***HyperGix***

A User-Friendly, Open-Source Hyperspectral Imagery Desktop Application Using Python and PyQt5

Master's Project

Department of Computer Science

Montclair State University

John Gallino

Fall 2021

Advisor: Dr. Stefan Robila

# Abstract

Contents

[Abstract 2](#_Toc89434069)

[List of Figures 5](#_Toc89434070)

[1. Introduction 6](#_Toc89434071)

[1.1. Overview of Spectral and Hyperspectral Imaging 6](#_Toc89434072)

[1.2. Obstacles in Hyperspectral Imaging 8](#_Toc89434073)

[1.3. Hyperspectral Data Processing Software 11](#_Toc89434074)

[1.4. Current State of Hyperspectral Satellite Imaging 12](#_Toc89434075)

[2. HyperGix Overview 14](#_Toc89434076)

[3. HyperGix System Requirements 15](#_Toc89434077)

[4. HyperGix Design and Structure 16](#_Toc89434078)

[4.1 Module System 16](#_Toc89434079)

[4.1.1. Image Viewer 16](#_Toc89434080)

[4.1.2. Spectra Manager 21](#_Toc89434081)

[4.1.3. USGS Search 23](#_Toc89434082)

[4.2 SQLite Database 25](#_Toc89434083)

[4.2.1 Table Schemas 25](#_Toc89434084)

[5. Code Structure Details 27](#_Toc89434085)

[5.1 Subclassed Classes within qViewer 27](#_Toc89434086)

[5.2 Use of PyQt Signals and Slots 28](#_Toc89434087)

[5.3 Worker Classes 29](#_Toc89434088)

[5.4 Creating Custom Material Classes 30](#_Toc89434089)

[5.5 Pixel Classification 31](#_Toc89434090)

[6. Conclusions and Future Work 33](#_Toc89434091)

[Appendix I – Installing HyperGix 39](#_Toc89434092)

[Appendix II – Installing GDAL 39](#_Toc89434093)

[Appendix III – HyperGix Source Code 39](#_Toc89434094)

# List of Figures

[Figure 1 Comparison of Hyperion sensor range to ALI sensor 6](#_Toc89251484)

[Figure 2 Example spectral signatures of three minerals 7](#_Toc89251485)

[Figure 3 GERBIL software interface 10](https://d.docs.live.net/f9ab3503441286fc/Documents/HyperGix%20Final%20Report.docx#_Toc89251486)

[Figure 4 Image Viewer Module 15](#_Toc89251487)

[Figure 5 headerFix() function code snippet 16](#_Toc89251488)

[Figure 6 Comparison of RGB, Single Band, and NDVI viewing modes 18](https://d.docs.live.net/f9ab3503441286fc/Documents/HyperGix%20Final%20Report.docx#_Toc89251489)

[Figure 7 Spectra Manager Module 20](#_Toc89251490)

[Figure 8 USGS Search Module 22](#_Toc89251491)

[Figure 9 Graphical depiction of signals and slots [29] 27](https://d.docs.live.net/f9ab3503441286fc/Documents/HyperGix%20Final%20Report.docx#_Toc89251492)

[Figure 10 Selection of 15 representative bands for pixel classification 30](#_Toc89251493)

# Introduction

The objective of this project was to develop a user-friendly software application to read and analyze hyperspectral images. Having some experience in making desktop applications for non-tech users, I aimed to create an easy-to-use, intuitive tool that would require minimum instruction to be used by someone unfamiliar with hyperspectral images. At the same time, it was important to design the software in such a way that it can be easily expanded with more capabilities by other developers in the future. The software, titled ***HyperGix***, was first designed in early development using *Python* and the *tkinter* GUI framework but was then rebuilt using the more advanced *PyQt GUI* toolkit. It allows users to browse the online library of Hyperion satellite sensor images catalogued by the United States Geological Survey (USGS), download images of their choice, import their own supported files, examine pixel spectra, create a custom spectral library, and perform pixel classification tasks. The project is open source.

