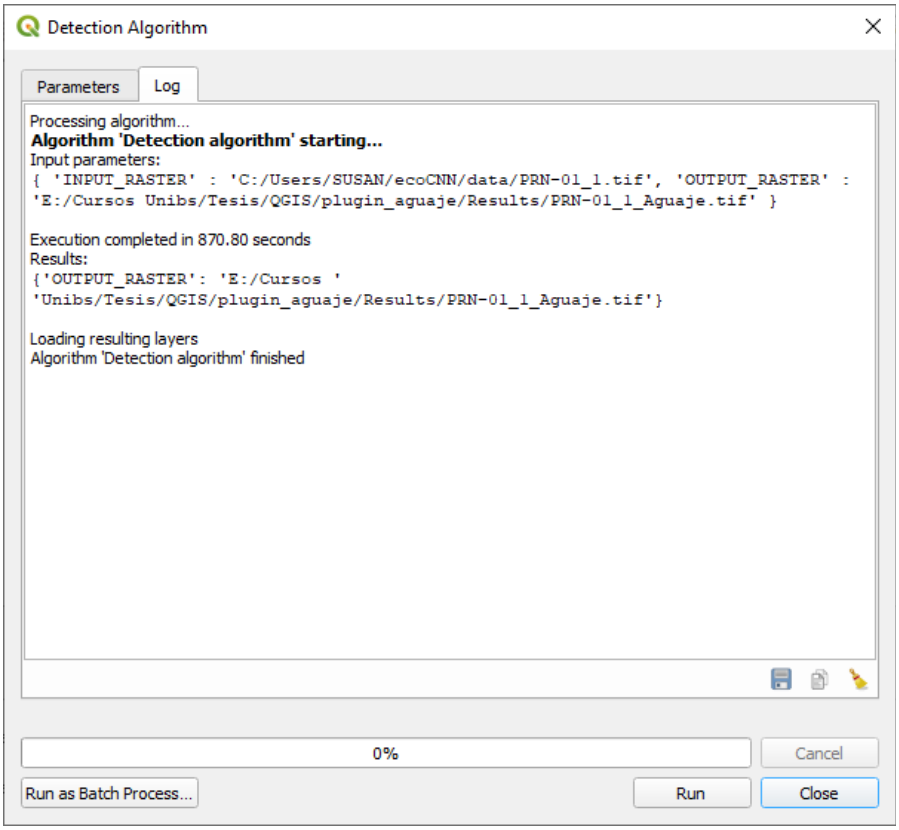


Fig. 24

9. If the *Open output file after running algorithm* option is selected, the result image will be loaded in the QGIS environment (Fig. 25).
   The segmented image .tif has -9999(NAN), 0, 1 pixel values.
   - '1' represents the Aguaje class.
   - '0' represents the No Aguaje class
   - '9999' represents NAN

