A Small Satellite in Low Earth Orbit: Machine Learning, Image Compression, and Reconstruction using Generative AI

Anonymous WACV Algorithms Track submission

Paper ID *****

Abstract

Advances in sensing technology enable orbiting satellites to collect high-frequency multi- or hyperspectral imagery from large areas. We describe how Amazon Web Services Greengrass, an IoT cloud service, managed the deployment, operation, and update of software and machine learning components onboard a small satellite in low-earth orbit. We focus on the processing of multispectral imagery and quantify the extent to which lossy compression techniques affect the performance of machine learning algorithms applied to the downloaded imagery data. We further describe how generative AI techniques, such as image upscaling, can further assist in data compression and post-download image reconstruction.

1. Introduction

In recent years, the development of launch capacity and the reduction in prices have led to a surge in the number of small satellites deployed in orbit. In addition, it has become increasingly common for multiple users to share access to the sensors and computational resources onboard the satellite, enabling greater availability of satellite services.

Low Earth Orbit (LEO) satellites move at extremely high speeds relative to the Earth's surface, completing an orbit in approximately 90 to 120 minutes. This means they are only within the line of sight of a ground station for a short period, often in the range of a few minutes during which several software processes and radio communication must complete.

Amazon Web Services (AWS) recently partnered with aerospace technology companies D-Orbit and Unibap to manage software and process data directly on a satellite in orbit, addressing these challenges using techniques and approaches from the Internet of Things (IoT) area. AWS IoT Greengrass [4] is a service that extends AWS's cloud capabilities to local devices, allowing them to collect and analyze data closer to the source of information, while also securely handling communications between each device and the cloud. IoT systems require quick responses to local events, intermittent connections, and low-cost data transmission to the cloud. AWS Greengrass makes it possible to carry out machine learning inference tasks on Greengrass-enabled devices, which is especially useful when the round-trip to the cloud is too slow or unreliable. After being trained in the cloud, the machine learning models are deployed directly to the IoT device, allowing for real-time predictions without needing to be constantly connected to the cloud.

Satellites in low-earth orbit have limited and intermittent radio contact with the ground, and the data collected is often large. To address this, recent advances in machine learning have enabled the use of computational methods to analyze geospatial data on a large scale. However, these techniques often require a lot of computing power, which makes them unsuitable for use on small satellites. We present fast and reliable supervised and unsupervised techniques that can be used within time and computational limits. We also discuss pipelines for onboard data collection and scheduling retraining of the algorithms. The techniques we use include Principal Component Analysis (PCA) and Linear Discriminant Analysis (LDA), both of which have closed-form solutions that guarantee rapid convergence.

The availability of open source data for geospatial data analysis is essential for the advancement and testing of machine learning models. To illustrate this, we use the Landsat8 image dataset and its image segmentation masks to demonstrate how different methods of processing and compressing the data can lead to different results when the downloaded data are processed further on the ground. Using Generative AI techniques can improve the ground-based processing of downloaded images. Projecting the data onto a small number of PCA or LDA components, as well as downsampling the images to smaller sizes prior to transmission to the ground, can increase image compression. Once on the ground, generative techniques such as upscaling can be used to forther enhance the downloaded images.

