**Coastal Hawkeyes**

**Manual Version 0.1**

**Author**

**Zhanchao Shao**

[**z.shao@waikato.ac.nz**](mailto:z.shao@waikato.ac.nz)



**Table of Contents**

[***Section 0 Launch Coastal Hawkeyes 3***](#_heading=h.gjdgxs)

[Section 0.1 Initialisation in PyCharm 3](#_heading=h.qteuauo22rxp)

[Section 0.2 Google Earth Engine and Google Cloud 3](#_heading=h.8cvhg4usk0yq)

[Section 0.3 Launching the GUI 5](#_heading=h.b3lolkf7xmpk)

[***Section 1 Request Satellite Data 6***](#_heading=h.30j0zll)

[**Section 1.1 Authentication 6**](#_heading=h.1fob9te)

[**Section 1.2 Search Imagery (and Export) 8**](#_heading=h.3znysh7)

[**Section 1.3 Composite Bands (Reflectance Correction) 10**](#_heading=h.2et92p0)

[***Section 2 Label Landcovers For Machine Learning 12***](#_heading=h.tyjcwt)

[***Section 3 Supervised Classification and Density Estimation 16***](#_heading=h.3dy6vkm)

[**Section 3.1 Landcover Detection 16**](#_heading=h.1t3h5sf)

[**Section 3.2 Seagrass Density Estimation 19**](#_heading=h.4d34og8)

# 

# Section 0 Launch Coastal Hawkeyes

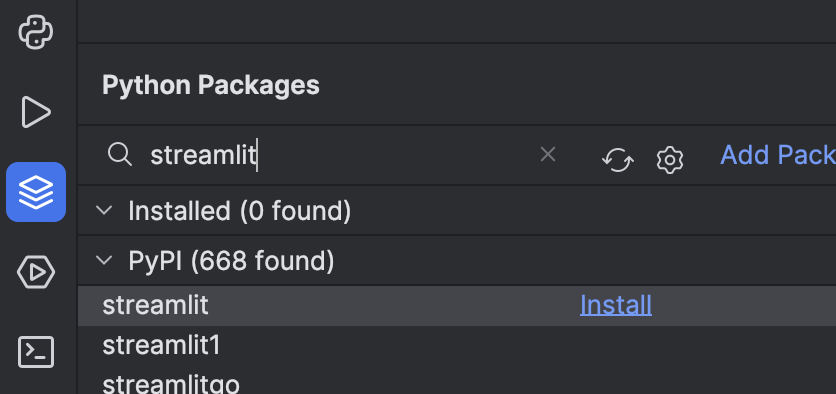
## Section 0.1 Initialisation in PyCharm

Create a new project in PyCharm. Under the project, create a folder called ‘temp’ to store results.

Download the GUI (CoastalHawkeye.py) from GitHub and save it in the new project.

Set up the interpreter Python 3.9.

Before starting the GUI, make sure all required packages are installed including: json, streamlit, ee (EarhEngine-api), geemap, geopandas, rasterio, joblib, pandas, numpy, folium, base 64 (probably already be on pc), scikit-learn, google-auth-oauthlib, GoogleAPI, shapely, zipfile (probably already on pc), scipy, scikit-image, streamlit\_folium and io (probably already on pc).



## Section 0.2 Google Earth Engine and Google Cloud

Make sure that your Google account is activated on the Google Cloud ([Cloud Computing Services | Google Cloud](https://cloud.google.com/)) and in the Google Earth Engine (<https://earthengine.google.com/> ). You’ll need to fill in your credit card information in the Google Cloud website, but now worries it’s FREE!

Users need to email ([z.shao@waikato.ac.nz](mailto:z.shao@waikato.ac.nz) ) the author to get a Google Key and the access to the shared Earth Engine project:

1. A Google Drive key starting with *Client\_secret…..json (Might be available on Github)*

Both keys should be saved into the same directory as the CoastalHawkeye.py.

To help the code find the key, you’ll need to change 1 line in the CoastalHawkeye.py code**:**

*Line 38:*

CLIENT\_SECRETS\_FILE = "C:/Users/svaa064/PycharmProjects/WetlandClassificationModel/client\_secret\_742644021504-tkc9tnfvq8vg7ksd1885a87p0uutut55.apps.googleusercontent.com.json" *# Update with your path*

where you add the path to the folder where the code and the keys are saved and enter the names of the respective keys.

## Section 0.3 Launching the GUI

A manual launch command should be entered in A blue square with white text in it

Description automatically generated:

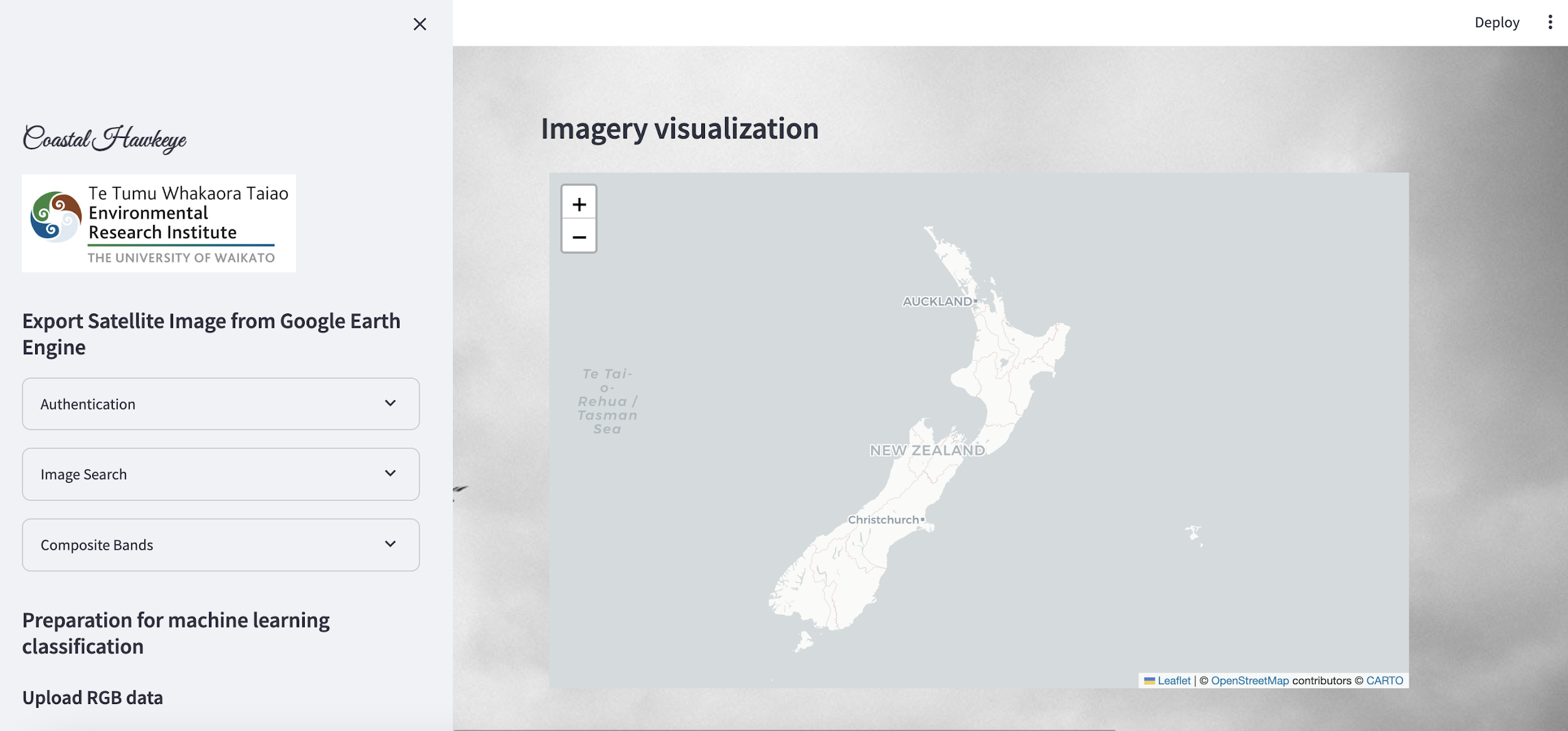
*streamlit run CoastalHawkeyes.py --server.maxUploadSize=8000 --server.maxMessageSize=8000*

****

## double check the name of .py (sometimes, the name has the version number).

Users are able to set the maximum input/upload file and result size depending on the study sites. Usually, 8GB is enough for most occasions.

The GUI is going to opened in the default browser (Chrome and Firefox are recommended) (see below).



# 

# Section 1 Request Satellite Data

In this section, users can export the satellite data (Sentinel-2) from Google Earth Engine and save the data in their personal Google Drives. The copyright of data from Google Earth Engine may be restricted if the data is for commercial use. Users can instead download the data directly from European Space Agency (<https://browser.dataspace.copernicus.eu/>) (Registration required).

## Section 1.1 Authentication

In order to export Sentinel-2 data from Google Earth Engine (GEE), the users need to connect the GUI with GEE and Google Drive.

Click “*Authenticate with Google Earth Engine*” . If GEE is successfully connected, a notification will appear (See below).

A screenshot of a satellite image

Description automatically generated

Click “*Authenticate with Google Drive*”. A new tab will pop out and select the users’ Google account (See below).

A screenshot of a computer

Description automatically generated

If the user is notified that Google hasn’t verified this app, kindly ignore it and click “*Continue”* twice (See below).

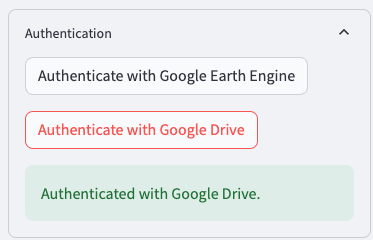
A screenshot of a computer

Description automatically generated A screenshot of a computer

Description automatically generated

If the GUI is successfully connected to Google Drive, two messages will appear in the window and GUI, individually (See below).





## Section 1.2 Search Imagery (and Export)

In this subsection, users will be able to search and save needed satellite imagery based on different conditions.