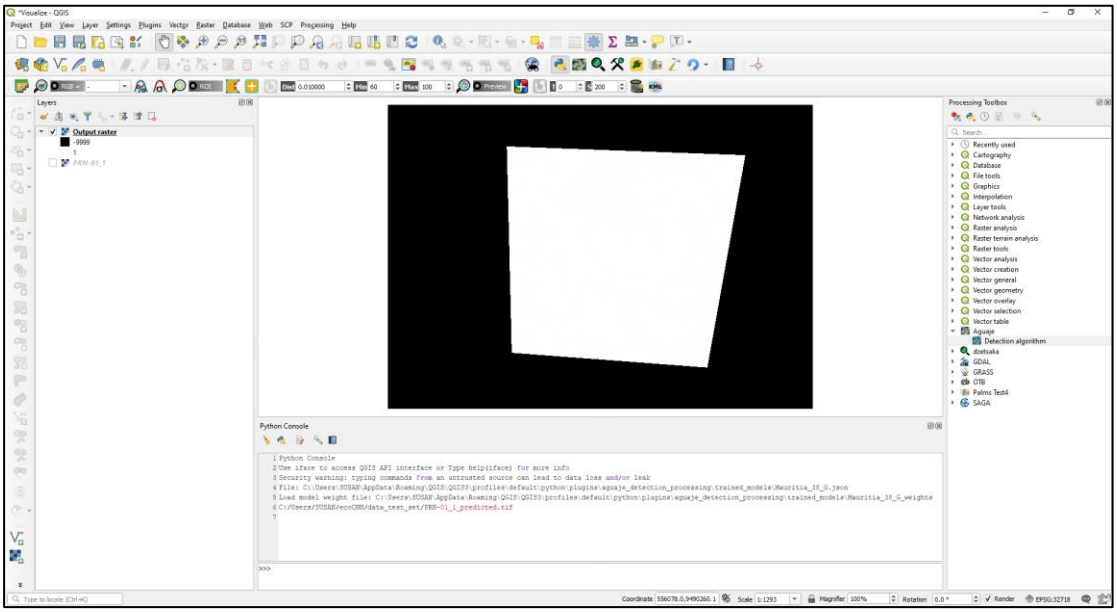


Fig. 25

10. The -9999 value should be introduced as a *No Data Value* in the layer properties (Fig. 26)

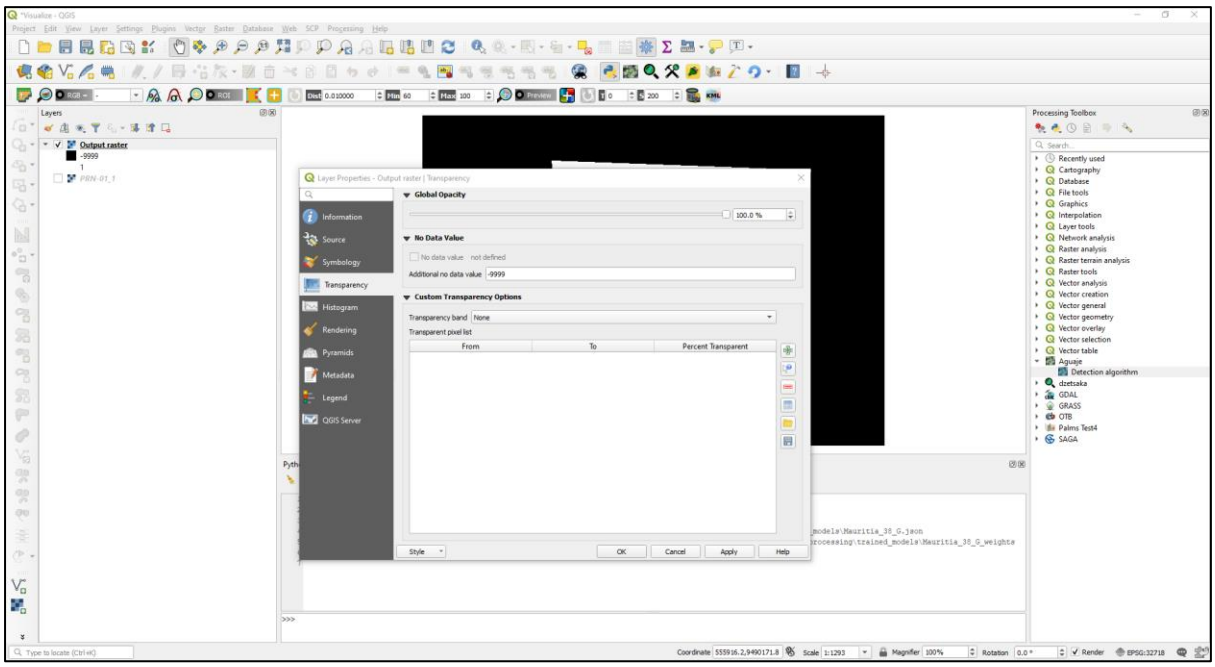


Fig. 26

11. We can observe the segmented image (Fig.27)

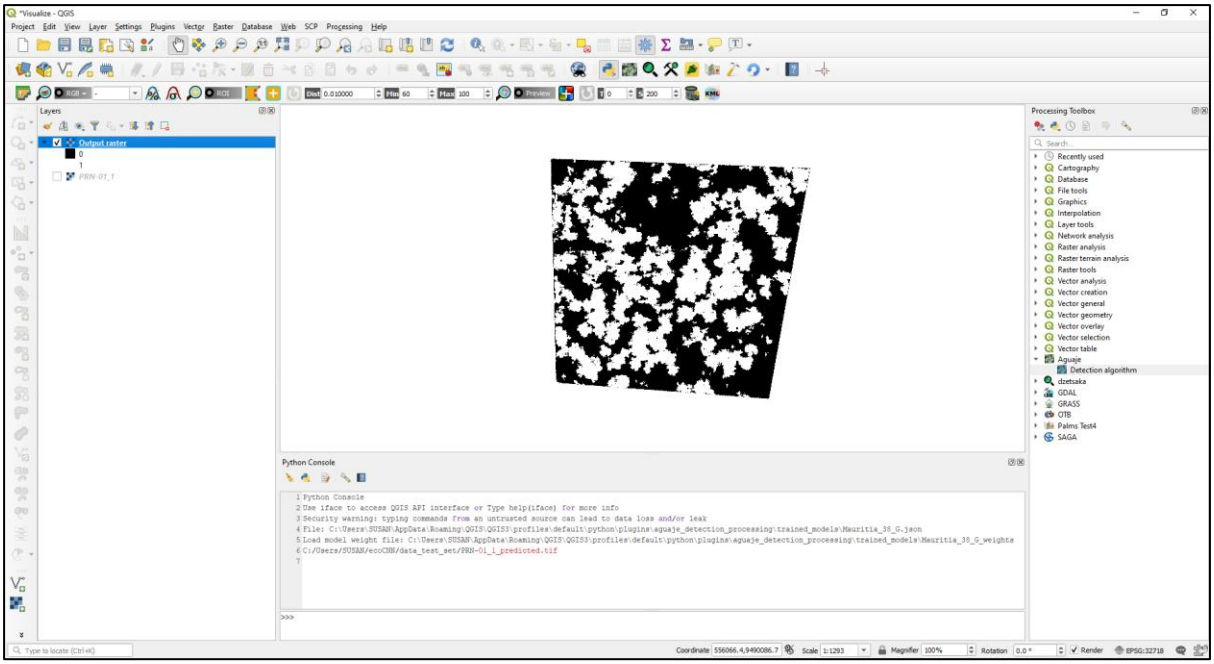


Fig. 27

12. If the *Open output file after running algorithm* were not activated, the result would not be loaded, but the results can found in the folder indicated by the user. (Fig. 28)
    The generated results are:
    - A segmented image .tif, where '1' represents the Aguaje class, '0' represents the No Aguaje class, and '9999' represents NAN. (Fig. 29)
    - A RGB classified image .tif with the Aguaje palms. (Fig. 30)