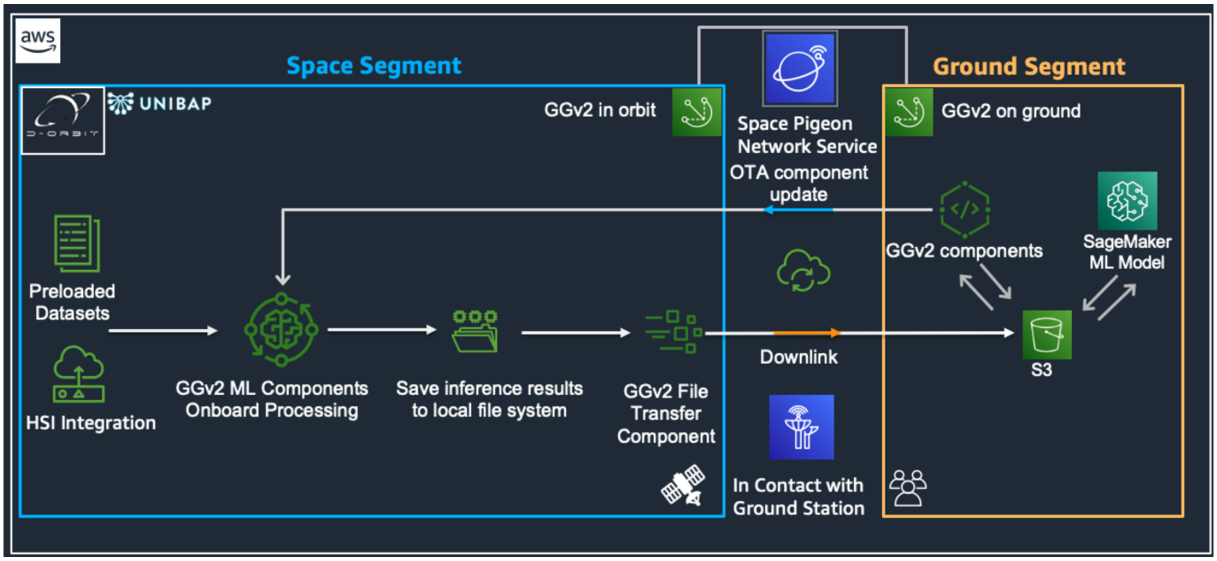

Figure 1. High level architecture of the Greengrass ML components

This paper is organized as follows: first, we discuss the overall architecture and capabilities of the AWS Greengrass in relation to several machine learning components deployed onboard; second, we focus on the image processing and compression utilized, their accuracy metrics, and we highlight their relative advantages and shortcomings; finally, we suggest future areas of research and potential improvements.

2. Deployment of Machine Learning Models as Greengrass Components

AWS and its partners developed a software prototype that included multiple machine learning models packaged as AWS Greengrass components. Unibap, a Swedish high-tech company and AWS Partner, built a space-qualified processing payload that was integrated with the software prototype. This payload was then mounted on a D-Orbit ION satellite and launched into space. On 21 January 2022, the team made contact with the payload and sent the first remote command from Earth to space using the AWS infrastructure. Subsequent experiments were conducted over the following months. Models were designed for real-time satellite imagery, to detect anomalies in internal sensor data, and Greengrass AWS IoT components for cloud management and analytics even when there is limited connectivity. A high-level architecture diagram is shown in Figure 1. We prepared for the satellite launch by training image processing models. The tasks also included requesting the image data from the onboard multispectral imager, compressing the data, and queuing the processed imagery for download. AWS Greengrass commands enabled us to retrain the compression and outlier detection algorithms based on data collected and stored onboard for a period of time specified by the ground operator. Furthermore, we can create new ML components and upload model files from the ground while the satellite is in orbit.

3. Image Compression Algorithms

Our initial algorithm for image compression was based on Principal Component Analysis (PCA). We collected multispectral imager data over a period specified by the operator and sampled it to train a PCA model. The new imaging data was then projected pixel-wise onto the three most significant components and saved as.jpg or.png files for download. When tested on Landsat8 data, the first three PCA components captured 95 percent of the variability in the pixel data. We used the inverse transform to recreate the original data and obtained a correlation coefficient of.76 when correlating the original pixel values with the recreated ones.

We further tested our approach by trying to segment Landsat8 images according to land cover types. We implemented a pixel-wise classification ExtraBoostedTree algorithm using pixel values from all 10 visible and near-infrared (NIR) channels, achieving an accuracy of 91 percent. However, when using only the loadings from the first 3 PCA components, the accuracy decreased to 44 percent, with some classes having much lower accuracy numbers. This implies that even though the inverse PCA transform can recreate a seemingly good version of the original image, it is not suitable for downstream classification tasks. Consequently, we refined the approach using image masks from the Landsat8 dataset to obtain a new set of components derived from a linear discrimination task.