Click “*Imagery Search*” and select “*Sentinel-2*” (Landsat data is currently not available). Specify the “*Start Date*” and “*End Date*”. Users will be asked to define the region of interest (ROI) based on the *Latitude and Longitude* or *upload the GeoJSON* file (The uploaded GeoJSON file will also be displayed in the map, see below. Make the geospatial coordinate system of GeoJSON is WGS84 EPSG 4326). The uploaded GeoJSON will be used to clip the satellite image to avoid exporting extremely large size data. Therefore, using GeoJSON is more recommended than simple latitude and longitude.

A screenshot of a map

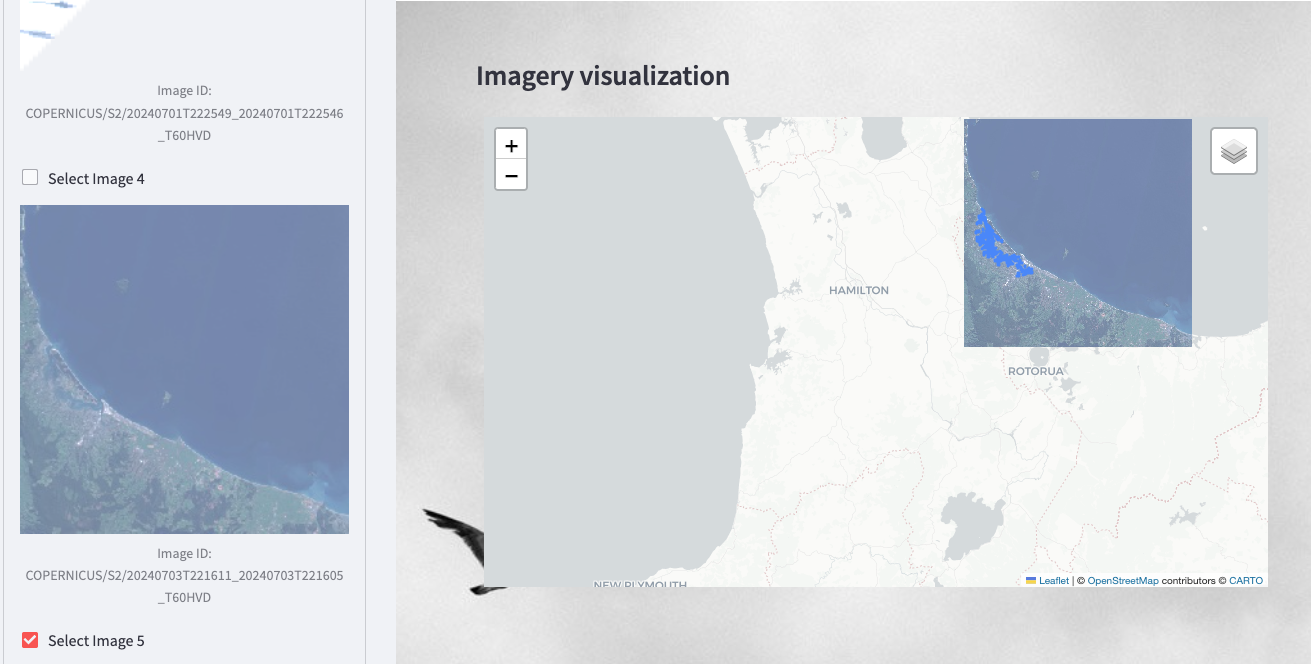
Description automatically generated

After defining the ROI, users have to set the “*Max Cloud Coverage (%)*”. The common values range from 10 to 30 depending on the users’ needs. Click “*Fetch Data*” and search the data after all conditions are set (see below).

A screenshot of a phone

Description automatically generatedA screenshot of a phone

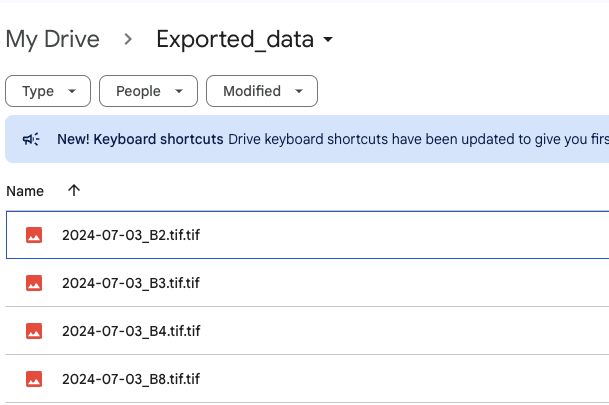
Description automatically generated

All Sentinel-2 images which satisfy the conditions will be displayed in the sidebar. Detailed images will be displayed in the map if users tick the boxes (see below).

If the selected image satisfies users’ need, users can select the required bands (usually Band 2, 3, 4 and 8 are needed) and export the image to the personal Google Drive after entering the folder name (see below). Users may wait for a couple of minutes to see the exported image.

Check Google Drive whether the data is exported. Each band will be exported individually as a Geotif file. The names of these files correspond to the data when satellites revisit the ROI.

A screenshot of a phone

Description automatically generated

## Section 1.3 Composite Bands (Reflectance Correction)

In this subsection, users are able to conduct reflectance correction and composite four bands into one big Geotif file, which allows users to visualize the data with RGB. These functions are compatible with the data exported using this GUI and downloaded from ESA.

