Automated Shoreline Extraction and Analysis Tool

User Manual

Version v.1.1.0

Author: Men Vuthy (M.Eng)

Water Resource and Energy Department

Nippon Koei Co., Ltd.

Tokyo, Japan

Contents

List of Figures

Automated Shoreline Extraction and Analysis Tool

User Manual

I Introduction

Automated Shoreline Extraction and Analysis Tool known as ASEA Tool is a rapid and user-friendly tool capable of extracting the shoreline from Sentinel-2 satellite imagery at a high accuracy and calculating the shoreline change between the past and present condition. The tool is fully operated in a cloud platform connecting to Google Earth Engine (GEE) so that the complicated environment setup is not necessary in local computer to utilize this tool. By giving the target area and date range as input information, the model will automatically collect all Sentinel-2 satellite images at the given area of interest (AOI) to calculate the median image, process them, and download the post-processed images directly from GEE to user's Google drive. From post-processed images, the tool analyzes the sub-pixel values of Near-Infrared Band and classify them into land and water area by using an unsupervised classification algorithm known as K-Means. The boundary dividing land and water area is determined as preliminary shoreline, which basically has stair-like shape. This boundary line is then converted to a smooth shoreline through shape-correction process in the model. The shape-corrected shoreline is treated as the main shoreline for calculating growth and retreat distance and rate per year between the past and present time based on transect method. By using this tool, several kinds of output will be produced in various format ranging from images in raster file (.tif) to shorelines in geojson file (.json). This tool enables coastal engineers and scientists to extract shorelines from open-source satellite images and explore shoreline changes at a regional scale with good accuracy, saving time, budget, and labor compared to manual extraction and high-cost image purchases.

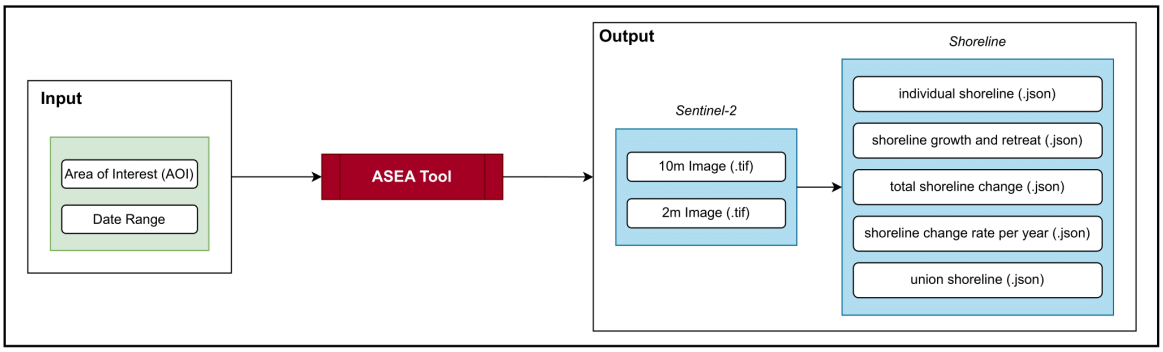


Figure 1: Schematic diagram of Automated Shoreline Extraction and Analysis (ASEA) Model.

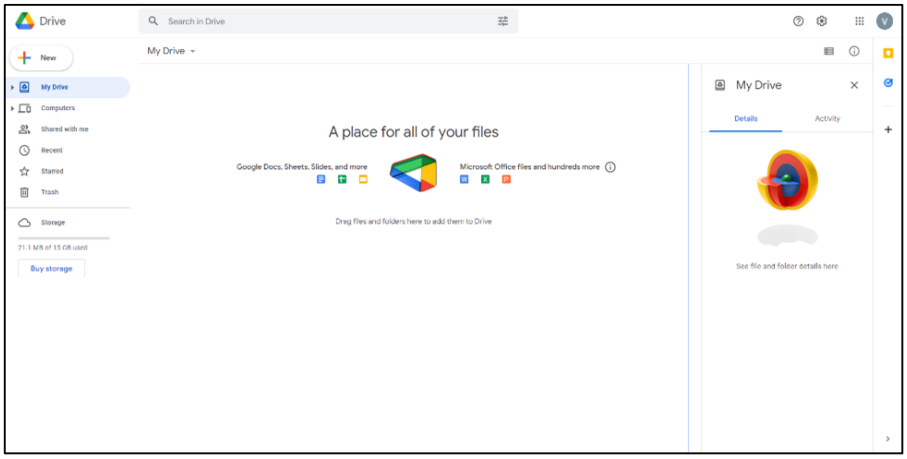
II Requirement

To use ASEA tool, Google Colaboratory and Google Earth Engine are required to implement the execution codes. Colab and GEE are two different platforms. Google Colab is a place where execution codes of ASEA model are executed, while GEE is a place where geometry of AOI is generated and also where all Sentinel-2 satellite imagery are stored and downloaded by the tool for analysis.

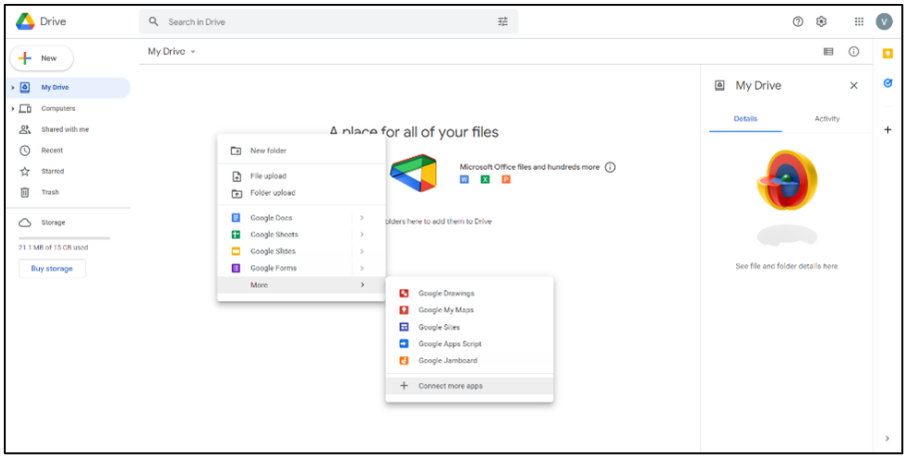
1 Install Google Colaboratory

Google Colaboratory also known as “Colab” is a product from Google Research. Colab allows us to write and execute Python code through web browser for free, and it uses Google drive as the main storage. Since Colab is not pre-installed in Google drive, it is necessary to install it for first user. The process to install Colab is as follows:

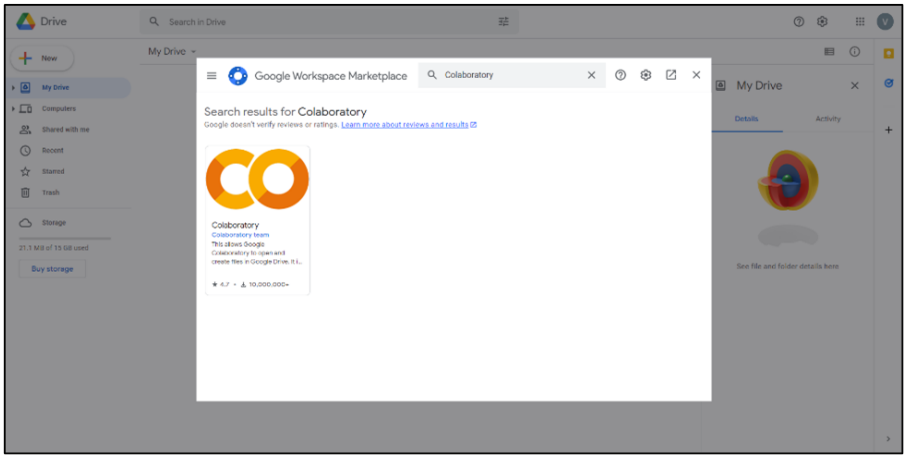
Step 1: Open Google drive with Gmail account

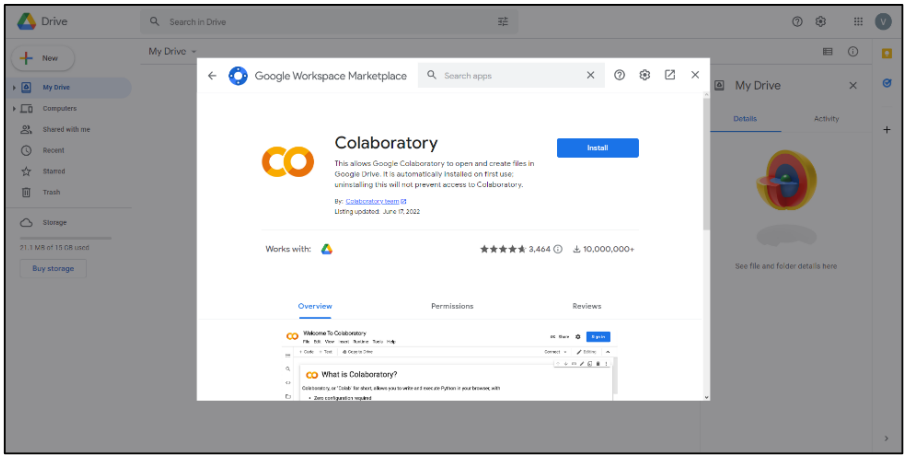


Step 2: Right-click and choose “Connect more apps”.