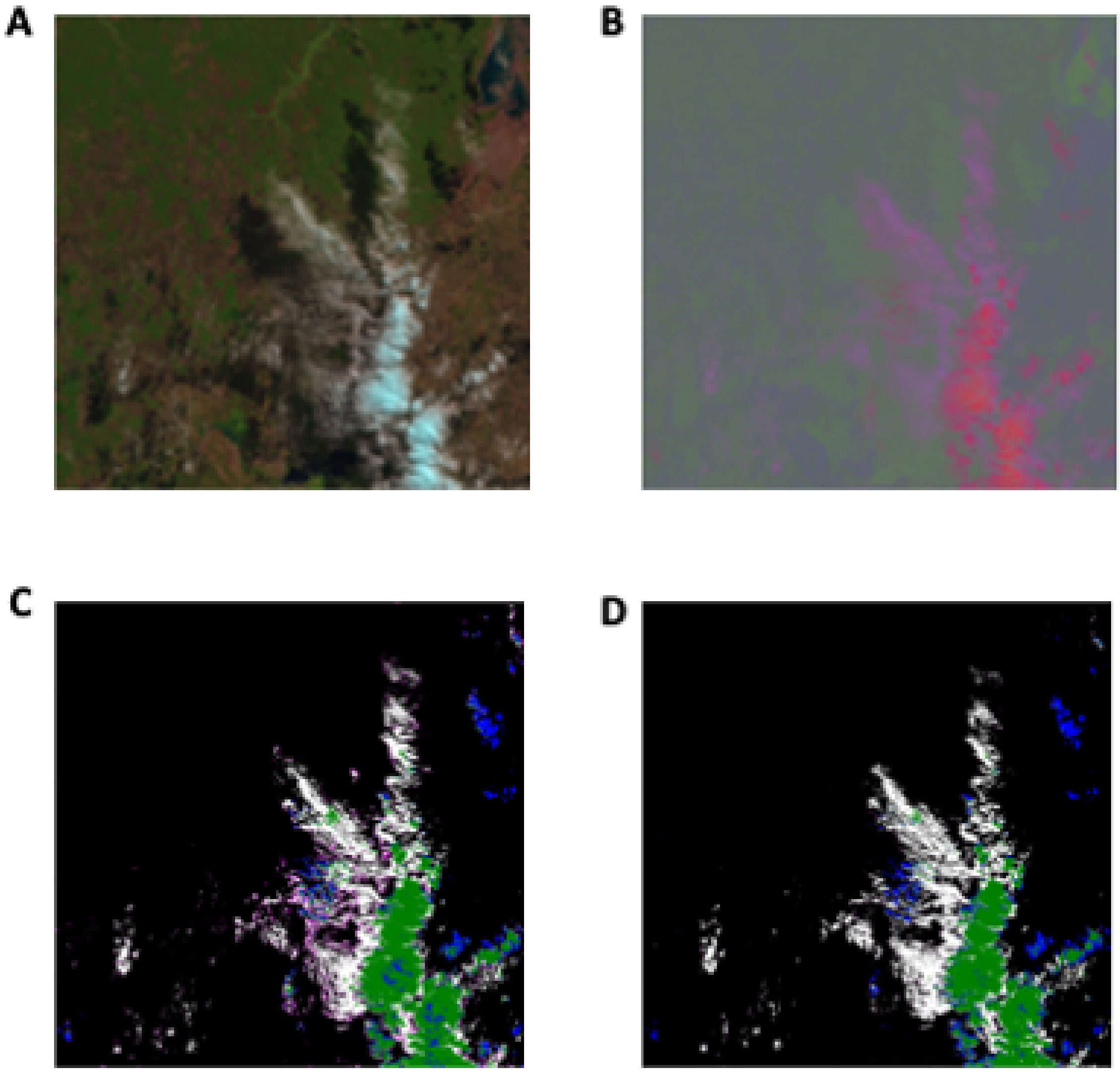

Figure 2. Illustration of the results from image compression and segmentation. A: Example of Landsat8 image, visible spectrum. B: Image projected on top 3 LDA components and displayed with false colors. C: Ground-truth image segmentation mask. D: Image segmentation produced by ExtraBoostedTree algorithm applied to LDA projected image

PCA obtains the projection components by attempting to maximize the variability preserved in lower-dimensional projections of the data, while LDA obtains the projection components by choosing them such that a linear classification task will obtain optimal performance. When applying the same ExtraBoostedTree classification algorithm on the first three LDA components, we achieved an accuracy of 84 percent. We thus concluded that compressing the data utilizing the LDA approach achieves a 70 percent data compression (only 3 LDA components instead of 10 original channels) at the expense of a 7 percent drop in accuracy in a classification task. An example of images segmentation results is presented in Figure 2.

4. Ground post-processing utilizing generative upsampling

Once the images have been downloaded to the ground, there is no longer a restriction on the amount of computational resources available, allowing for a wide range of algorithms to be applied for further processing and analy-