All bands are required to be uploaded to the GUI in the sidebar (See below).

A screenshot of a phone

Description automatically generatedA screenshot of a phone

Description automatically generated

Tick the box if reflectance correction is needed (normally required). Set *“Scanning Date”* of the uploaded data, which should be consistent with the file name if the data is exported from this GUI. ESA has two different ways of reflectance correction before and after January 2022. Click “*Apply Reflectance Correction*” after setting the conditions for reflectance correction (see below). If reflectance correction is not need, users will be able to go to the next step.

A screenshot of a computer

Description automatically generated

In “*Composite Bands and Export”*, the user can *set coordinate system* and *the path for the composited image* (Caution: from this step, all the paths for results will be saved locally instead of Google Drive. If users install Google Drive on their PCs and prefer to save the data on the drive, a full path to Google Drive is need. For example: /Users/zhanchao/Library/CloudStorage/GoogleDrive-XXXXXX@gmail.com/My Drive/Exported\_data/composited.tif ) (or save the files on your pc, see below). Click “Save Composited Images” after all conditions are set.

A screenshot of a computer

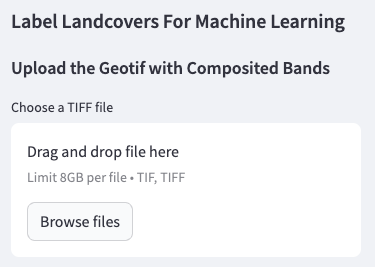
Description automatically generated

# 

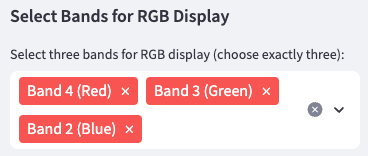
# Section 2 Label Landcovers For Machine Learning

In this section, users will be able to manually label different landcovers using folium map. This process can also be accomplished in QGIS and ArcGIS.

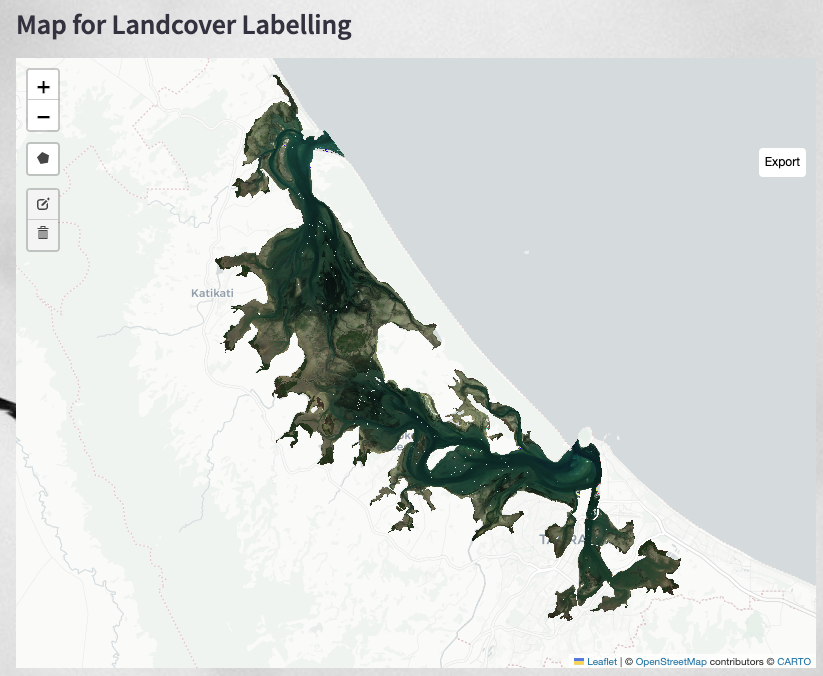
Upload the Geotif file with the composited bands.



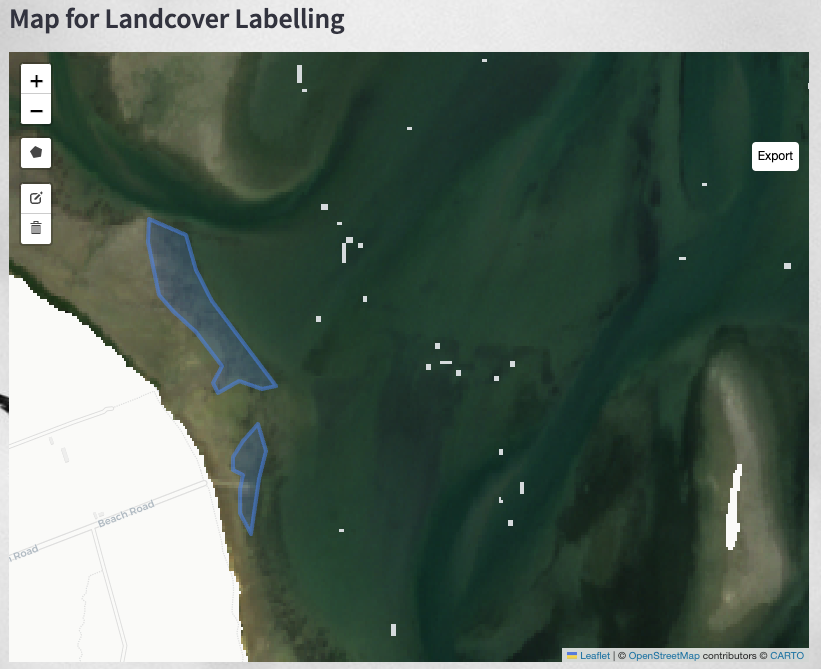
The GUI automatically selects band 2 blue, 3 green and 4 red to display in the map. Users are allowed to change different bands based on their needs in “*Select Bands For RGB Display*”.



