

Satellite Imagery Monitoring Documentation

This documentation will walk you through how to set up and use the code to monitor satellite images. All code has been commented to help make it readable, extra explanations and examples can be found in my thesis. If you need additional help please feel free to reach out to me, Allison Roush, at agr2022@vt.edu or 571-294-5890.

One-Time Set-up Tasks:

Planet Labs Account Set-up

- This analysis program uses satellite imagery from Planet Labs, to access this imagery you will need an account.
- Go to <https://www.planet.com/markets/education-and-research/> and apply for a research account (takes about a week to get approved)
- Once you have an account go to the “My Settings” tab and copy the API Key, you will need this to put in the code later
 - API_KEY: (insert here)
- Note: a basic research account allows for 5000 km² of data download each month

Google Cloud Storage (GCS) Account Set-up

- A GCS account is needed to store the images after they are requested from Planet Labs
- Follow instructions to create a GCS account
<https://support.google.com/cloud/answer/6250993?hl=en#zippy=%2Ccreate-buckets-to-hold-files>
- Create a new project and save the project id for later
 - GCS_PROJECT_ID: (insert here)
- Create a new bucket within the project to store all your images (sub folders will automatically be creating within this bucket for each new subscription)
 - GCS_BUCKET_NAME: (insert here)
- Next you need to change the permissions to allow read and write privileges
 - Go to “APIs & Services” -> “Credentials”
 - Click “Create credentials” -> “Service account key” -> “Create a new service account”
 - Create the account with “Storage Admin” role allowing read and write operations
 - Choose JSON as file type and click “Create”
 - This will automatically download the key file for the GCS account
 - Name this file and save it to the same directory that you want save your main code files in
 - GCS_CREDS_JSON: (insert file name here)

Get Secure Gmail Password


- To send emails automatically through your gmail account you need to set up an app password so the code can access your account
- Go to the google account you want to send emails from go to “Security” -> “2-Step Verification” -> “App Passwords”
- From “App Passwords” title a new app and a password will automatically be generated for you, save this password for future use
 - SECURE_APP_PASSWORD: (insert here)

Install All Required Packages for Code

- Download the “Pip_Installation_Packages.ipynb” code from the github and save to your directory
- Run this code once through Jupyter Notebook to install all necessary packages

Set-up Tasks for Each New Monitoring Area or Time-Frame:

Specify an Area of Interest (AOI) to Monitor

- Go to <https://geojson.io/#new&map=2/0/20> to create an AOI
- Use the world map to find the location you want to monitor then select the “Draw Polygon” tool  and select points to surround your AOI (max 200 points)
- Once you have selected all points hit enter on the keyboard, this will solidify your AOI turning it green and a JSON style file will be shown on the right side of the screen
- Click save at the top of the screen to save as GeoJSON file, save this to the same directory as the main code
 - GEOJSON_FILE: (insert file name here)

Create a Subscription Request

- Download the “Create_Subscription.ipynb” code from the github and save to your directory
- Open the file in Jupyter Notebooks
- Copy and paste required information in the top code segment, be sure to keep the quotation marks
 - API_KEY:
 - GCS_PROJECT_ID:
 - GCS_BUCKET_NAME:
 - GCS_CREDS_JSON:
 - GEOJSON_FILE:

- Set start date and end date for monitoring
 - Format: YYYY-MM-DD
 - START_DATE:
 - END_DATE:
- Give a name for the subscription
 - SUB_NAME:
- Once all variables have been added run all cells in Jupyter Notebooks to create the subscription (be sure to only run it once)
- Copy the outputted Order ID that is displayed when the code is run, this will need to be saved for the next step
 - ORDER_ID: (copy here for future use)
- Below the main code there are multiple code blocks that are commented out, these can be uncommented to edit your subscription, cancel your subscription, or view the status of all running subscriptions on your account

Set-up Analysis Code

- Download the “Satellite_Imagery_Analysis.ipynb” code from the github and save to your directory
- Open the file in Jupyter Notebooks
- Copy and paste required information in the top code segment, be sure to keep the quotation marks
 - GCS_BUCKET_NAME:
 - GCS_CREDS_JSON:
 - ORDER_ID:
 - Set-up who the notifications will be sent to and who they will come from, include the app password you generated for the sending gmail account
 - MAIL_FROM: (insert gmail address)
 - SECURE_APP_PASSWORD:
 - MAIL_TO: (insert gmail address)
 - Can choose to change threshold for significant change or leave preset values
- Download the “Satellite_Imagery_Analysis.ipynb” code as a .py file from Jupyter Notebooks once the code has been updated
 - File->Save and Export Notebook As.... ->Executable Script
- Save this downloaded .py file in the same directory as the other files

Set-up Analysis to Run Autonomously

- To set-up the analysis code to run autonomously windows task scheduler is used
- On your computer search “Task Scheduler” and open the program
- Select “Create Task”

- In the “General” tab give the task a name such as “Satellite Analysis” and give the task a description
 - In security options select “Run only when user is logged on”
 - Also check “Run with highest privileges”
- Open the “Triggers” tab
 - Create a new trigger
 - Begin the task: “At log on” this will run the task everytime the computer is logged in
 - Select “ok”
 - Create another trigger
 - Begin the task: “On a schedule”
 - Select “Daily” and choose the execution time (note the code will only run if the computer is turned on)
 - Select “ok”
- Open the “Actions” tab
 - Create a new action
 - Action: “start a program”
 - Enter the directory to pythonw.exe in the Program/script box (this allows the code to be run in the background)
 - Enter “Satellite_Imagery_Analysis.py” in the Add arguments box
 - Enter the directory to your file in the Start in box
 - Select “ok”
- Open the “Conditions” tab
 - Uncheck “start the task only if the computer is on AC power”
- Open the “Settings” tab
 - Check the box to “run task as soon as possible after a scheduled start is missed”
- Select “ok” to finalize and create the task
- Now the analysis is all set up and will run automatically