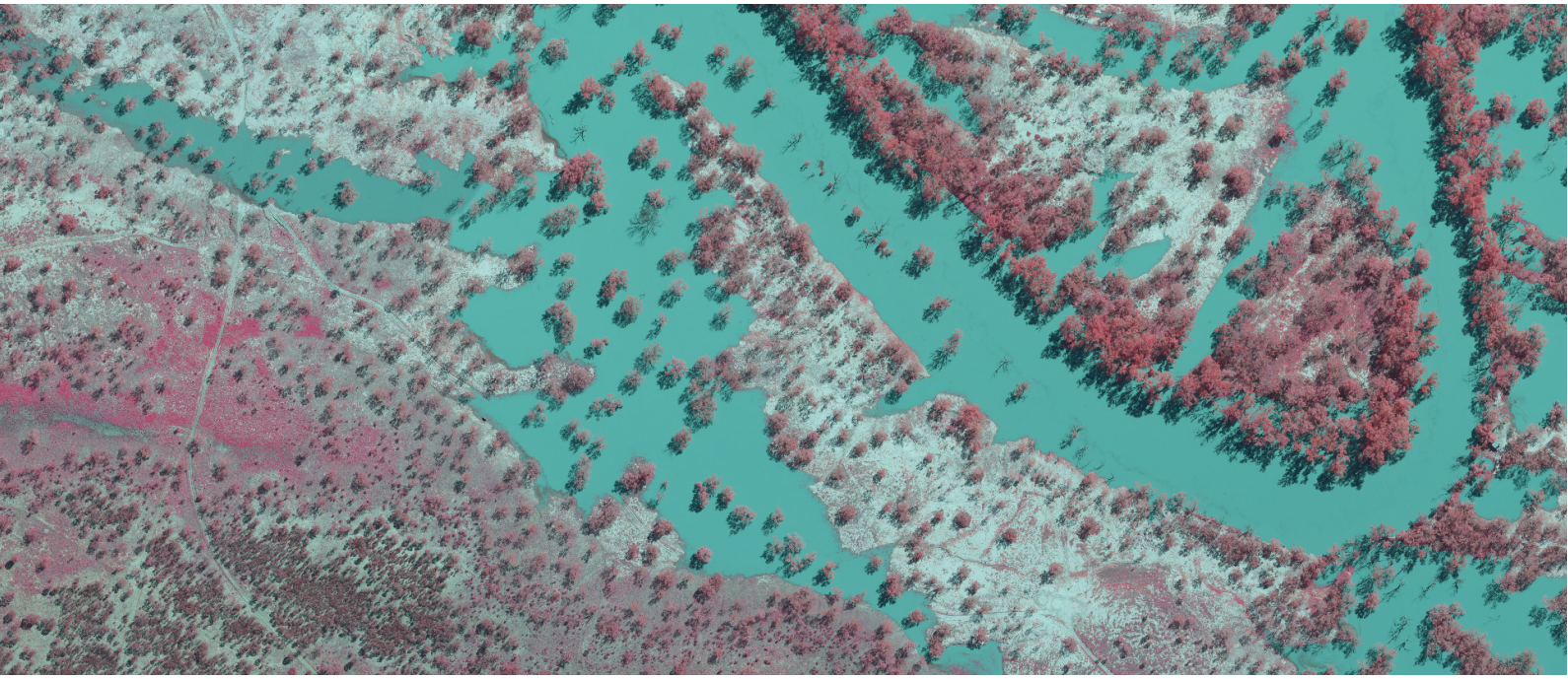
G5

PROJECT USER MANUAL



Gaussian Mixture Model (GMM) Pipeline

Aerial Imagery Initiative

Contents

Flood Extent Extraction NoteBook User Manual

Gaussian Mixture Model (GMM) method

Introduction

This manual describes the deployment and usage of the Flood Extent Extraction (FEE) notebooks developed for the Department Of Customer Services (DCS). This manual contains guides on how to download and execute said Notebooks within the Amazon Web Services AWS Sagemaker Environment. This manual will also cover the expected input data type and outputs that are configured.