Here is an example of Sentinel-2 imagery for landcover labelling (see below).

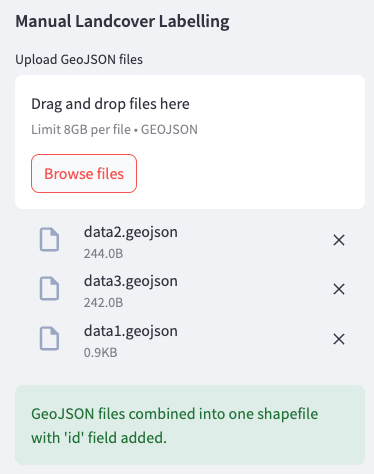


Users can label different land with the toolbox in the map. Click the polygon in the toolbox to start manual labelling. Users can draw as many polygons as they want. However, at each time, make sure all the polygons should be the same class (e.g. seagrass) (see below). ##This function might be affected if you are using Python 3.12 and above.

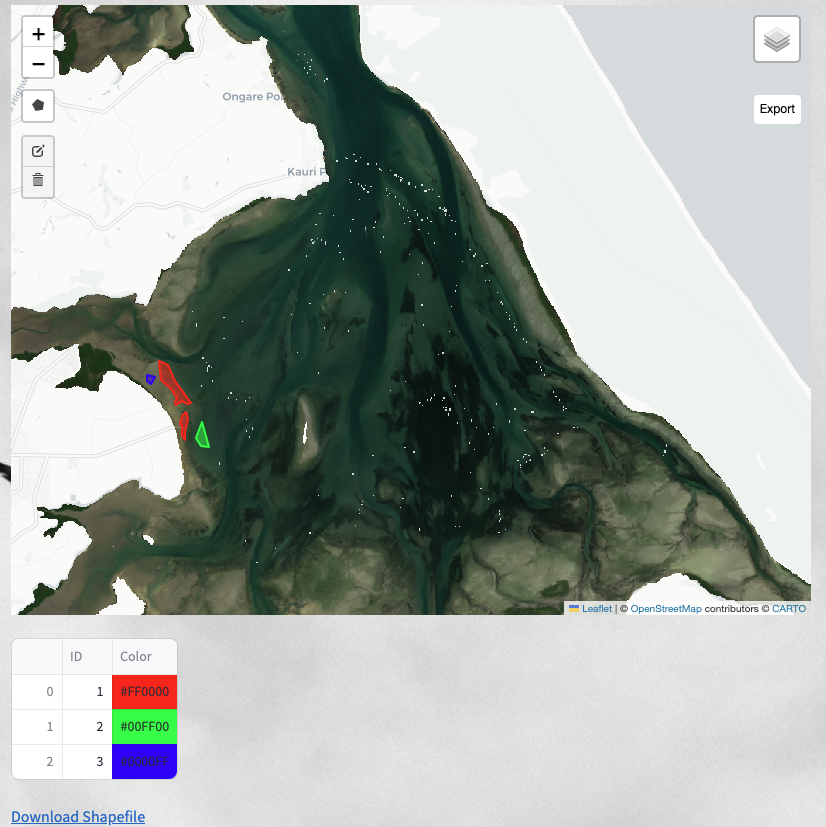


Once all polygons for seagrass are drawn, click “*Export*” to save the polygons as GeoJSON. It is recommended that save the name with a class number (e.g., xxx1.geojson). After the polygons for seagrass are exported, click “*Clear All*” to restart the labelling for other landcovers.

When all GeoJSON files are created, a large shapefile containing all these GeoJSON files should be created for supervised classification. Go back to the side bar, in “Manual Landcover Labelling”, users are able to upload all the GeoJSON files.



The GUI displays all classes in the map with different colours. A table of different classes with unique IDs is generated below the map for users to double check. Click “*Download Shapefile*” to save the shapefile into the local drive.



These are the complete steps for manual labelling in the GUI. Users can also label landcovers in QGIS or ArcGIS.

# 

# Section 3 Supervised Classification and Density Estimation

There are two main functions in this section. The GUI can help 1) detect the distribution of different landcovers using supervised classification with random forest and 2) estimate the seagrass density using support vector regression.