## Overview of Spectral and Hyperspectral Imaging

Spectral imaging is the art and science of capturing image-based information across the electromagnetic (EM) spectrum. This spectrum includes visible light, but also infrared, ultraviolet, microwaves, x-rays, gamma rays and radio waves. Wave type is determined simply by its frequency and wavelength. Radio waves are the largest and have the lowest frequency while gamma rays are at the opposite end of the spectrum with wavelengths as small as the nucleus of an atom [1]. All physical objects reflect these waves in different amounts. Plants appear green not because they absorb green light, but because they reject it and reflect it away back towards our eyes. However, human eyes can only see within the visible light spectrum and cannot measure the light reflected in other frequencies without special tools, and while they are precise enough to distinguish among various nuances of the same color (i.e. changes in wavelength), they still have very coarse wavelength resolution compared to current state of the art digital imagers. Spectral imaging was developed to capture and display that type of information, using specialized film or digital sensors sensitive to specific frequencies of interest.

Hyperspectral imaging (HSI) is a more recent advancement within this field. Rather than capturing light in a few choice frequency ranges, hyperspectral sensors aim to capture reflectance values across a wide and continuous expanse of the EM spectrum – usually beginning with ultraviolet and visible light and expanding to the near infrared and shortwave infrared frequency ranges.

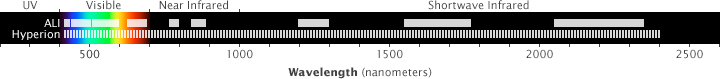


Figure Comparison of the continuous Hyperion sensor hyperspectral frequency range and the Advanced Land Imager (ALI) sensor's more limited multispectral range. Human visible light range is also represented. [2]

Different materials have a unique reflectance pattern, like a fingerprint. A material like silica reflects energy across the EM spectrum in a very different way than water vapor, or soil, or iron. Thus, by utilizing hyperspectral imaging sensors, scientists can log these patterns – referred to as a *spectral signature* – from known material samples, store them in a database known as a *spectral library*, and then refer to that database in the future to identify unknown materials in the field [3]. Figure 2 displays the spectral signature of three different minerals.

Chart, line chart

Description automatically generated

Figure Example spectral signatures of three minerals [2]

This method of data capture can be used not only in a lab setting, but in aerial capture from planes or satellites. The Hyperion sensor, mounted aboard the Earth Observing-1 satellite, captured hyperspectral images of the Earth’s surface from space from 2000 to 2017 [4]. Figure 1 displays the difference in spectral sensitivity between the NASA’s Hyperion sensor and Advanced Land Imager sensor, both present on the Earth Observing-1 satellite [2]. The upper row in the figure shows the 9 separate, coarse bands of the ALI sensor, while below that are displayed the 220 narrow, contiguous bands of the Hyperion sensor. Data collected from this sensor is possibly the largest open source of hyperspectral imagery available to the public as of this writing, with over 80,000 images available for download from the United States Geological Survey’s (USGS) EarthExplorer online tool [5]. Aerial capture of HSI is particularly useful for mining, agriculture, and many fields of earth science [3]. HSI technology can be used to measure the health of crops, rainforests, or coral reefs, or to see through the smoke of a forest fire [6].

## Obstacles in Hyperspectral Imaging

Because hyperspectral imaging is a relatively young technology, it is still hindered by several issues that present challenges to data scientists. One issue facing researchers is the size and quantity of the data captured. Compared to typical digital images which contain three values per pixel (red, green, and blue), hyperspectral images often contain hundreds of values per pixel. Scans from the Hyperion sensor, which records roughly 220 bands, contain 242 values for every pixel in the image (including bands with unusable data) [4] [7]. This results in a single hyperspectral image containing 200 to 400 megabytes of data. Working with a satellite that captures images 24 hours a day, this quickly adds up to many terabytes of data that must be stored on site or transmitted back to Earth. For data scientists, this immense amount of information is sometimes more than necessary to perform material classification or other analytical tasks, so they rely on dimensionality reduction techniques that isolate the most relevant information.

The outstanding spectral resolution of hyperspectral images means spatial resolution is often sacrificed. *Spatial resolution* is defined as the real-world area covered by a single pixel in an image [9]. In factory settings, hyperspectral image sensors used along assembly lines to assess product quality are often recording images of only a few hundred pixels [8]. The spectral information of these few pixels is usually enough to identify bruises on fruit or sugar content in potatoes. Satellite hyperspectral imagery does not have nearly the spatial resolution one might find on Google Maps Satellite View. White lines on roads are clearly visible from Google’s satellites in most areas, indicating a spatial resolution of only a few inches. In contrast, a single pixel in a Hyperion scan represents an area of 30 square meters [4]. Thus, when speaking about satellite hyperspectral imagery, it is often impossible to distinguish any structure smaller than a large building or highway as of this writing. This limitation can be addressed to an extent by bringing the sensor closer to the subject (i.e., flying the sensor lower to the ground or using a plane or unmanned drone instead of satellites).