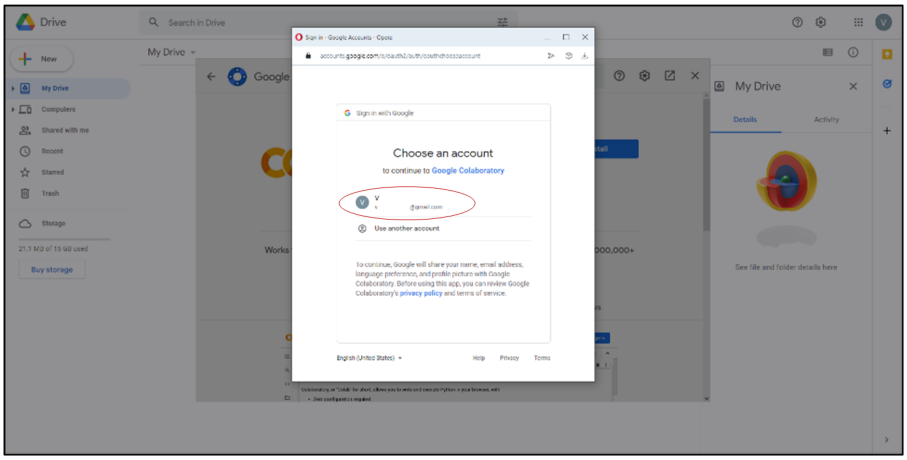


Step 3: Search for “Colaboratory” and click “Install” to install it.

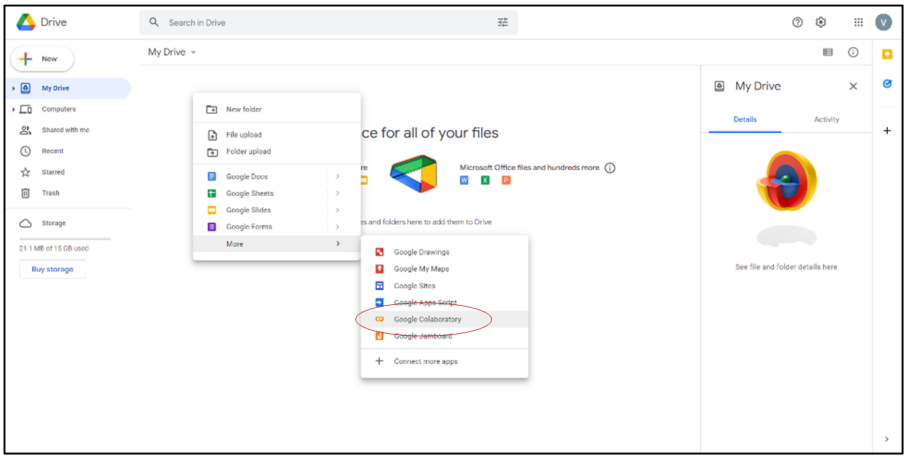




Step 4: Choose Gmail account to use Colab



Step 5: Confirm by Right-click in drive and check the list for “Google Colaboratory”



If you see “Google Colaboratory” in the option list, it means you can open any Python code (.ipynb) file and run it with your Gmail account in Colab.

Watch Introduction to Colab to learn more, or just get started here.

2 Create Google Earth Engine account

Google Earth Engine(GEE) is a computing platform that allows users to run geospatial analysis on Google's infrastructure. GEE contains a multi-petabyte catalog of satellite imagery and geospatial datasets with planetary-scale analysis capabilities. To interact with the platform, the Code Editor at code.earthengine.google.com which is a web-based IDE for writing and running scripts is used, and it requires log in with a Google Account that's been enabled for Earth Engine access. The Code Editor has the following elements (illustrated in Figure 2):

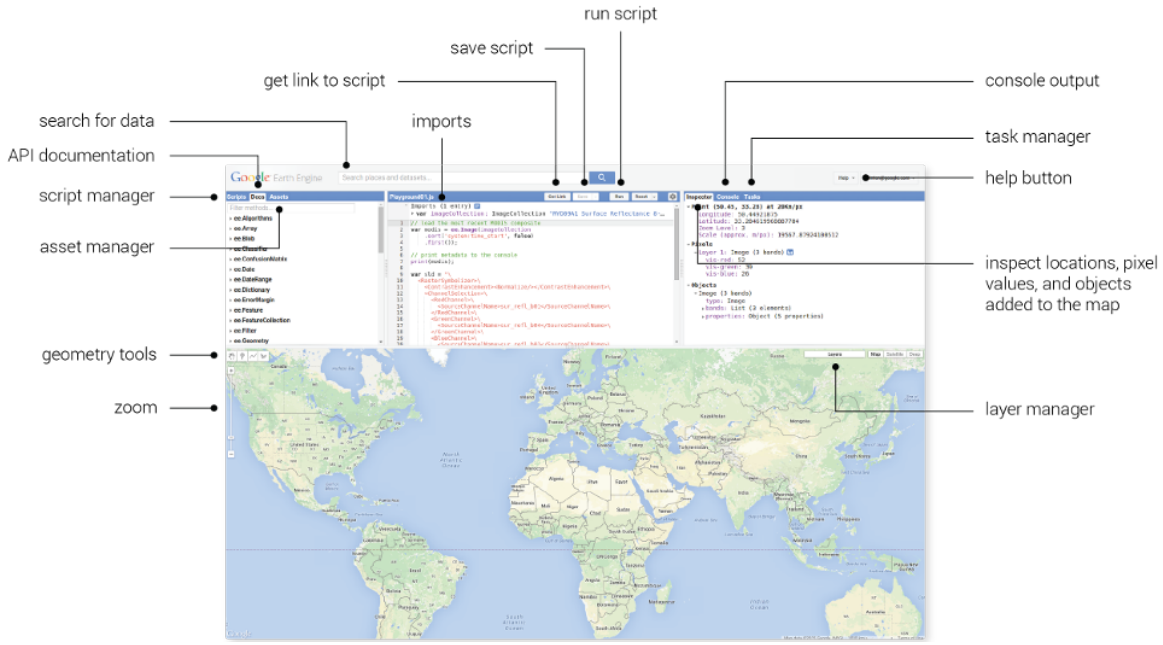
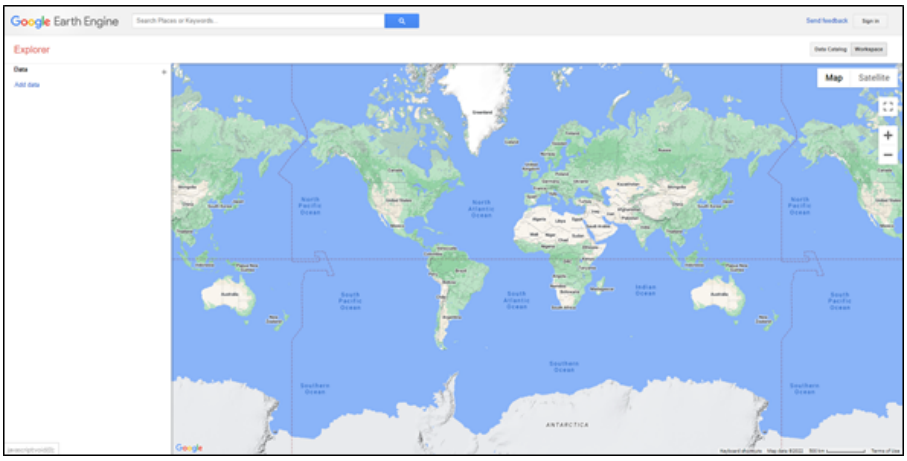


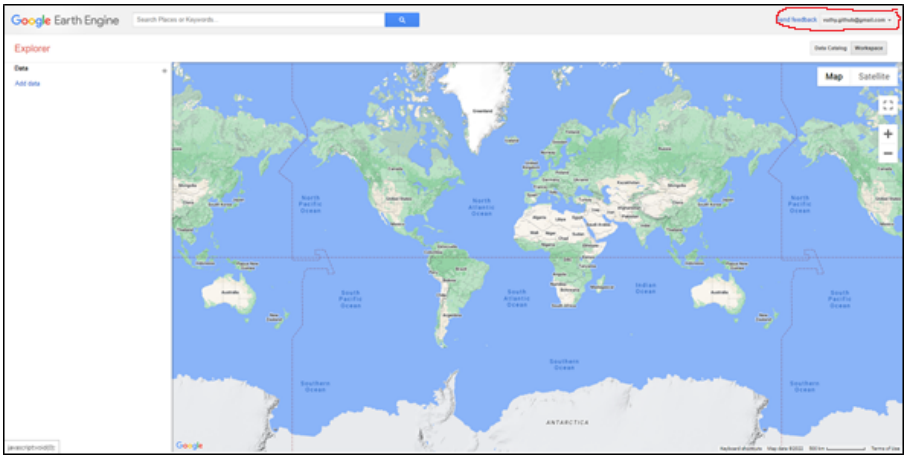
Figure 2: Interface of Google Earth Engine Code Editor.