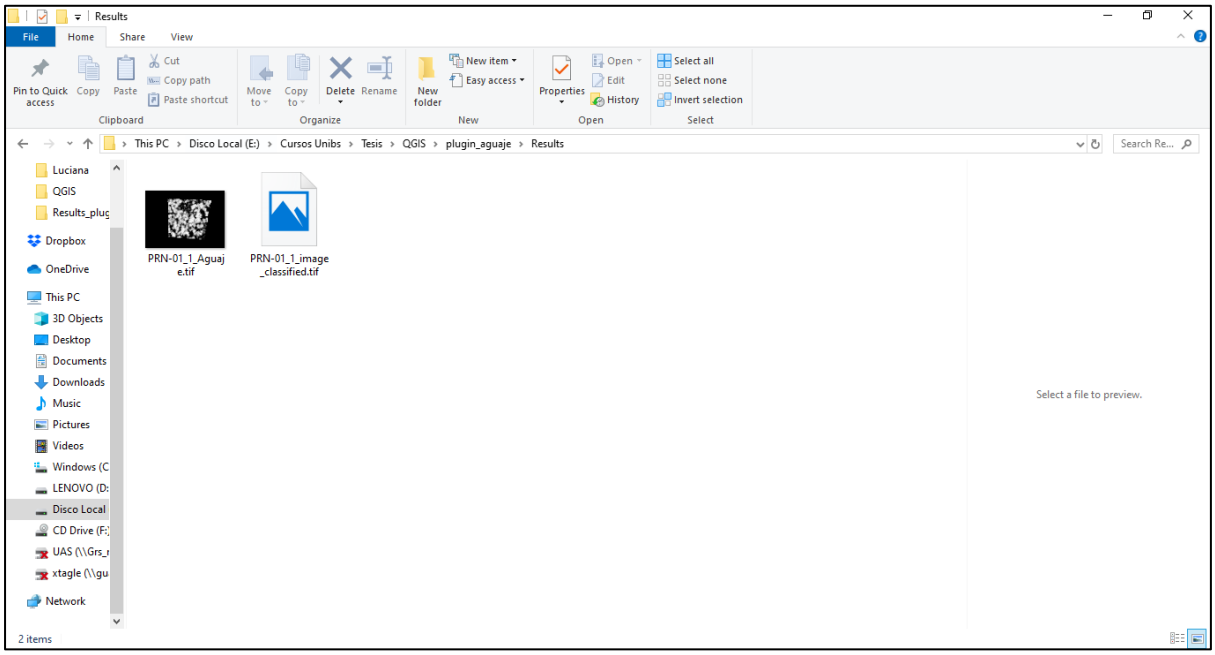


Fig. 28

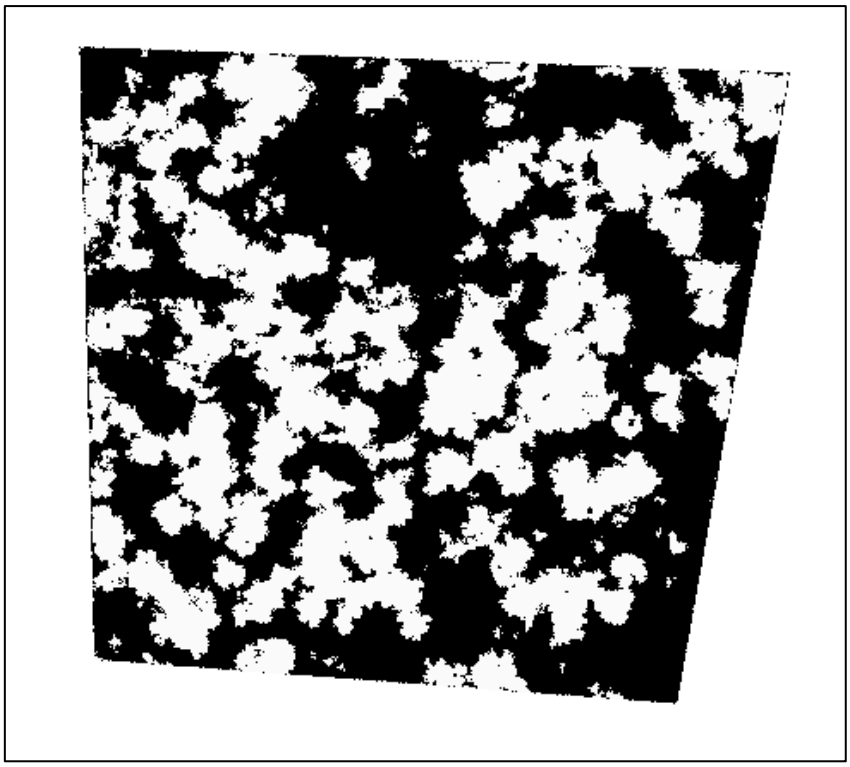


Fig. 29

Fig. 30

CHAPTER 5: ERROR

The errors can be observed in the Python Console (Fig. 31) and the Log in the Plugin Algorithm (Fig. 32).