Once one has a hyperspectral scan of their research area, what is next? A farmer may be interested to know if one area of their soy field has higher iron content than another. A biologist may be scanning a forest for the presence of a particular damaging fungus. A city planner may be interested in measuring the growing number of buildings in a particular area over the past decade. Central to all these tasks is the classification of pixel data. Considering HSI’s poor spatial resolution, a single pixel in an image usually does not contain a uniform material, but a blend of several different materials. Determining the makeup of this combination is known as ‘spectral unmixing’ and is yet another challenge data scientists must reckon with [10].

For much of the past few decades, the lack of data standards also posed a challenge to researchers. As of this writing, there is no agreed upon standard for hyperspectral image data. This means HSI sensor manufacturers are recording data in whatever matter they see fit. While some sensors record every band of a single pixel before moving to the next pixel, others may record every pixel of one band before proceeding to the following band. [11] These differences in methodologies create headaches for anyone developing software or algorithms to manipulate this data. The IEEE Standards Association, the group that establishes such rulings, has been working on this problem since 2018 [12] , but have yet to complete their work as of December 2021.

Regarding ­­data analysis, the large number of variables when capturing data out in the field make data processing especially challenging. Although materials do have their own unique reflectance pattern, this pattern can be influenced by a variety of factors in the field, including time of day, time of year, angle of sunlight, intensity of sunlight, cloud cover, atmospheric haze, angle of the surface, and differences in light and shadow within the image. Furthermore, the HSI sensor in use may be more sensitive to some areas of the EM spectrum than others, further distorting captured reflectance values. In my own project, I took certain steps to minimize this issue, which will be discussed shortly.

## Hyperspectral Data Processing Software

Chart

Description automatically generatedToday there are many software packages that can read and analyze hyperspectral data. Many of these packages are provided by the same companies that manufacture HSI cameras. Examples include *Resonon* with its *Spectronon* software [13] and *Prediktera* with its own software suite [14]. This software not only controls the HSI cameras remotely, but usually include several processing and data analysis tools to make sense of collected data. Common features are Principal Component Analysis (PCA), clustering algorithms for classification and quantification, and the implementation of machine learning. *ENVI*, produced by L3 Harris Geospatial, claims to be the industry standard for image processing and analysis among scientists and GIS professionals [15]. GIS, or Geographic Information Systems, is a field of study that often goes hand in hand with aerial and satellite image analysis. *QGIS*, a popular open-source GIS software package, includes installation of *GDAL* – an open-source library of tools to read and manipulate geospatial imagery that has been in development since 1998 and upon which many newer libraries rely on [16] [17]. *MultiSpec* is another open-source program developed by Purdue University which reads hyperspectral images and provides several statistical tools to cluster and classify the pixel data. [18] One final open-source software of note is *GERBIL* [19]. Produced by German researchers, *GERBIL* tackles the challenging task of visually representing hyperspectral information to the user and includes several interesting visualization features, as shown in Figure 3.

Figure GERBIL software includes several innovative data visualization techniques

## Current State of Hyperspectral Satellite Imaging

In September 2021, NASA launched Landsat 9 – the latest generation of imaging satellites in the Landsat program which first started in 1972 [20]. This satellite, like its predecessors, will contribute thousands of images of the Earth’s surface to the US Geological Survey which are vital to tracking the distribution of species and ecological changes occurring over its many years in service. However, Landsat 9 does not include a hyperspectral sensor. In fact, it records in only 11 bands, which is sufficient for its purposes of detecting vegetation, cloud cover, and thermal infrared radiation. This demonstrates how specialized the field of hyperspectral imaging can be. NASA did maintain a hyperspectral sensor (Hyperion) in Earth’s orbit aboard The Earth Observing-1 satellite from 2000 to 2017, but it has yet to reintroduce a new hyperspectral sensor to take its place.