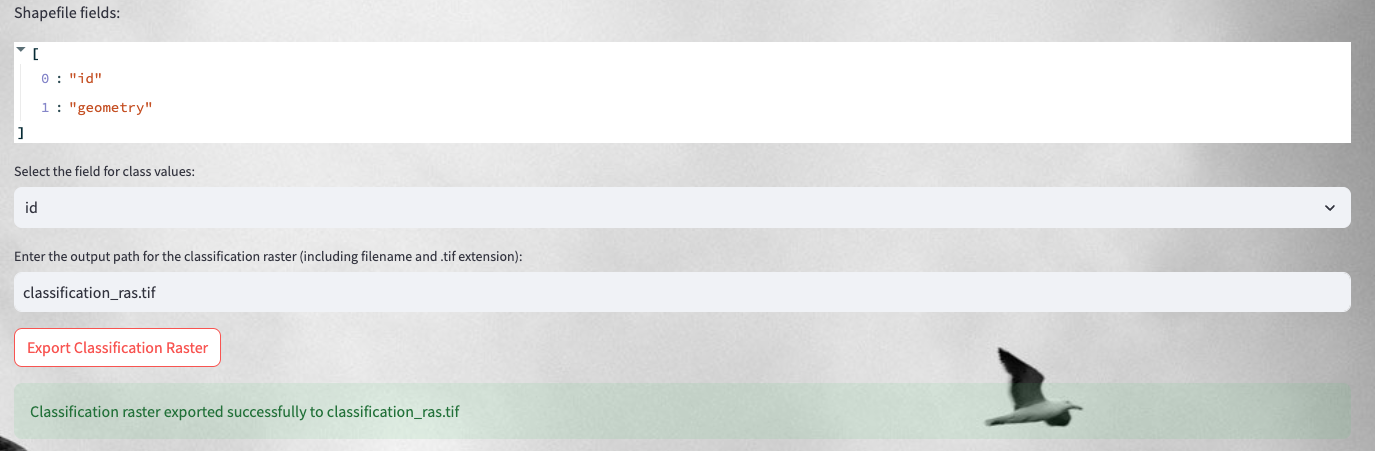
## Section 3.1 Landcover Detection

In this section, users are able to train a supervised classification model using the manual labelling data (or ground truthing points).

Click “*Train New Model*” if the classification model is not prepared. Users need to upload the Geotif file with all bands (Geotif with Band 2, 3, 4 and 8 after composition in Section 1.3) and shapefiles (manual labelling classes in Section 2) (see below). And set the “*Training Percent*”. The recommended value is 70~90, which means 70~90% of the data points from the shapefile are for training and the rest are for testing. Validation is not required as the recommended values for hyperparameters will be given in the coming steps.



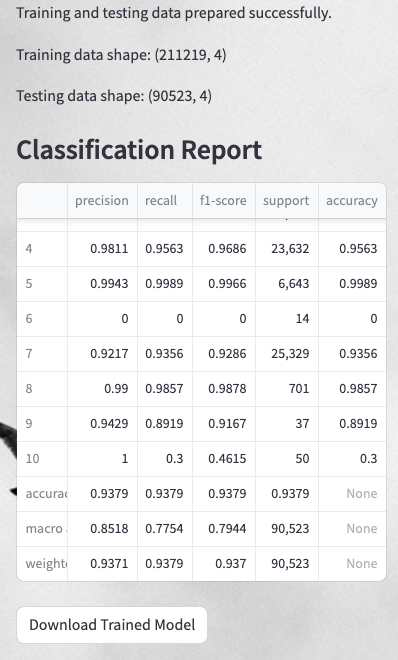
The detailed information about the shapefile is given (see below). Users need to select the field which records the class values for each type of landcover (normally ID). The GUI will then automatically extract the reflectance values of all bands at the locations of the manually labelled polygons (or ground truthing points) and export the values as classification raster in the form of Geotif (because shapefile is not directly compatible with machine learning). Users can specify the path to export the classification raster. Click “*Export Classification Raster*” to start the process (see below).



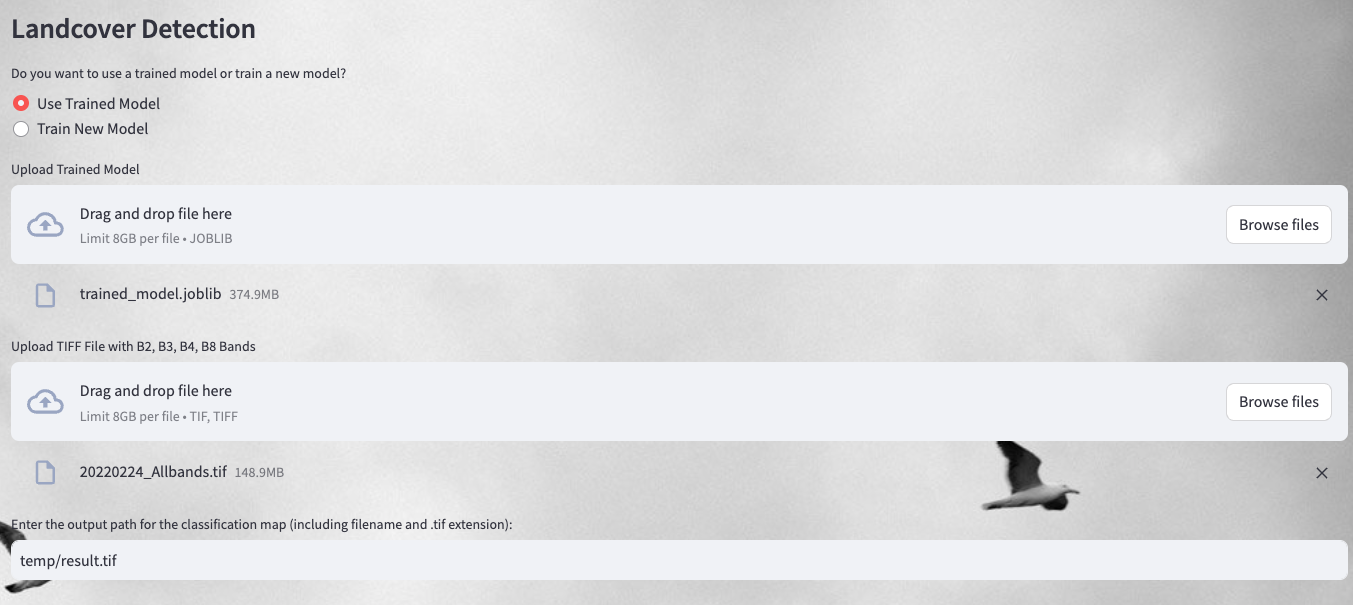
Users are allowed to change the values of hyperparameters for machine learning. The recommended values are given (see below). Click “*Start classification*”.