ASEA tool does not require users to interact with GEE code editor using JavaScript API; however, the users needs to have an account and a cloud project in GEE for model authentication in Colab every time the tool is used. The process to setup an account in GEE is as follows:

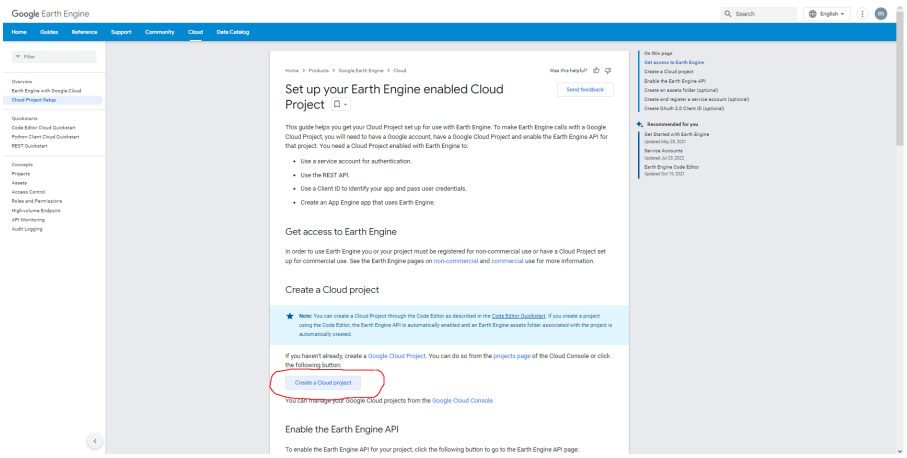
Step 1: Open Google Earth Engine Explorer and sign in with your Gmail account.



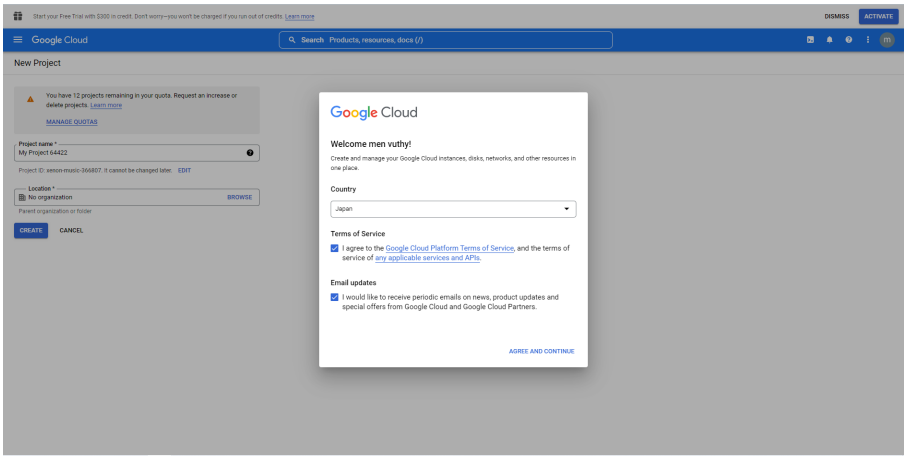
Step 2: After signing in, confirm your Gmail which appears in the top right corner.



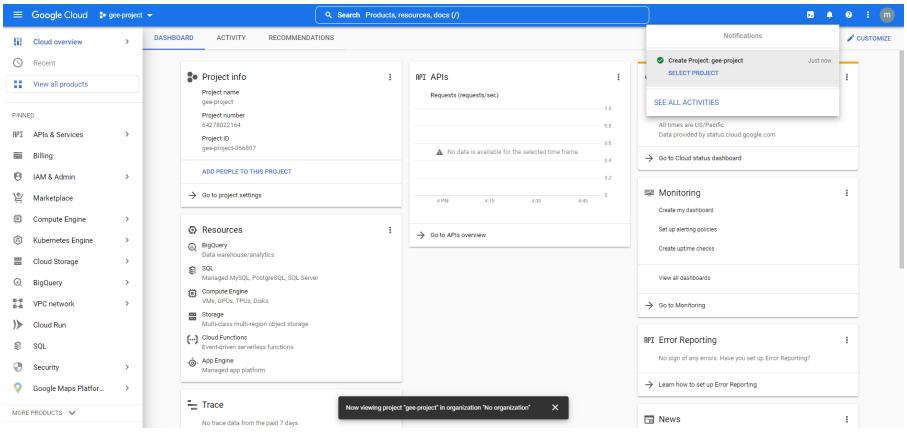
Step 3: Open Cloud Project Setup and create a cloud project by clicking on “Create a Cloud Project” button.



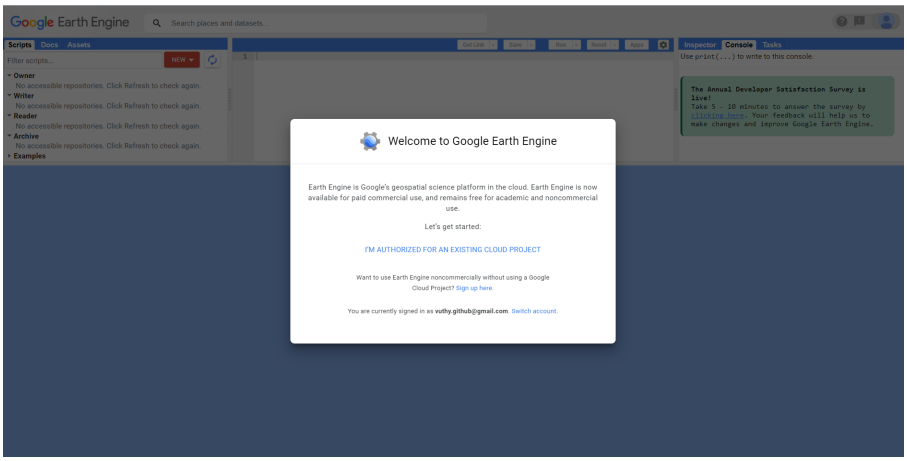
Step 4: Agree to the terms and continue. After that, write the name of project and click “Create” button.



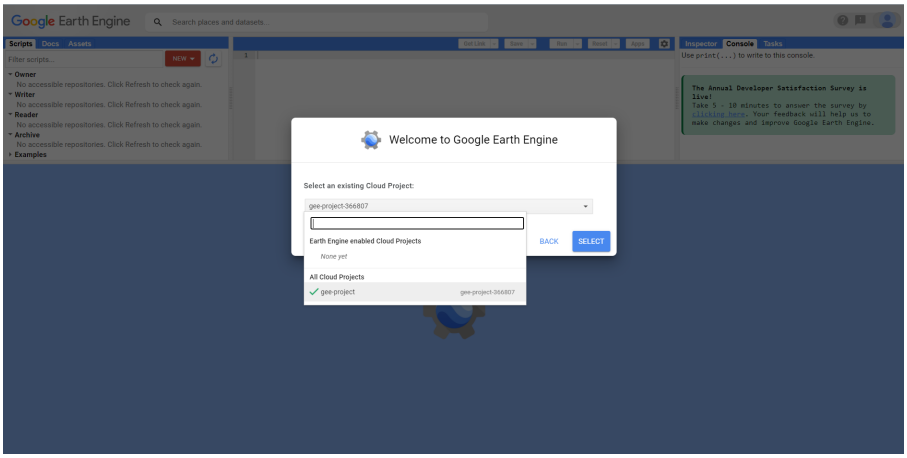
Step 5: Confirm this dashboard after creating the cloud project.



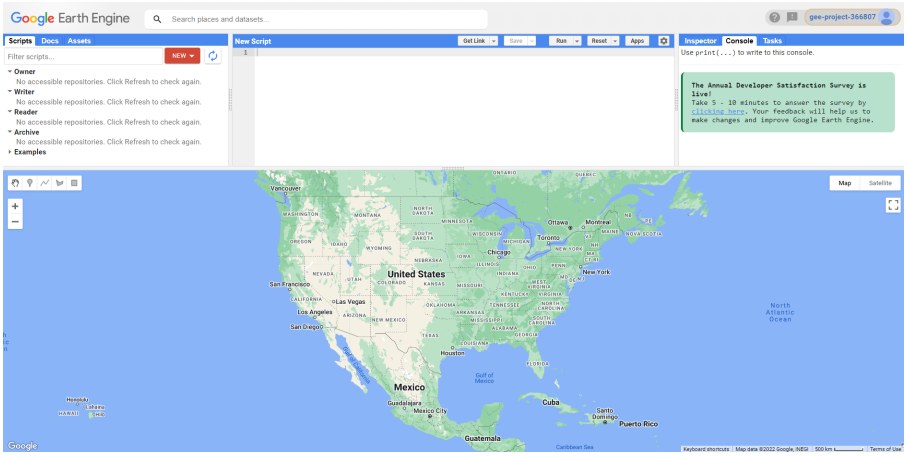
Step 6: Open Code Editor and sign in with the same Gmail account, then you will be welcomed by Google Earth Engine. Then click on “I’m authorized for an existing cloud project” button.



Step 7: Choose cloud project that you have just created, and click “Select”.



Step 8: If you see this interface, it means you have successfully created a GEE account and a cloud project for using GEE.



III Model structure

ASEA is a tool developed using Python programming language and mainly operated in Co-lab. It also can be run in local computer or other cloud platforms besides Colab; however, the environment setup might be different, and users have to take this into account and install necessary modules by themselves before use. However, it is highly recommended to run the tool in Colab because Colab has many built-in modules and simple authorization work-flow to connect Google Earth Engine through Notebook Authenticator verification code which facilitates the use of ASEA tool.

1 Download

ASEA tool is available in GitHub repository under the MIT License and Copyright ©2022 Men Vuthy (https://github.com/menvuthy/ASEA-Tool), and can be downloaded either by ZIP or cloning repository through URL (Figure 3).

