Data Pipeline for Hyperspectral Camera in LDFZ

Naman J. Parikh

July 26, 2024

Design

Hyperspectral Camera [1]

The primary data producer in this pipeline is the hyperspectral camera. The camera captures a line of pixels (horizontal line as it is set up in the LDFZ). In contrast to a standard camera which captures three values (red, green, and blue) for each pixel, the hyperspectral camera captures values at 371 wavelengths for each pixel. This allows us to determine temperature and emissivity of a blackbody imaged by the camera using blackbody physics. Data from captured images is stored by the Hyperspec III software running on a dedicated computer. Each image capture generates a folder that is named with the date and time of capture and contains the image files.

Pipeline Overview

The goal of this pipeline is to download and analyze data on a seperate server for storage and efficiency while making the result of analysis available to the LDFZ user. To do this, we build two separate Kafka topics; one for raw data and one for the analysis result. We use OpenMSIStream [2] for streaming the files. The topics can be found on the brokers associated with `paradim01` as `hyperspec_LDFZ_data` for raw data and `hyperspec_LDFZ_result` for results.

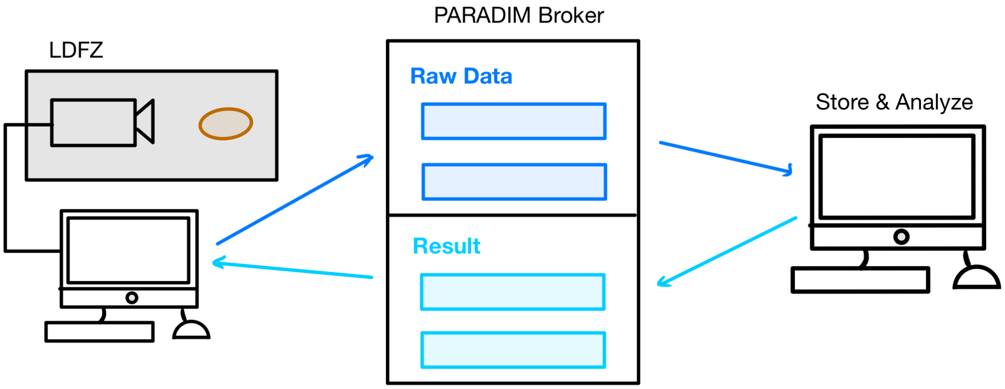

Figure 1: High-level design of the data pipeline

Camera: Data Producer and Result Processor

The camera's dedicated computer continuously runs a Python script, which is saved at `"C:\Users\Headwall\streaming.py"` and corresponds to `"streaming_scripts/hyperspec.py"` on my GitHub repository. This script prepares and spawns two threads.

1) The first runs a DataFileUploadDirectory producer watching `"C:\Headwall\sensor1\captured"`. This uploads image folders to `hyperspec_LDFZ_data` as they appear in the captured directory.

2) The second runs a DataFileStreamProcessor consuming files from `hyperspec_LDFZ_result`. The result files are assumed to be temperature arrays stored in NumPy (`".npy"`) files. The stream processor saves a copy of the file as well as a heatmap of the array, which represents a thermal gradient of the image.

Data Processor

The data processor script can be found at `"streaming_scripts/processor.py"` in my GitHub repository. This script runs a stream processor that tracks which files for each image have been received and triggers a pyrometry analysis when all requisite files are received. The pyrometry analysis can be foundon my GitHub repository in the analysis function in `"streaming_scripts/temperature_analysis.py"`. To run this script on another machine, the directories for the broker config file and stream processor directory will need to be adjusted accordingly.

Additional Consumers

Girder consumer?

References

[1] Headwall Photonics. https://headwallphotonics.com/products/machine-vision/micro-hyperspec-vnir-e-series/.

[2] Margaret Eminizer, Sam Tabrisky, Amir Sharifzadeh, Christopher DiMarco, Jacob M. Diamond, K.t. Ramesh, Todd C. Hufnagel, Tyrel M. McQueen, and David Elbert. Openmsistream: A python package for facilitating integration of streaming data in diverse laboratory environments. *Journal of Open Source Software*, 8(83):4896, 2023.