Other nations are taking the lead, however, with the German Environmental Mapping and Analysis Program (EnMAP) in development and set to launch some time in 2022 [21]. The Italian Space Agency (ASI) launched its own hyperspectral imaging satellite, PRSIMA, in March of 2019, with the intention of eventually sharing its data with the public [22]. HySIS is a hyperspectral imaging satellite currently in orbit that was launched by the Indian Space Research Organization (ISRO) in November 2018 [23]. China has several hyperspectral imaging satellites in orbit currently. OHS-01, 02, 03, and 04 were launched together in September 2019 [24]. All four satellites can capture data in 256 bands. These joined Gaofen 5 (GF 5) which was launched in 2018 as part of the China High-definition Earth Observation System (CHEOS). A second Gaofen 5 (GF 5-02) was launched in September 2021 [25] .

These satellite missions are public knowledge, but it is safe to assume there are other hyperspectral sensors currently in orbit in use by various military forces around the world. Hyperspectral technology can prove useful in detecting “camouflaged or hidden military targets, such as missile launch sites or testing facilities for nuclear weapons.” [26] The American TacSat-3 satellitein service from 2009 to 2012, provided real-time hyperspectral data in over 400 spectral bands to US forces on the ground. The China Commercial Remote-sensing Satellite System (CCRSS), of which little information is known publicly, was scheduled to launch in 2016 and can allegedly outperform the TacSat-3 [27].

# 2. HyperGix Overview

HyperGix is an open-source, user-friendly desktop application to experiment with and learn about hyperspectral imagery. It is written in Python 3 and uses the PyQt GUI framework for its interface. The backend of HyperGix is a SQLite database included with the application that stores data about the user’s hyperspectral files, as well as spectral information to be discussed in detail later in this report. The software is still in alpha and has not yet gone through thorough bug testing.

HyperGix aims to be intuitive and easy to use, with little instruction necessary. Presently it consists of three modules: *Image Viewer*, *Spectra Manager*, and *USGS Search*. Using HyperGix, the user can organize and view hyperspectral files downloaded to their system, in either RGB (red, green, blue), single-band or Normalized Difference Vegetation Index (NDVI) views. The user may also perform Principal Component Analysis (PCA) on an image, define custom material classes and assign individual pixels from hyperspectral images to those classes to create a training set for classification tasks.

If the user does not have access to hyperspectral files, or is inexperienced in where to find them, the USGS Search interface allows them to simply search for a geolocation and browse available hyperspectral scans to download from the United States Geological Society’s database of Hyperion satellite scans captured from the years 2000 to 2017. However, the USGS interface is cumbersome and difficult to navigate. By using HyperGix, and its user-friendly Google Maps-based interface, the data access is significantly improved.

# 3. HyperGix System Requirements

HyperGix was written and designed for Windows 10 desktop machines. Though both Python 3 and the PyQt library are supported on Mac, Linux, and older versions of Windows, no testing has been performed with HyperGix on those systems and support cannot be guaranteed.

HyperGix requires the installation of the Geospatial Data Abstraction Library (GDAL) to operate. GDAL is a free, open-source translator library and command-line tool to read and manipulate geospatial data files such as those used by HyperGix. Installation of GDAL can be complex, and it is not included with HyperGix. A guide to installing it on Windows 10 can be found in the Appendix. Information about GDAL can be found at [**www.gdal.org**](http://www.gdal.org)**.**

HyperGix includes an installation of Python 3.6, as well as several Python libraries listed below. These libraries are already packaged with the software so no action should be required from the user. However, in the event of an issue, any of the libraries below can be easily acquired from the Python Package Library (pypi.org) and installed with PIP.

* brokenaxes 0.5.0 https://github.com/bendichter/brokenaxes
* gdal 3.0.2 https://pypi.org/project/GDAL/
* geocoder 1.38.1 https://github.com/DenisCarriere/geocoder
* matplotlib 3.4.3 https://matplotlib.org/
* numpy 1.21.2 https://numpy.org/
* pyqt 5.9.2 https://anaconda.org/conda-forge/pyqt
* requests 2.26.0 https://docs.python-requests.org/
* spectral python 0.22.2 https://www.spectralpython.net/
* sqlite 3.36.0 https://anaconda.org/conda-forge/sqlite

The HyperGix USGS Search feature also requires login credentials to the USGS EROS Registration System. Users may register an account at **ers.cr.usgs.gov/register**

For demonstration purposes, login credentials are already included with the HyperGix download. However, if the user intends to use HyperGix extensively, it is encouraged they register their own account and input the credentials when prompted.