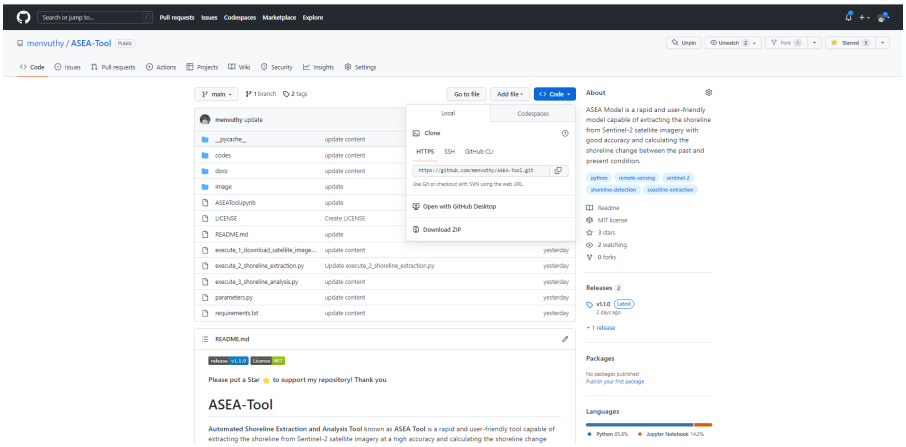


Figure 3: ASEA tool in GitHub repository (Release version: v.1.1.0).

2 Content

ASEA-Tool directory contains multiple files including LICENSE and README.md. Inside the directory, it contains a variety of files and folders as illustrated in Figure 4.

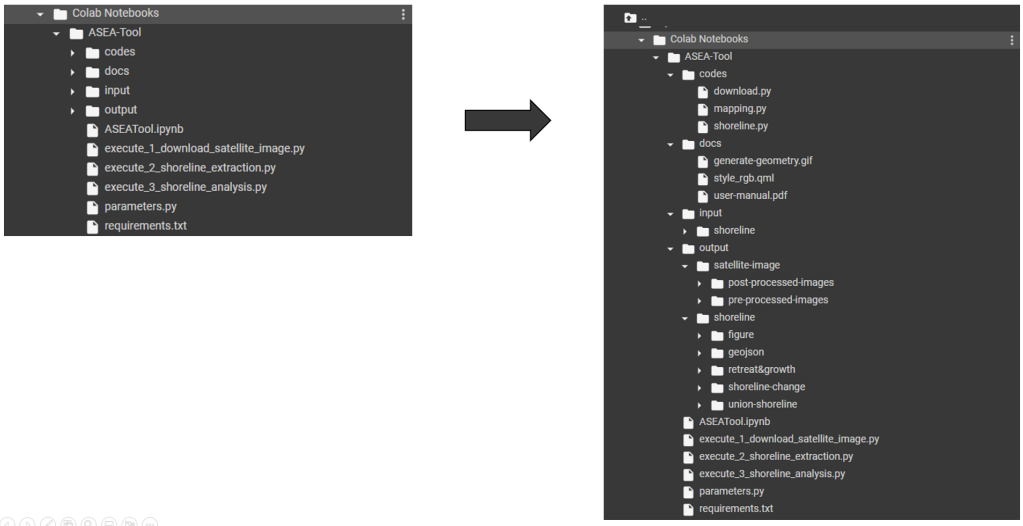


Figure 4: Content of ASEA model.

2.1 File description

- `ASEATool.ipynb` : is the IPython Notebook and the main interactive computation platform, in which all codes are executed.
- `requirements.txt` : is the environment file containing all modules required to pre-install to use many built functions.
- `parameters.py` : is the parameter setting file where the inputs such as area of interest and date range are given.
- `execute_1_download_satellite_image.py` : is the execution file to download the satellite images from Sentinel-2 data collection through Google Earth Engine.
- `execute_2_shoreline_extraction.py` : is the execution file to generate shoreline from retrieved Sentinel-2 images and create a plot of true-color image and extracted shoreline.
- `execute_3_shoreline_analysis.py` : is the execution file to calculate the shoreline growth and retreat, and generate total shoreline change, shoreline change rate per year, and union shoreline.

2.2 Main folder description

- `codes` : is the folder containing the source codes for downloading satellite images, analyzing shorelines, and mapping results.
- `docs` : is the folder containing instruction and user manual documents.
- `input/shoreline` : is the folder where user places the checked shorelines to analyze the shoreline change.
- `output` : is the folder containing all kinds of output produced by the model.

2.3 Sub-folder description

- `output/satellite-image/pre-processed-images` : is the folder containing Sentinel-2 satellite image at 10m spatial resolution downloaded based on AOI and date parameters. Downloaded image has 5 bands: *1. SWIR1, 2. NIR, 3. Red, 4.Green, and 5. Blue*.

- `output/satellite-image/post-processed-images` : is the folder containing Sentinel-2 satellite image resampled from 10m to 2m using bilinear interpolation method. This image has the same bands as original image.
- `output/shoreline/figure` : is the folder containing the plot figure of extracted shoreline and true-color image.
- `output/shoreline/geojson` : is the folder containing each and every extracted shoreline.
- `output/shoreline/retreat&growth` : is the folder containing shoreline retreat and growth result.
- `output/shoreline/shoreline-change` : is the folder containing one shoreline change which includes attribute information about total shoreline change and also shoreline change rate per year in meter.
- `output/shoreline/union-shoreline` : is the folder containing the union of all extracted shorelines.

IV How to use

ASEA tool allows users to produce three main results of shoreline analysis through three executions (i.e. download, extraction, and analysis). The instruction on how to use ASEA tool is described below:

1 Getting started

As instructed in Section 1, the tool can be downloaded as zip file from GitHub repository. After unzipping the file, upload it to Google Drive to any location as illustratedHin Figure 5.

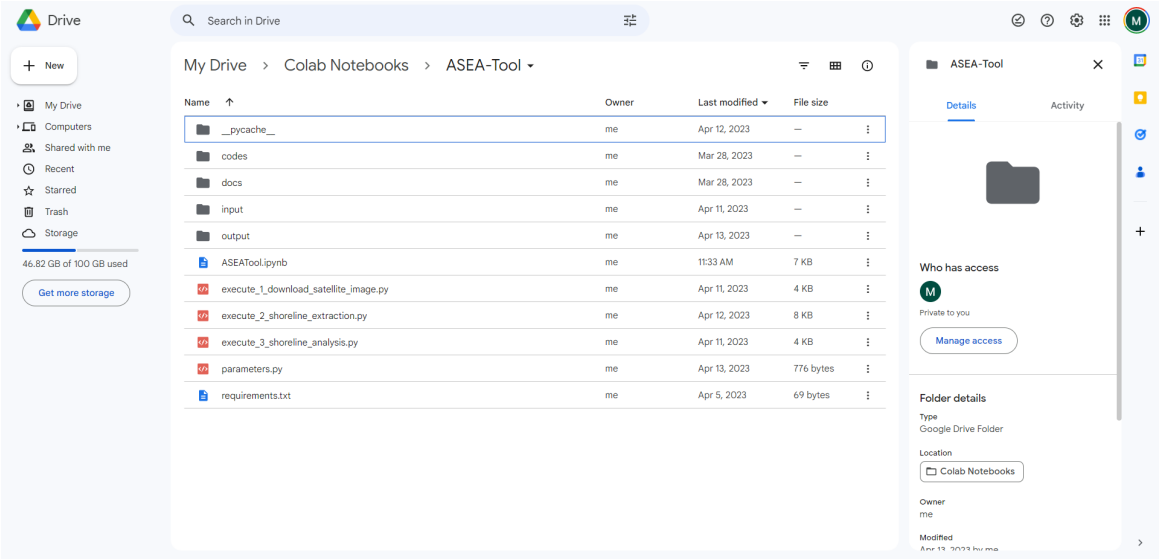


Figure 5: ASEA tool in Google drive.

1.1 Open the tool

Inside ASEA-Tool folder, there are multiple files and folders. To start the tool, double click on `ASEATool.ipynb` or right click on it and choose “Open with Google Colaboratory”. After that, the interface of ASEA tool will appear as shown in Figure 6.

