Architecture Notebook

Aerial Imagery Initiative

Index

Purpose

In the initial phase of the project the system or project will mostly consist of researching and training a SageMaker instance / model on AWS to process LiDAR imagery and determine flood level and or flooded areas. At this point it is not entirely clear if the system is to be able to distinguish actual flood levels of a current event or determine the maximum height a flood level reached, or both.
DCS has informed that the primary objective is to develop a method to prepare data and train the model. It is unclear if there will be further phases if / when the model is trained and ready for production. However, if time allows an abstracted method to interact with the model may be developed such as an API or web app.
The training data and real-world data is expected to be very large files in JPEG 2000 format. Early suggestions are that files will be in the area of 10GB. Because of this it is likely that files will be stored in and retrieved from S3 buckets. The file size will be of concern when performing feature extraction on the images. One method may be to reduce the image size for feature extraction and have the feature data applied to the full-size file. Eg. Interactively draw a polygon around the flood area on a local desktop and have that image coordinate data applied to the full size image. This will allow files to be worked on locally however this will need to be cleared for any security concerns regarding where the data can and can't be stored.
SageMaker makes use of Jupyter notebooks to interact with the SageMaker environment. The handling and manipulation of data for preparation of training the model will be mostly performed within Jupyter Notebooks however some feature extraction maybe performed locally if methods and tooling can be developed for

Architectural goals and philosophy

For the most part the system has been described as a SageMaker instance. The instance will have a trained machine learning model that is able to classify flood/ed areas on LiDRAR imagery. Interaction with the instance will be via Jupyter notebooks. Using Jupyter notebooks the user will be able to prepare data for processing by the SageMaker instance. Raw data will be uploaded to S3 buckets where it can be retrieved by Jupyter notebooks. It is unclear if the data produced by the image will be another image with annotated flood areas or coordinate data that can be drawn / overlayed on the original image.
The SageMaker instance and Jupyter notebooks will reside within the DCS AWS environment. This environment is controlled and managed by DCS with access only given to authorized and trained personnel so it is not planned that security and access control will not be in scope for this project.

In summary the overall architecture will be relatively simple. A SageMaker instance running. Trained model, raw data and output will be stored on S3 buckets and Jupyter notebooks used to manipulate data into and out of the SageMaker instance.

Assumptions and dependencies

AWS infrastructure – It is assumed that the AWS environment is the best solution for this. The assumption based on the reputation of the AWS environment and that DCS has already make significant investment in AWS infrastructure

SageMaker – Assumptions have been made that SageMaker has the ability to process and be trained for the required result of determining flood/ed areas within LiDAR imagery. Processing of LiDAR imagery is expected to be computationally expensive, and it is currently assumed that a reasonable time complexity is possible. It is also assumed that SageMaker has the required Machine Learning algorithms that are suitable for the application.

Skills / Ability – The skill and ability to learn are a dependency for the projects success. The team are not Machine Learning experts and have next to no experience. However, the team does have direct access to ML experts in the form of David Tein and Intellify consultants.

Time – The CSU project students are available to work on the project till the end of the last session of the CSU calendar. It is being assumed that all work will be completed and handed over to DCS before the date of 29th Oct' 2021.

Architecturally significant requirements

- Identify Flood/ed areas within provided imagery
- Output is useable with little to no manipulation
- System is operational within the AWS environment
- Ability to input/upload new imagery for classification
- System uses resources that are financially viable

Decisions, constraints, and justifications

Majority of architecture decisions have been made by DCS. However, many of these can be related to the following.

- Time. Group 5 put forward the possibility of wrapping the SageMaker instance in an API to offer a level of abstractions for interacting with the instance. However, this was denied and suggested that during the initial phase that interaction with the instance via Jupyter notebooks will be sufficient.
- Unknown data delivery methods, because it is currently unknown exactly how feature extraction will be performed or how data will be fed into the SageMaker instance. Defining complex or abstracted architecture may prove to be wrong and/or consume time that can be used for other activities more closely related to building a successful model.

Architectural Mechanisms

- Jupyter Notebook instances will be used to manipulate the data required for input into the machine Learning model and analyses of the results. Jupyter will be used at all stages of creation and utilization of the Machine learning model.
    - During training it will be used to retrieve and organize data, place data into locations accessible by the machine learning model, review and analyze results of the model after training runs.
    - Once the model has reached a satisfactory level of performance Jupyter notebooks will also be used to interact with the model by organizing and providing data into the model and retrieving results.

- SageMaker Machine Learning Instance - It is imagined that a SageMaker Machine learning instance can be trained for the goal of identifying flood/ed areas. Once the model has been trained it will be launched as an instance and accessible within the AWS environment.

Key abstractions

- Machine learning model Abstracted by SageMaker Instance. The trained machine learning model will be abstracted by a SageMaker instance. The SageMaker instance will have the ability to have data sent via HTTP POST and response will be in text format.

Layers or architectural framework

- Interface / Interaction – Jupyter NoteBooks
- Machine Learning Model – SageMaker Endpoint
- Storage – S3 Bucket

Architectural views

Logically all layers can be combined and referred to a SageMaker studio. SageMaker studio is a suite of mechanisms that are combined for the purpose of data science. This suite includes but is not limited to Jupyter Notebooks, SageMaker Instance Creation, Running of training experiments .etc