Assumptions

Because of the nature of the FEE system the following assumptions have been made about the user as well as the ability to access private storage systems (bitbucket, AWS S3) used to host the data and executable Notebooks. While many steps are covered some details may be missing such as logging into AWS / Bitbucket, navigating AWS / Bitbucket. etc

- User has access to an AWS account
- User has access to an AWS S3 bucket containing flood imagery for analysis
- User understands how to start / stop Sagemaker Notebooks
- User understands how to open Sagemaker JupyterLab Notebook
- User has access to Bitbucket repository hosting required notebooks
- User understands cloning of Bitbucket repositories
- User is computer literate and has some programming experience (ideally python)

Overview

This manual will cover the following components in higher detail.

Image / Data inputs

Flood data to be processed is expected to be JPEG2000 format images with Near Infra-Red Red Green (NRG) colour channels in the respective Channel 0 = Near Infra-Red, Channel 1 = Red, and Channel 2 = Green. Images need to have geodetic information embedded in the images metadata, specifically pixel size, projection datum, image location in relation to datum. Often these images are exported from GIS systems that already contain the required geodetic information in the image metadata.

Environment

The FEE has been developed and tested with the AWS Sagemaker environment. The Sagemaker environment is primarily a preconfigured Conda and python environment in AWS. The execution Environment stack is as follows:

- AWS Sagemaker
- AWS Instance (Linux based host)
- Python / Conda Environment
- Executable Jupyter Notebook

FEE Notebook execution steps

The complete start to stop process for operating the FEE can be described in the following steps:

1. Starting a correctly specified Sagemaker Notebook
2. Cloning the FEE Bitbucket repository
3. Configuring the Notebook
4. Executing the Notebook
5. Examine the output
6. Shut down the Notebook instance

Standard and high probability data

During the data extraction process the GMM will produce the probability of a pixel belonging to any of the clusters contained in the GMM model. This allows us to produce 2 data sets;

standard probability flood extent: Contains the default GMM pixel. Ie the pixel's highest probability is a known flood cluster.

high probability flood extent: Contains the default GMM pixel clustering output and pixels that have a high probability of belonging to a flood cluster. Ie the pixel's highest probability is a known flood cluster, and the pixel also has a high probability of belonging to a flood cluster.

The high probability flood cluster is aggressive compared to the alternative and may be able to include areas missed in the standard probability but may also result in false positive identification of the flood area.

Both of the datasets are contained in the Shapefile output from the Notebook and can be filtered by tag in GIS software.

Prerequisites

- Access to the Bitbucket repository containing the required Notebook: https://bitbucket.org/csu-spatialservices/flood-extent-extraction.git
- Authorised access to AWS and AWS Sagemaker
- Compatible JPEG2000 Images hosted on AWS S3 and access to said images from same AWS Sagemaker instance

Sagemaker Notebook

Notebook instance type

A Sagemaker Notebook will need to be started within the AWS environment. The Notebook Instance type best suited to the GMM process is a memory optimised instance. This is not because of GMM specifically high in memory usage, but opening JP2 images uses a lot of memory.

The suggested Sagemaker Notebook instances are as follows:

ml.r5.24xlarge (memory optimised)
Tested on whole flood regions of up to 179590px * 72510px (13 Giga Pixel)

ml.r5.4xlarge (memory optimised)
Tested on tiled flood region sets of up to 144 images of 10000px * 10000px

Volume size

It is recommended to configure the instance to have at least **10GB of local storage**. It is unlikely that all of this will be used however it does provide storage in case of larger datasets.

Sagemaker Notebook Example

Instance suitable for large single images.