WACV #*****

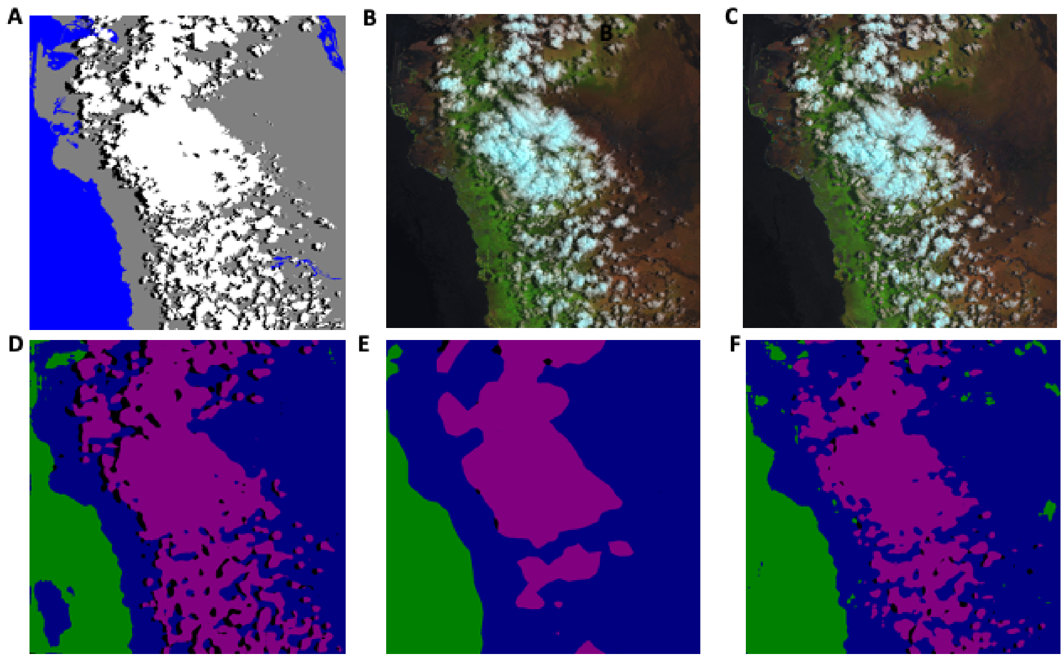

Figure 3. Illustration of images processed utilizing generative upsampling and subsequent segmentation. A: Mask of the ground truth image. B: Visible spectrum image downsampled from 1000 by 1000 to 256 by 256 size. C: The image was upsampled back to 1000 by 1000 size utilizing a generative AI algorithm. D: Segmentation results utilizing the original visual spectrum image. E: Segmentation results obtained from the 256 by 256 downsampled image. F: Segmentation results obtained from the upsampled 1000 by 1000 image.

sis. AWS offers a broad selection of pre-made generative techniques and APIs that make rapid prototyping and experimentation possible [1]. We used the SageMaker Python SDK for upscaling with a pretrained Stable Diffusion model [2]. Upscaling is the process of creating a high-resolution image from a low-resolution image and a textual prompt that describes the image. This algorithm can be used on images that are low-resolution, blurry, or pixelated, with the aim of transforming them into higher-resolution images that appear smoother, clearer, and more detailed. In our context, the goal is not only to achieve a smoother appearance of the resulting image but also to evaluate the performance on subsequent tasks such as image segmentation. For the image segmentation task, we use the Amazon JumpStart standard Semantic Segmentation algorithm [3] and the results are displayed in Figure 3. By visually inspecting and comparing panels E and F, it can be concluded that upscaling the image leads to better performance in the image segmentation task, which encourages further experimentation with generative models.

5. Conclusions and future directions

This paper presents the findings of the first implementation of Amazon Web Services (AWS) Machine Learning (ML) components on an orbiting satellite. The Greengrass framework allowed ML components to be flexible in the face of computational and transmission limitations, allowing ground operators to retrain, update and create new ML components and models for deployment.

We demonstrate the use of robust, low-footprint ML techniques, such as PCA and LDA for compressing large sets of imaging data and downloading them to the ground for further processing. In addition, we describe our initial experiments using generative techniques to enhance compressed imagery data.

AWS SageMaker JumpStart models facilitate rapid experimentation and prototyping until the most promising candidate algorithms are identified. However, more research is needed to quantify the performance of the generative algorithms presented here. In the context of geospatial data analysis, perhaps the generation of labeled data for training ML algorithms is an area of greatest need. Generative Artificial Intelligence (AI) coupled with physical simulations may be a productive direction for the rapid development and deployment of novel types of imaging sensors.

References

[1] How to Use Proprietary Foundation Models with Amazon SageMaker JumpStart for Building Generative AI Applications. 4

[2] Stability AI's New Upscaler Can Quadruple the Resolution of an Image | PetaPixel. 4

[3] Run image segmentation with Amazon SageMaker JumpStart | AWS Machine Learning Blog, Aug. 2022. Section: Amazon SageMaker. 4

[4] Indraneel Mitra and Ryan Burke. *Intelligent Workloads at the Edge: Deliver cyber-physical outcomes with data and machine learning using AWS IoT Greengrass*. Packt Publishing, Jan. 2022. 1