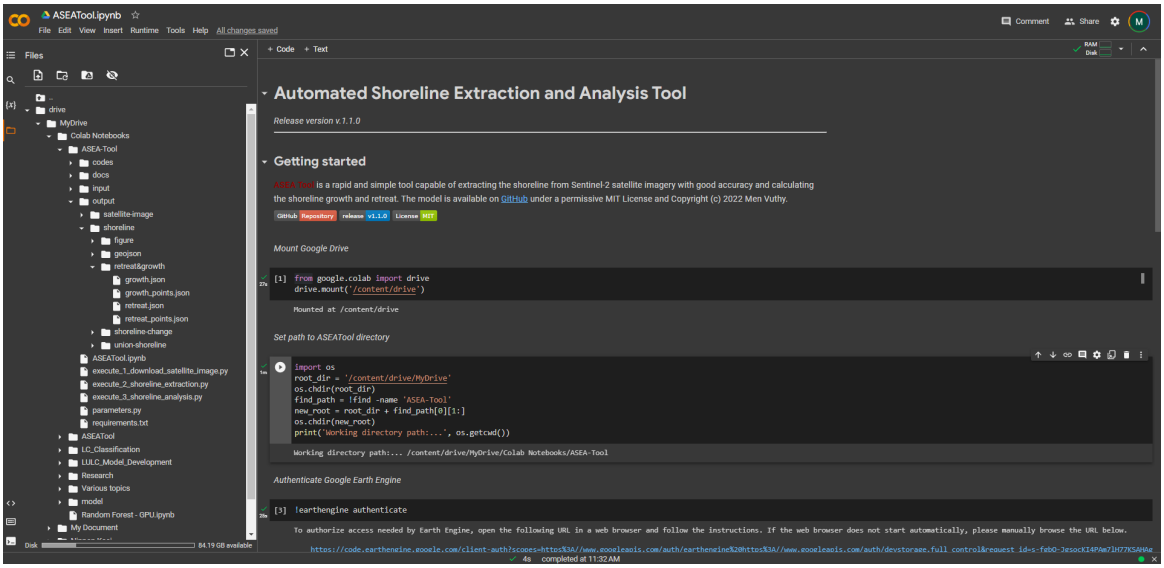


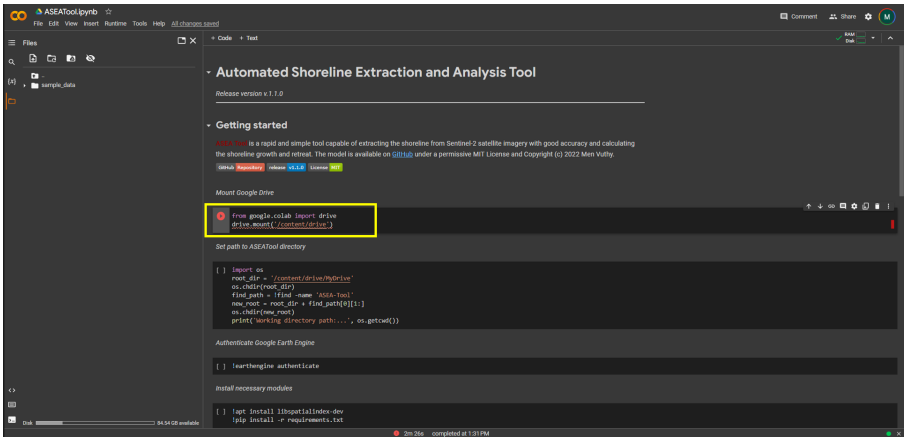
Figure 6: Interface of ASEA tool in Colab.

1.2 Mount drive

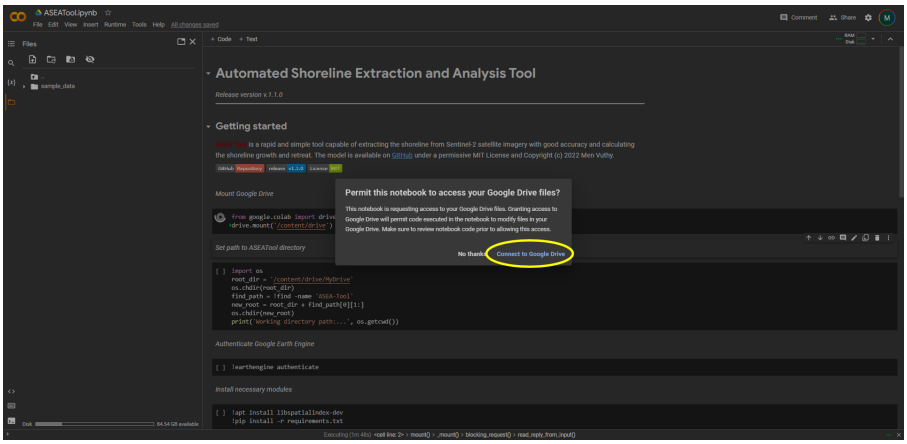
After opening ASEA model in Colab, it requires connection to Google Drive to access files and store output. There are a number of way you can connect Colab to drive; however, the

model here will do so by mounting Google drive in the runtime's virtual machine. The process to mount drive is as follows:

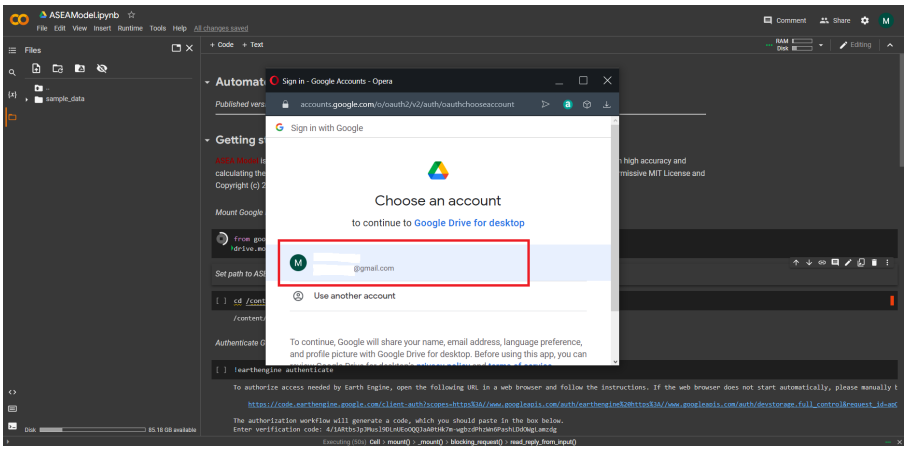
Step 1: Run the code below:



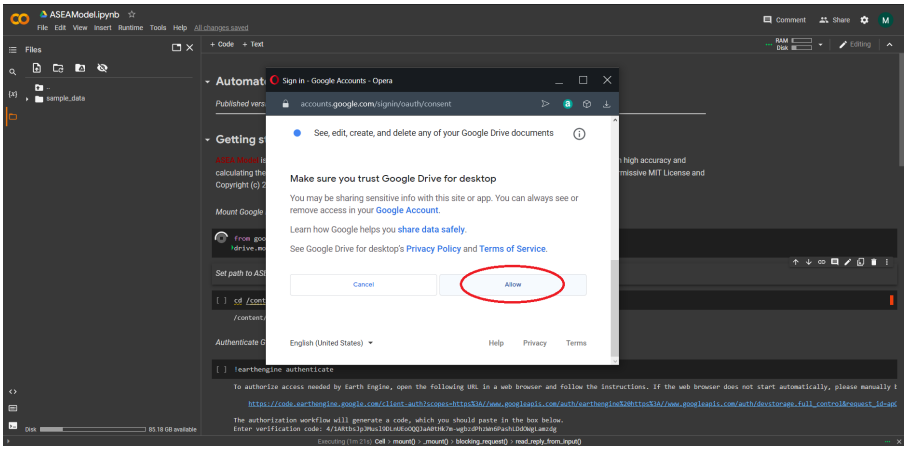
Step 2: Connect to Google drive:



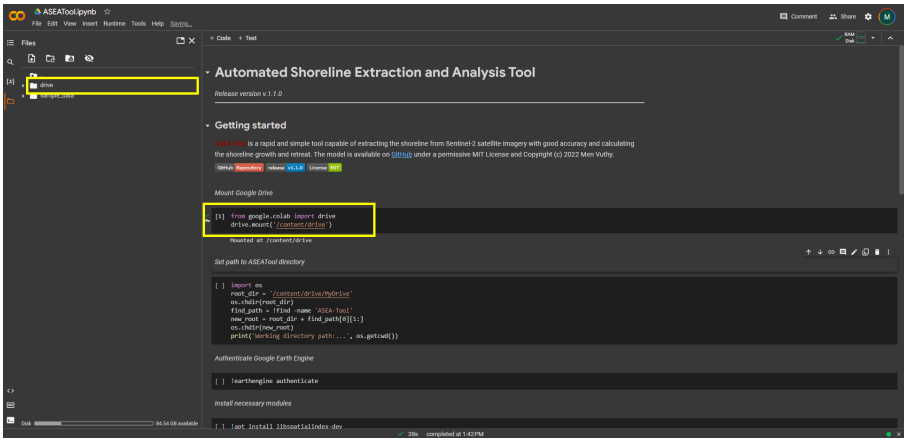
Step 3: Select Gmail for mounting drive:



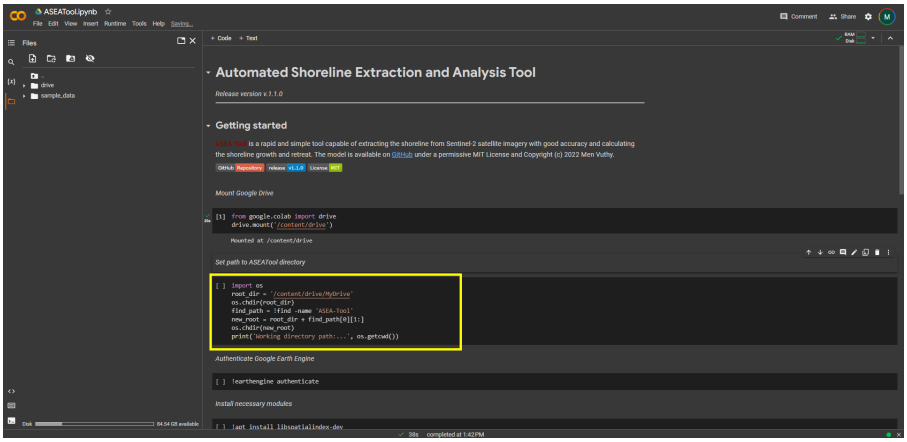
Step 4: Click “Allow” to trust Google Drive:



Step 5: Confirm connection by checking “drive” folder in the table of content and “green tick sign” next to code snippet:



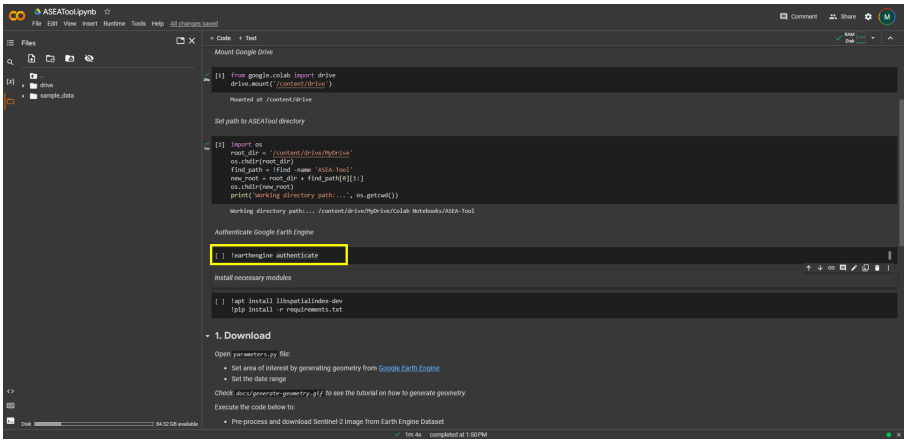
Step 6: Set path of working directory to ASEA-Tool by running the code snippet in yellow rectangle.