Notebook instance name	GMMNoteBook
Notebook instance type	ml.r5.24xlarge
Lifecycle configuration	None
Volume size	10gb

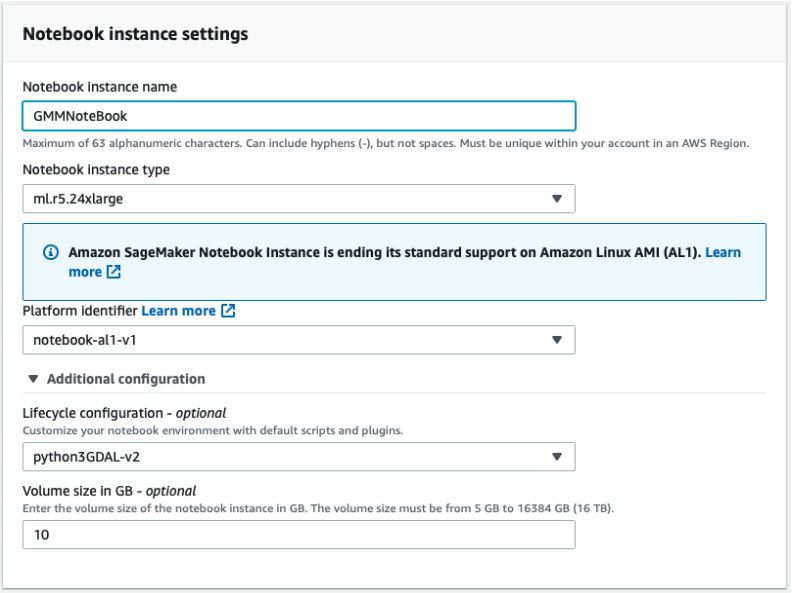


Figure 1 - Example Sagemaker instance

Clone the repository

All functional notebook code is held within the following repository within Bitbucket at the following repository URL:

https://bitbucket.org/csu-spatialservices/flood-extent-extraction.git

This repository will need to be cloned to the Sagemaker Notebook / Instance. The easiest method is to use the inbuilt Notebook Git tools access from the left menu Git icon and selecting “Clone a repository”. Once prompted enter the repository URL and fill in your Bitbucket credentials to access the repository.

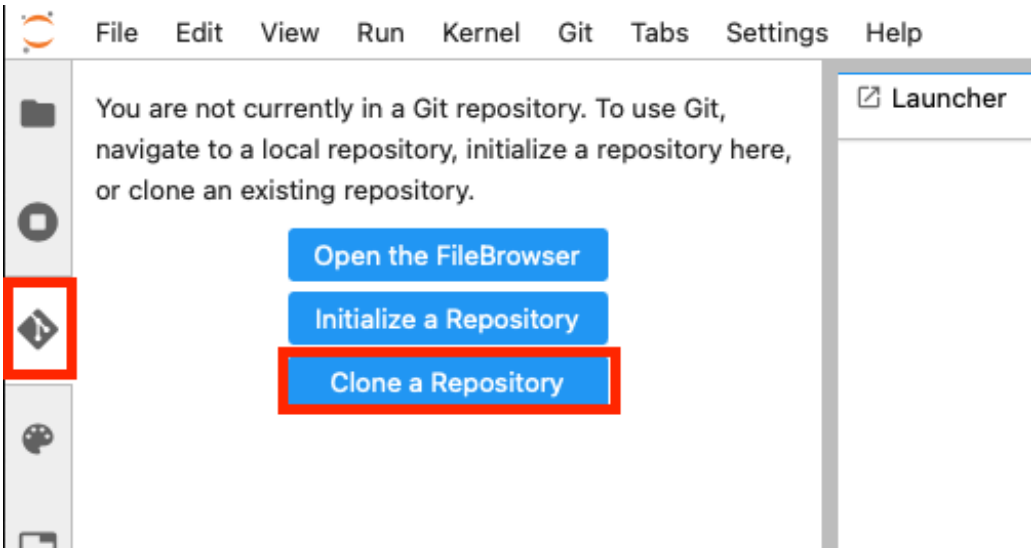


Figure 2 - Accessing Git clone button

Repository Folder Structure

The repository will have the below folder structure. Most folders are empty and are intended to be default locations for runtime data (Images = Flood images, Logs = Runtime Logs of the notebook, Out = Shapefile outputs. etc). These will be the preconfigured locations used by the notebook during execution

```
flood-extent-extraction/
├── App/
│   ├── GMM Extraction Method/
├── Documentation/
├── Images/
├── Logs/
├── Models/
│   ├── GMM/
├── Out/
```

The 3 directories of most interest will be:
App – Contains executable Notebooks. For this document we will be looking at GMM Extraction Method
Documentation – Documents such as this
Models – Machine learning models and artifacts

Notebook Setup and Execution

To execute the GMM clustering technique notebook can be found at **flood-extent-extraction/App/GMM Extraction Method/GMM_Flood_Extent_Extraction.ipynb**. The default settings should be suitable for the execution of the notebook. However they should be checked to prevent errors during runtime.