Could not load model weight file: C:\Users\SUSAN\AppData\Roaming\QGIS\QGIS3\profiles\default\python\plugins\aguaje_detection_processing\trained_models\Mflexuosa_VEN05_G_32_weights

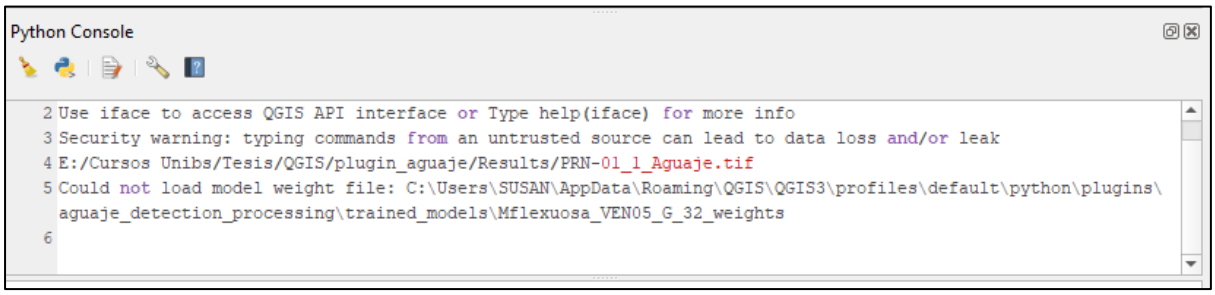


Fig. 31

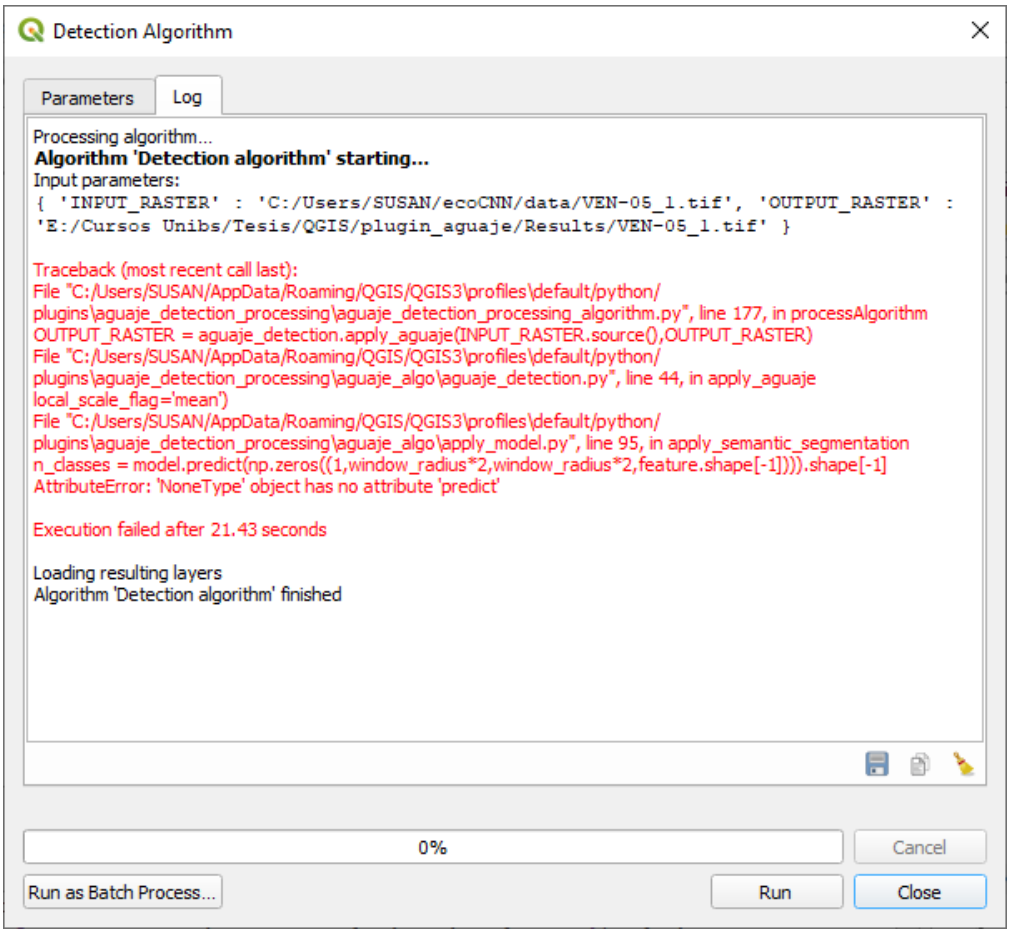


Fig. 32

Solution: Restart QGIS.