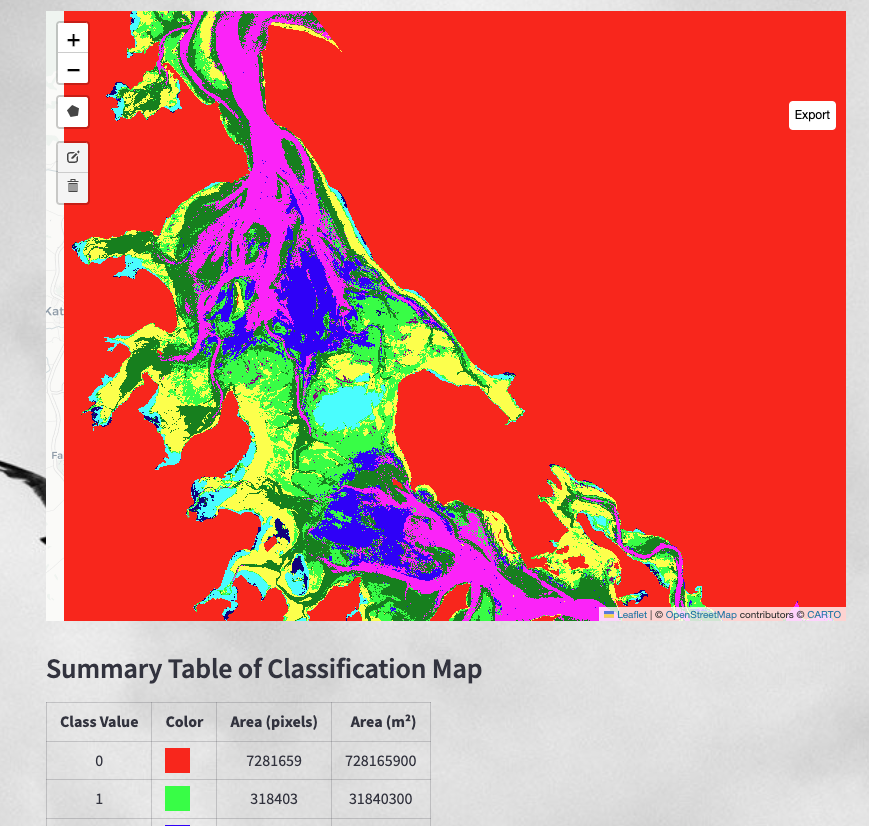
After classification, an overall report will be given. Users will be able to download the trained model by clicking “*Download Trained Model*”. (Notes: Values in the training data shape means 211,219 points are used for training with four different bands).



After the classification model is created, go back to the beginning question and select “*Use Trained Model*”. Upload the trained model and the Geotif file with Band2, 3, 4 and 8 which needs to be classified. Enter the output path to save the classification map (see below).



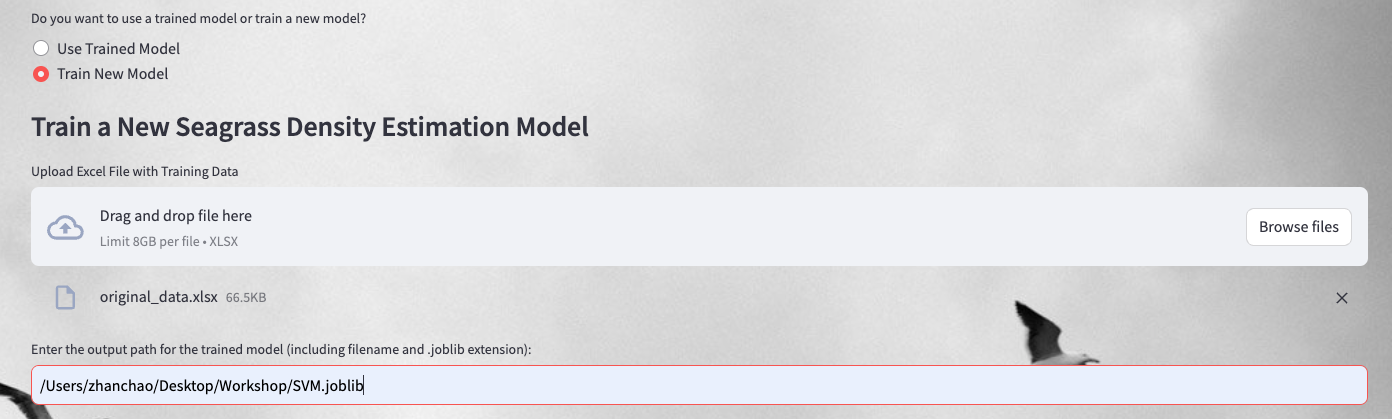
A classification map and summary table will be given after processing (see below). Users need to clip the classification map to exclude nan-coastal regions (red part) in QGIS or ArcGIS.