Jupyter Kernel

Select **conda-python3** as the kernel for the Jupyter Notebook.

Python modules

The conda environment will also need additional modules for the notebook to run. This is completed by executing the last cell in **Imports and system setup.** This process can take some time (~25mins).

```
#conda commands for environment setup
%conda install -c conda-forge gdal fiona rasterio

#pip commands in case conda cant get the required packages
!pip install 'opencv-python>=4.5.3.56' # required as older versions fail opening some jp2 images
```

Figure 3 - Cell containing module install commands

Notebook settings

All user settings are contained in 1 cell “Notebook settings”. Below is an explanation of each setting. A brief explanation of each explanation is also contained in the Notebook.

The only setting that is likely to be changed is ***nrg_image_s3_source*** as it specifies the source S3 bucket of flood images.

- Input Image settings
    - **nrg_image_s3_source** - String: Amazon S3 source directory containing flood images destined for processing eg. s3://ss-csu-dataset/raw/Brewarrina_Flood_2021_04_15cm_NRG/
    - **nrg_image_storage_directory** - String: Storage directory where flood images will be stored for processing.
    - **image_scale** - Float: Reduce image size. This is used to boost performance. This should be set to a value approximate to the scale of images used to train the GMM model.
- GMM settings
    - **gmm_storage_directory** - String: Directory containing the pre trained GMM model
    - **gmm_flood_clusters** - Tuple: Clusters that contain flood pixels
- Contouring / polygon settings
    - **minimum_contour_size** - Int: Minimum pixel area for a contour / polygon to be considered valid
- Logging settings
    - **log_file_prefix** - String: Prefix that will be attached to all log files.
    - **log_storage_directory** - String: Location to store log files
- Output settings
    - **raster_shp_output_prefix** - String: Prefix for output shape and raster files. Output files names will also include date and time the file was created.
    - **output_directory** - String: Location to store shape files, inspection and zip file.

Execution

Once ***nrg_image_s3_source*** has been set to the source S3 bucket for the flood images the Notebook can be executed (Run -> Run All Cells).

Zip file output

After execution has successfully completed the ***output_directory*** will be zipped and saved to the parent directory of ***output_directory***. Eg "../***output_directory***"

Runtime outputs

Logging

During runtime the notebook will update the user via output below the executing cell. This output is limited and provides an overview of the Notebook state. A more detailed log is generated and output to the folder defined in ***log_storage_directory***, this is set to the *Logs* folder by default. An additional EXTRA logfile is also generated, this EXTRA log file contains logs of the Notebook and all imported modules. This can be used to very in depth troubleshooting of the notebook.

Raster and Shapefiles

Once the notebook has been successfully executed the flood extent will be output in the following formats:

JP2 Raster - A JP2 grayscale raster file is generated with 2 pixel values:

1 = Flood area

0 = Non Flood area

These images will contain geodetic information and will render correctly in GIS software. Opening in the image in other image software may produce a blank image as the pixel values may be rendered on a scale of 0-255.

ShapeFile - Flood extent will also be output in an ESRI ShapeFile.