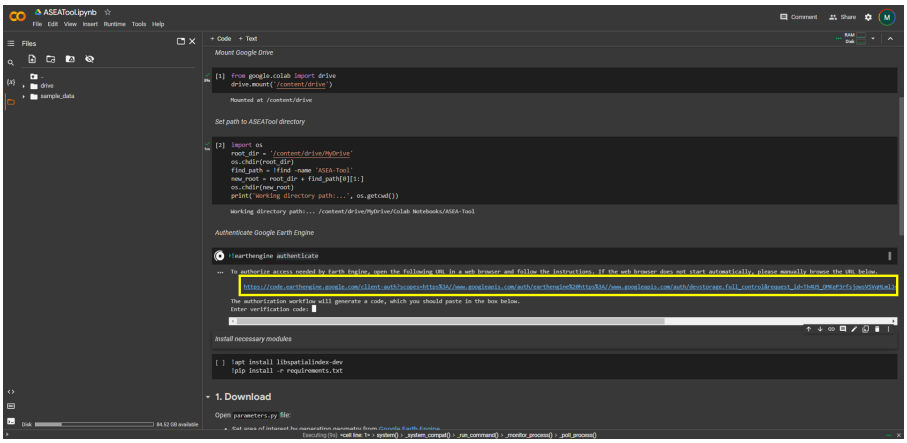
1.3 Authenticate Google Earth Engine

Google Earth Engine can be authenticated in Colab by running `!earthengine authenticate`. To authorize access needed by Earth Engine, open the given URL in a web browser and follow the instructions. The process to authenticate is as follows:

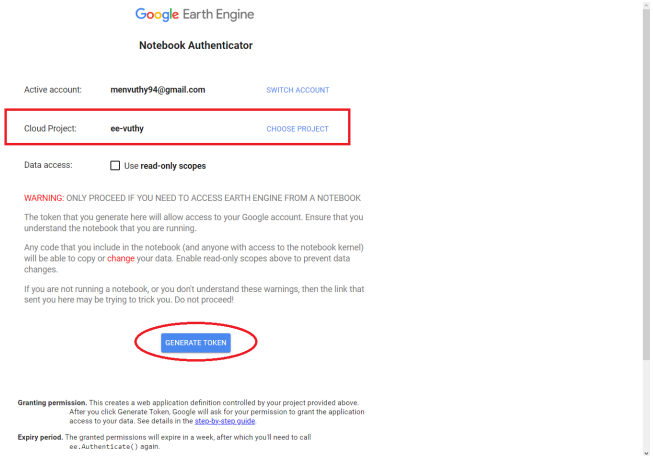
Step 1: Run `!earthengine authenticate` code snippet, and the URL will be given:



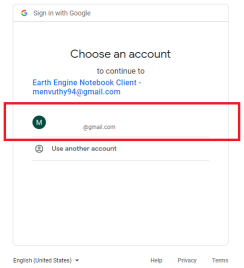
Step 2: Click to open the given URL:



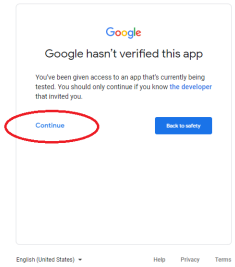
Step 3: Choose GEE cloud project, and click on “GENERATE TOKEN”:



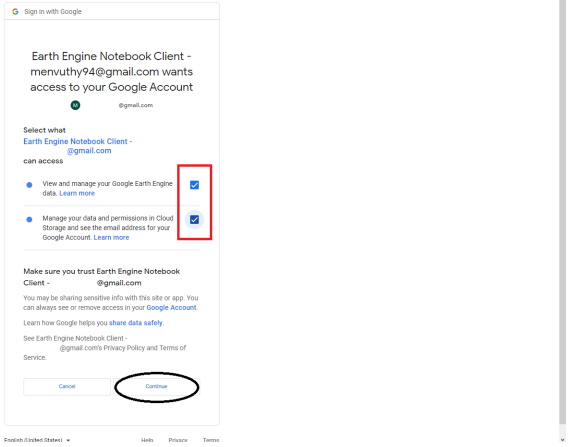
Step 4: Select Gmail account of Google Earth Engine:



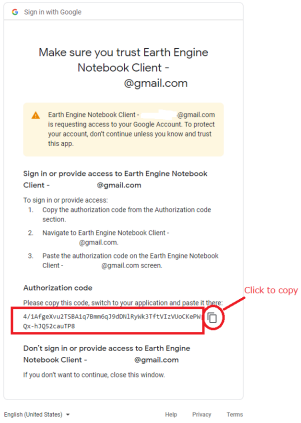
Step 5: Click “Continue” to verify the GEE app:



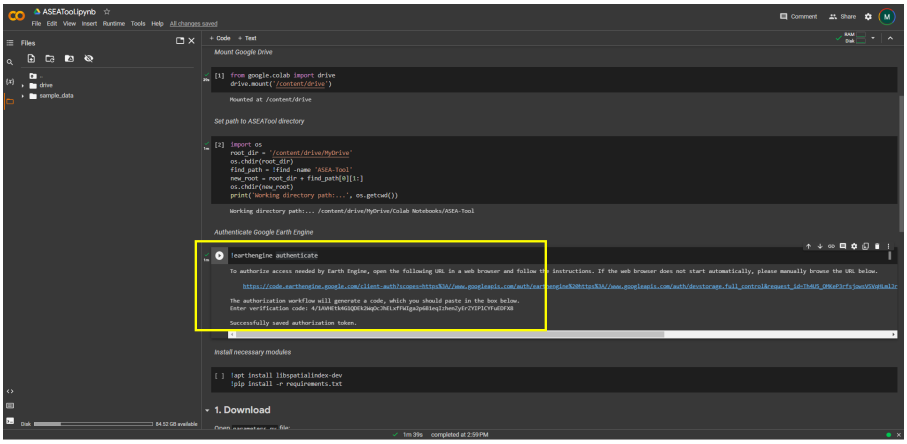
Step 6: Tick on the given two options and click “Continue”:



Step 7: The verification code is given. Copy it by clicking on the copy button in the red circle:

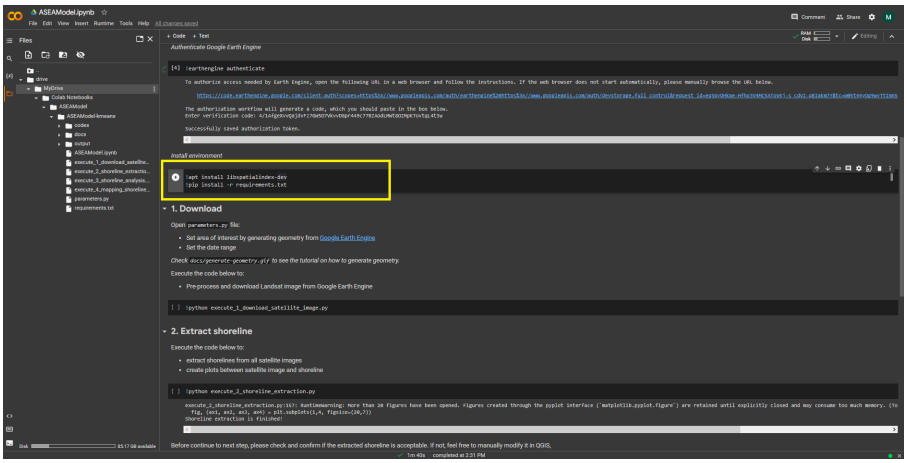


Step 8: Paste the verification code and Enter. After that, the connection will be successful:

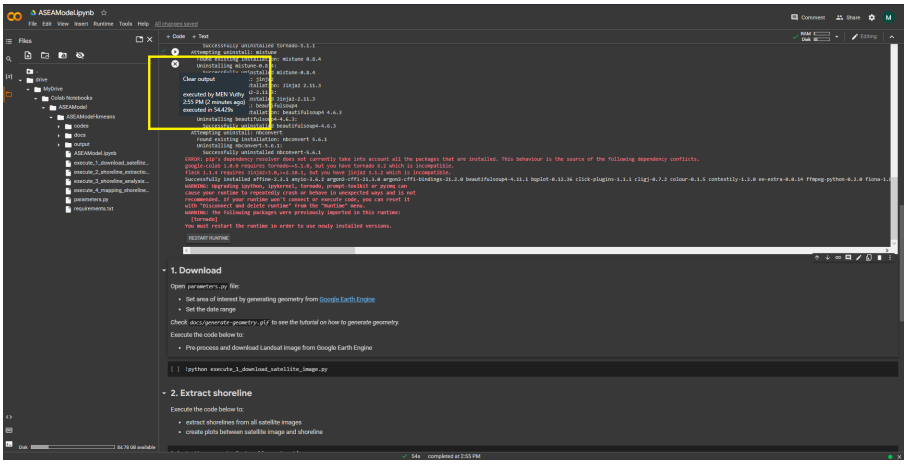


1.4 Install modules

Installing all necessary modules by running the code snippet as follows:



After running, the output will show the error warning in red color, but it is not a problem. It can be ignored by clicking on “X” sign to clear the output.