# 4. HyperGix Design and Structure

## 4.1 Module System

HyperGix version 1.0 consists of three modules separated into tabs at the top of the program interface. These modules are also separated in the code as distinct classes – qViewer, qProfileManager, and qHypbrowser – which are defined in *qviewer.py*, *qmanager.py*, and *qhypbrowser.py* respectively. The main application is defined in *main\_pyqt.py*. This file includes code for the HyperGix application window and tab structure (MyNotebook class) as well as the many signals and slots used by PyQt for the different modules of the app to correspond with each other. Developers are encouraged to continue this system of separating modules into distinct classes and adding a tab to the main interface that instantiates that class. It is required that all classes that intend to use the signal and slots system inherit from the PyQt QWidget or QObject classes.

### 4.1.1. Image Viewer

Most of the time in HyperGix will be spent in the *Image Viewer* module (see Fig. 4). This module is dedicated to viewing and managing the user’s hyperspectral image library and the files therein. Upon opening HyperGix, any suitable Hyperion images contained in the **downloads** folder in the HyperGix directory will be automatically added to the Scan Library list on the left side. Hyperion images must have the file extension **.L1R** to be recognized and added to the image library automatically.

Graphical user interface, application

Description automatically generated

Figure Image Viewer Module

Early in development, an error inherent to Hyperion image header files was discovered in which the ‘header offset’ value is mistakenly recorded as zero when it should be 2502, causing the images to display incorrectly. HyperGix automatically handles and corrects this error, so the images display accurately in the software, as displayed in Figure 5.

    def headerFix(self):

        old\_header = os.path.join(DOWNLOAD\_PATH, self.filename[:-11], f'{self.filename[:-11]}.bak')

        header = os.path.join(DOWNLOAD\_PATH, self.filename[:-11],

f'{self.filename[:-11]}.hdr')

        # Making a new copy of header with adjusted offset of 2502

        os.rename(header, old\_header)

        shutil.copyfile(old\_header, header)

        file = open(header, 'r')

        replacement = ""

        for line in file:

            line = line.strip()

            changes = line.replace("header offset = 0", "header offset = 2502")

            replacement = replacement + changes + "\n"

        file.close()

        fout = open(header, 'w')

        fout.write(replacement)

        fout.close()

Figure headerFix() function corrects faulty Hyperion image offset for accurate display by HyperGix

HyperGix also allows for importing supported hyperspectral images from external directories by use of the **Import Spectral Scan** button below the *Scan Library* list. The location of the imported file on the user’s computer is stored in the SQLite database so moving the file to the ‘downloads’ folder is not necessary. At present, supported filetypes are limited to **.lan, .hdr, .L1R, .tif, and .hd5**, though importing other filetypes can be attempted. HyperGix will attempt to use the GDAL library to convert an unsupported file into something it can read, but success is not guaranteed due to the lack of current standards in hyperspectral data storage.

The **Remove From Library** button in the Image Viewer module will remove information about a spectral image from the internal database, but it will not delete the file from the user’s system. A future update may include a pop-up window to present this option.

Image files can be viewed in the application by double-clicking their name in the *Scan Library* list. Information about the file is displayed in the File Properties panel in the lower left side of the Image Viewer module. This information is included in the image file itself and can vary by file format. For Hyperion scans, it includes the unique Scene ID assigned by the USGS, along with capture date and time, dimensions of the image, interleave order and other metadata. Hyperion images all consist of 242 bands and 256 columns (or samples) but the number of rows may vary.

The center area of the Image Viewer module allows the user to view an individual hyperspectral scan, pan and zoom on it, and save still frames of an area to their computer. It also allows switching between three viewing modes:

Chart, timeline, bar chart, waterfall chart

Description automatically generated **RGB Mode** – a full color representation of the captured area. By default, bands 32, 21, and 13 are used for red, green, and blue values respectively, as they correspond to wavelengths of 671 nm, 559 nm, and 518 nm on the Hyperion sensor, but the user may select any bands for these values.