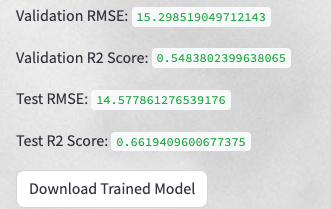
## Section 3.2 Seagrass Density Estimation

In this section, users are able to train a support vector regression (SVR) model to predict the seagrass density and apply the model to all seagrass pixels.

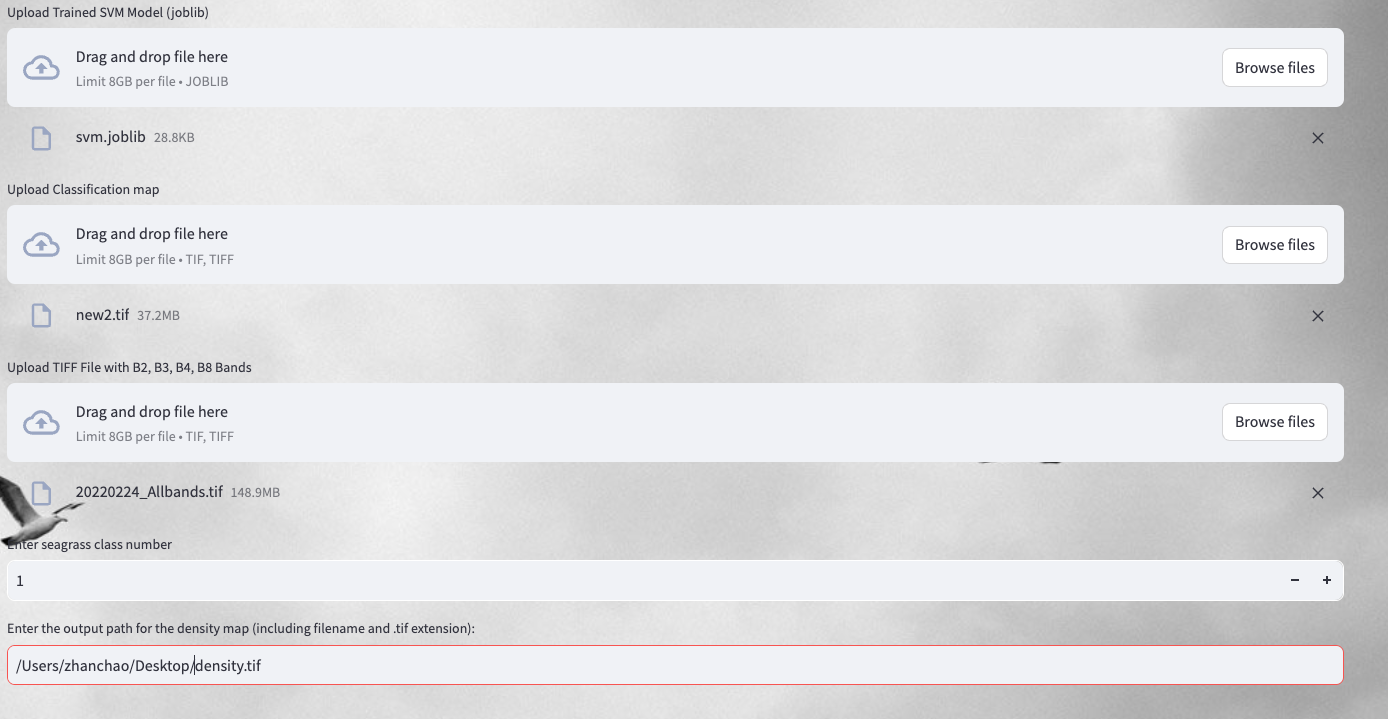
Click “*Train New Model*” and upload the excel spreadsheet which records seagrass percentage cover and reflectance data (Please check the excel provided). Enter the path for the output machine learning model (see below).



The GUI will provide the performances of the SVR model.



Users can apply this model to all seagrass pixels detected from the classification model. Click “*Use Trained Model*” and upload the trained SVM model, classification map (.tif), and the composited Geotif files with Band2, 3, 4 & 8. Users need to specify seagrass class number in the classification map. And set the output path (see below).



Notes:

If the GUI pop out the error, please retrain the machine learning model.