2 Download Sentinel-2 satellite image

To download Sentinel-2 satellite images from Google Earth Engine, the user has to give two main parameters: Area of interest and date range. Sentinel image consists of more than 10 bands but the model only downloads 5 bands: SWIR, NIR, Red, Green, and Blue since these bands are needed.

1. Area of Interest: can be generated in Code Editor as a Rectangle, multi-Rectangle, and a Polygon, or multi-Polygon. The model will not take input as Point and LineString. Example:

    - aoi = ee.Geometry.Polygon(
      [[[73.31901609299385, 0.2688094806410537],
      [73.31901609299385, 0.2445196397005524],
      [73.35609495041572, 0.2445196397005524],
      [73.35609495041572, 0.2688094806410537]]], None, False);

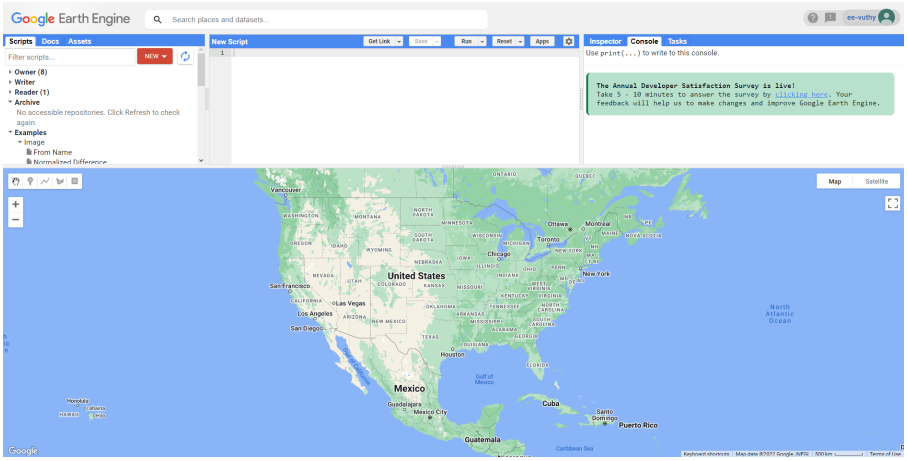
2. Date range: take start date and end date as input in the format of `YYYY-MM-DD`. In case of yearly shoreline analysis, it is suggested to set starting date and month from 01 Jan to 31 Dec, and change only target year. Example:

    - yearly analysis: date = [‘2022-01-01’, ‘2022-12-31’]
    - monthly analysis: date = [‘2022-01-01’, ‘2022-01-31’]
    - seasonal analysis: date = [‘2022-01-01’, ‘2022-05-31’]

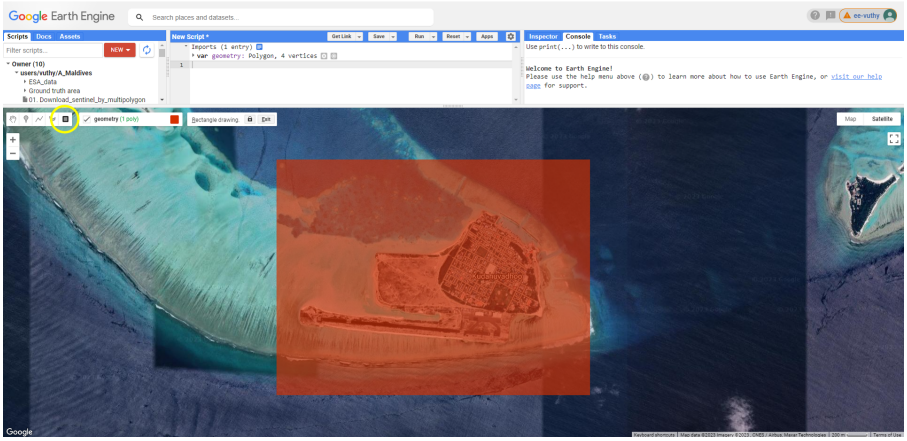
2.1 Generate AOI geometry

The model takes input of study area as geometry format of GEE JavaScript API. Hence, Code Editor will be used to generate AOI geometry. The process to create geometry for model input is as follows:

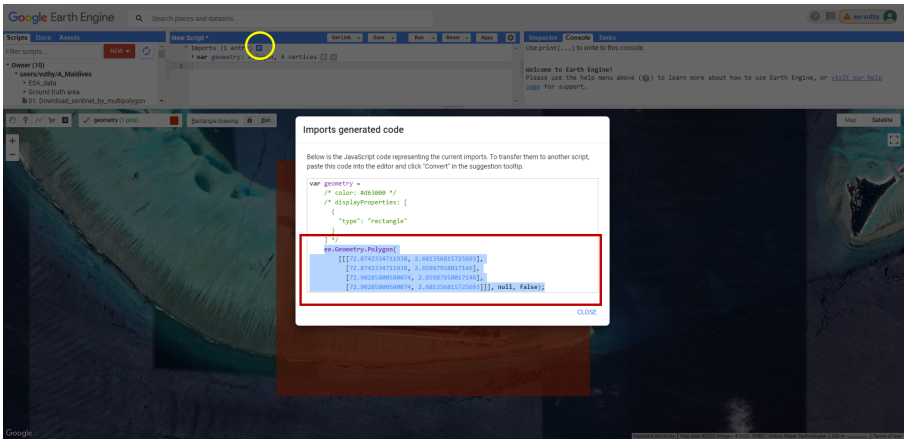
Step 1: Go to Code Editor:



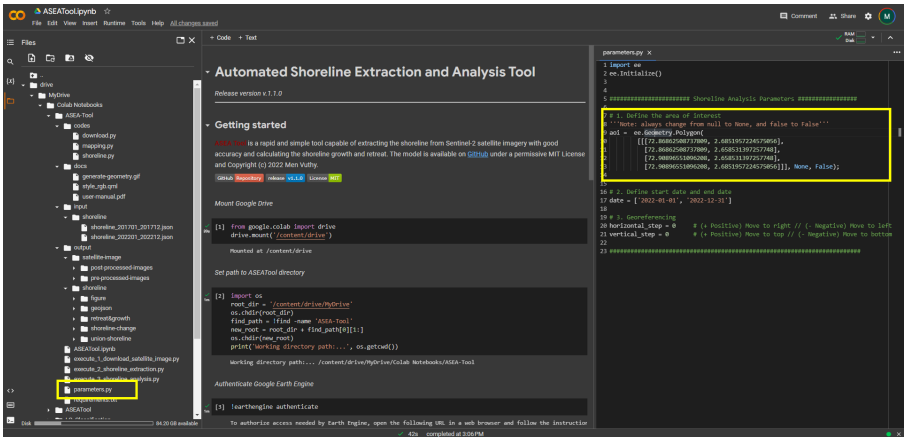
Step 2: Zoom to target area of interest, click on the rectangular shape (in yellow circle), and draw a rectangle covering AOI:



Step 3: Click on the blue button (in yellow circle), then the detail geometry of drawn rectangle will appear. Copy geometry detail in the highlight part:



Step 4: Open `parameter.py` and paste the copied geometry. After pasting, change from '`null, false`' to '`None, False`':



2.2 Download image

After giving AOI geometry input, it requires the date range, which indicates the number of years for shoreline analysis. In this example, we analyzed six years of shoreline change from 2017 to 2022. Hence, the date input in `parameter.py` is as follows:

- date = ['2017-01-01', '2017-12-31']
- date = ['2018-01-01', '2018-12-31']
- date = ['2019-01-01', '2019-12-31']
- date = ['2020-01-01', '2020-12-31']
- date = ['2021-01-01', '2021-12-31']

- date = [‘2022-01-01’, ‘2022-12-31’]

Each date range will be input to `parameter.py` file one by one. To download Sentinel image of the target AOI and date, run the code: `!python execute_1_download_satellite_image.py` as illustrated in Figure 7:

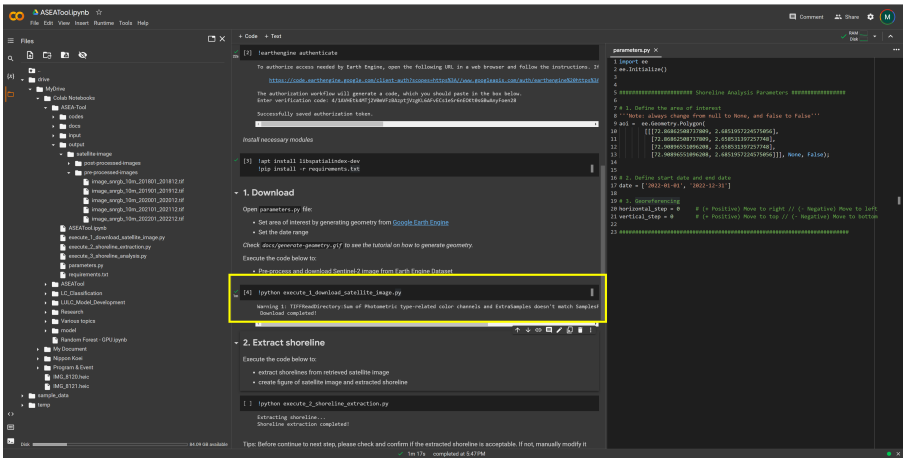


Figure 7: Executing code for downloading satellite image.