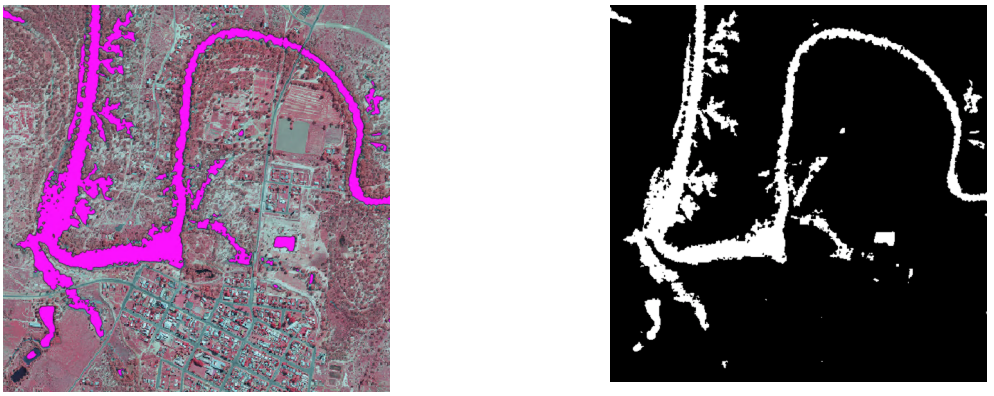


Figure 3 - Example shape file (left) and raster file (right)

Each flood image will result in a Shapefile output. The Shapefile output is a multipolygon that contains 2 data tags:

standard probability flood extent: Contains the default GMM pixel. I.e. the pixels highest probability is a known flood cluster.

high probability flood extent: Contains the default GMM pixel clustering output and pixels that have a high probability of belonging to a flood cluster. I.e. the pixels highest probability is a known flood cluster, and the pixel also has a high probability of belonging to a flood cluster.

The high probability flood cluster is aggressive compared to the alternative and may be able to include areas missed in the standard probability but may also result in false positive identification of the flood area.

GIS software such as QGIS allows the Shapefile to be filtered. By default, all polygons are shown however QGIS allows the user to filter the polygon by tag. Using this filter feature the user can switch between standard and high probability data.

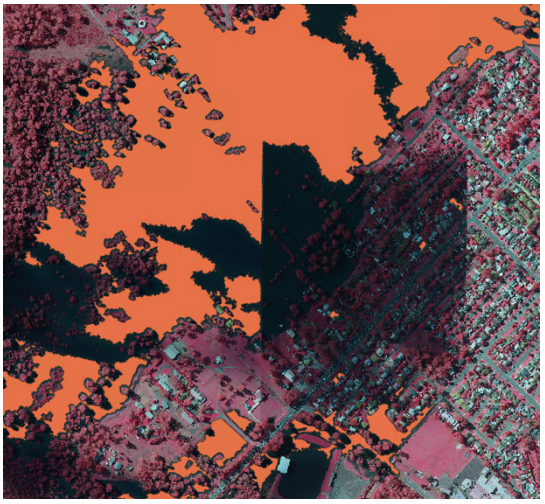


Figure 5 - Standard Probability

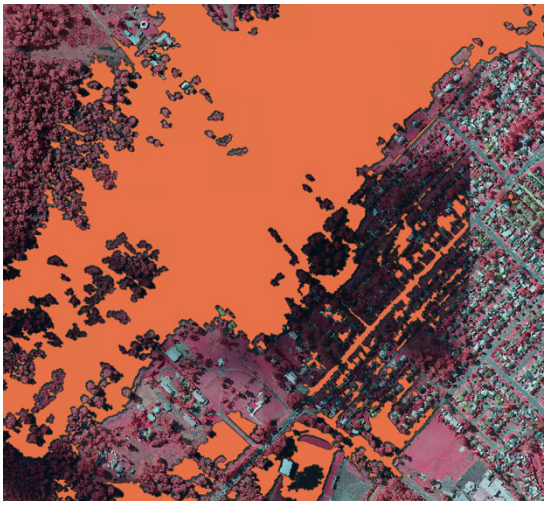


Figure 6 - High Probability

Clean up

At the end of the Notebook execution the system will automatically delete flood images and zip and save the ***output_directory.*** This is to ensure multiple runs of the notebook does not cause overfilling the local disk space and to ensure no old images are accidently processed when attempting to process new images.

The zipped outputs will be saved to the parent directory of ***output_directory.***

Note: Remember to Stop the Notebook after Execution to prevent a large AWS bill