The downloaded image will be stored in the folder `output/satellite-image/pre-processed-images` as illustrated in Figure 8.

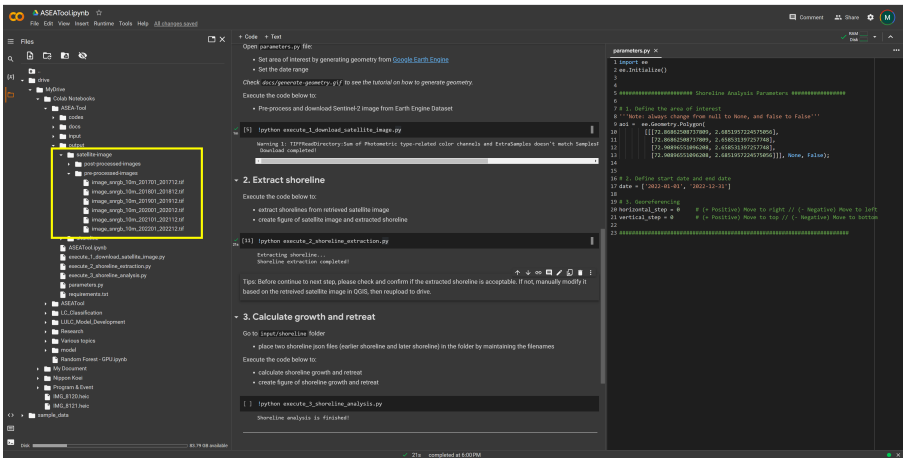


Figure 8: Output content after downloading.

3 Extract shoreline

After downloading, run `!python execute_2_shoreline_extraction.py` to extract shoreline and create figure as illustrated in Figure 9.

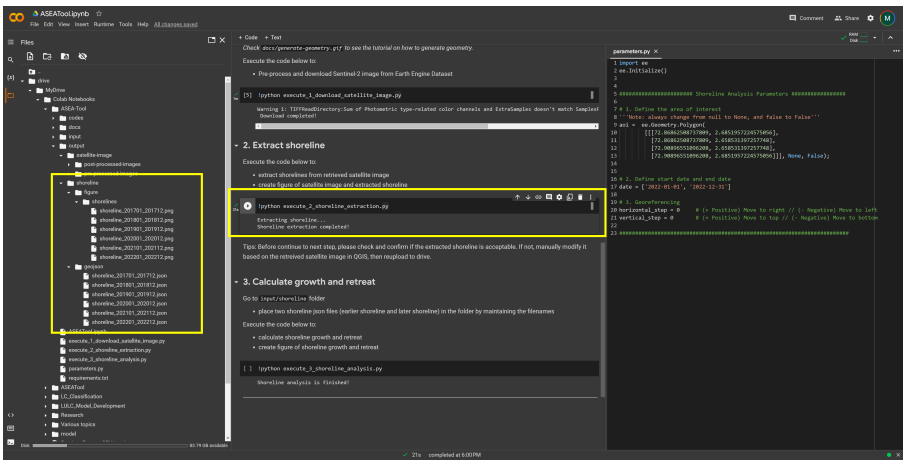


Figure 9: Executing code for extracting shoreline

Next, change to the next date range and re-run the download and extraction codes until we get all six shorelines. All shorelines are extracted in `.json` format, and they are stored in the `output` folder as can be seen in the left yellow box in Figure 9.

The output shorelines of each date are shown in the Figure 10 below.

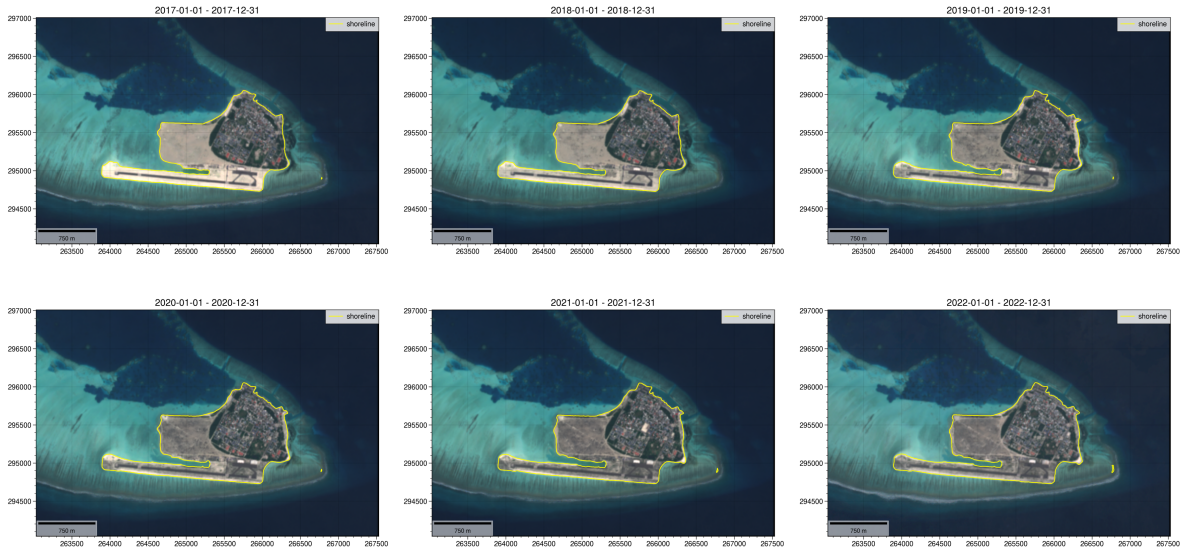


Figure 10: Extracted shorelines of all years.

4 Analyze shoreline

Shoreline growth and retreat are calculated by using `execute_3_shoreline_analysis.py`. Since there are 6 shorelines, the tool will calculate the difference between the earliest date and the latest date. In this case, 2017 is the earliest date, and 2022 is the latest date.

Extracted shorelines sometimes are not perfect due to some limitation caused by image quality, cloud cover, and algorithm. Therefore, we need to ensure it by opening it in GIS and manually adjust it. If everything's okay, move all corrected shorelines to `input/shoreline` folder. Then, run `!python execute_3_shoreline_analysis.py` to calculate shoreline growth and retreat between 2017 and 2022 as illustrated in Figure 11.

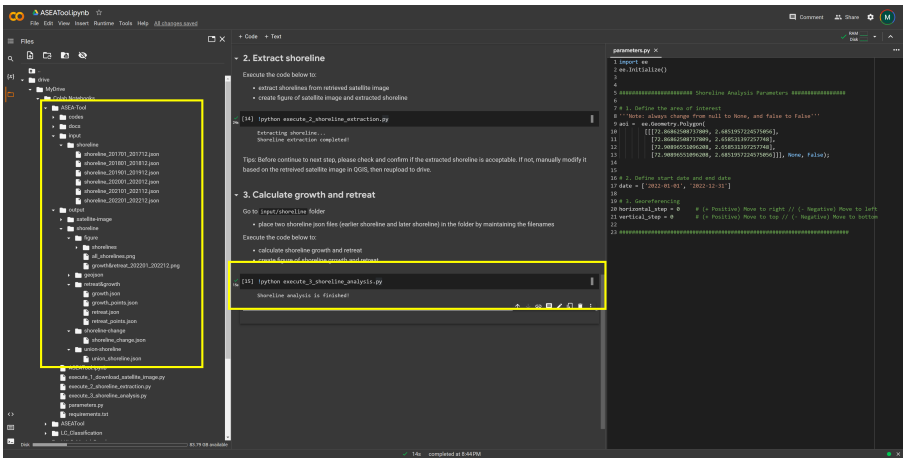


Figure 11: Executing code for analyzing shoreline

After code execution, 3 types of output and 2 figures are produced and stored in `output` folder as shown the left yellow box in Figure 11.

The results of shoreline analysis are shown in Figure below:

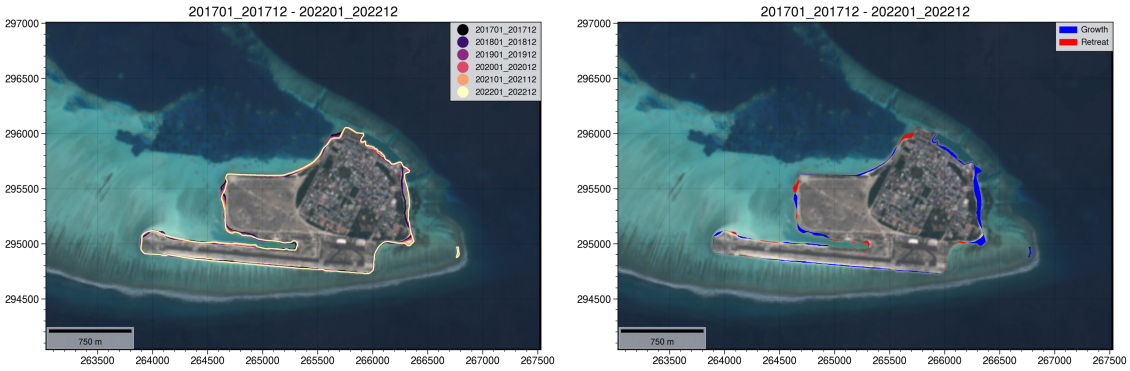


Figure 12: All extracted shorelines and shoreline growth and retreat from 2017 to 2022.

End of Document

Thank You