Figure Comparison of the same Hyperion scan in RGB, Single Band, and NDVI viewing modes

**Single Band Mode** – view a single band of the hyperspectral image. Some bands may include noisy data or no data at all. When hovering the cursor over the image, single-band values for the pixel currently under the cursor are shown in the upper right corner of the central panel.

**NDVI Mode** – Normalized Difference Vegetation Index is a graphical indicator of live green vegetation based on the nature of vegetation to reflect infrared light while absorbing red light [28]. It provides an approximation of vegetation health in a region. HyperGix uses green to indicate areas of lush vegetation, red to indicate inorganic materials or dead vegetation, while black usually indicates bodies of water.

By double clicking a spot within the image, the entire spectral profile of the single pixel clicked is presented on the right side of the *Image Viewer* module as a line graph. For Hyperion scans, a break is included on the X-axis to indicate the omission of bands 58 to 76 which are known to contain no data [7]. Near the bottom of the chart are bars that represent the areas of blue, green, red and near-infrared light among the spectral bands.

Below the Pixel Spectra chart are three buttons. **Add Pixel to Profile** will store the most recently viewed pixel spectra in the database, along with an assigned material class selected by the user. **Calculate PCA** will perform principal component analysis. This process may take 2-3 minutes depending on the size of the file and may appear to freeze up the application. Please be patient while the extensive calculations are performed. This process will produce a covariance matrix as well as the number of principal components necessary to represent 99% of the variance of the image data. In most cases, this is less than 5. The individual principal components can be viewed in the central panel by adjusting the band slider at the bottom of the panel.

### 4.1.2. Spectra Manager

While individual pixels can be assigned to a class from any imported hyperspectral image in the Image Viewer tab, the Spectra Manager module is where the user may create, delete, and view these material classes (also called Profiles). HyperGix includes several material profiles by default including **Water**, **Healthy Vegetation**, **Sand**, **Man-Made Structure** and others. These are simply for convenience and may be deleted or replaced by the user’s own custom material profiles.

To create a custom material profile, the user can click **Create New Profile**, and give the class a name. Upon completion, the profile will be added to the list, and the SQLite database on the backend.

A picture containing graphical user interface

Description automatically generated

Figure Spectra Manager Module

Double-clicking a profile name from the Saved Profiles list will chart all the pixel spectra assigned to that profile in the main area of the module. If no pixels have been assigned to the material, no chart will be visible. A legend is provided on the chart to identify each pixel by color for the detection of outliers and abnormalities. A thicker, navy-blue line on the chart represents the *average spectra* of all the pixels collected for that profile class. This average spectra value is stored in the SQLite database and can be used to identify the material during classification tasks.

Below the pixel spectra chart is the **Pixel Viewer** window. This allows the user to trace back a stored pixel to its source image. Upon selecting a pixel number on the left, the source image is loaded up and displayed along with the row and column that the pixel represents. The **Delete Pixel** button will delete the selected pixel from the profile.

### 4.1.3. USGS Search

The USGS Search module requires an internet connection and communicates with the USGS Earth Resources Observation and Science (EROS) Center using their *Machine-to-Machine API* to search for Hyperion sensor image scans from the Earth-Observing 1 (EO-1) satellite.

Graphical user interface

Description automatically generated

Figure USGS Search Module

Searches can be easily performed by entering a place name or a set of latitude and longitude coordinates. Using Google’s Geocoding API, place names are converted to coordinates in an instant. Searches can be performed by clicking the **Search** button or simply pressing Enter while the search box is in focus. By default, 3 image results are displayed at a time and only images with a Cloud Cover value of less than 60 are returned, though these parameters can be adjusted by the user.

Because each result requires two separate API calls – one to retrieve the preview JPEG image and another to retrieve the download link – searches may be slower to execute than desired. A future update may utilize multithreading to speed up this process.

The server response for each search is displayed in a panel on the left side. This may be useful if the user needs more specific information about the results, including coordinates for the four corners of the image. This panel may be closed by clicking and dragging the area separating it from the image results.

Any file returned by the search may be downloaded by clicking the **Download HSI File** button below each result. The user will be warned that these are large files and will have to confirm the download. Downloading may take several minutes and is indicated by a progress bar animation in the lower right corner of the HyperGix application window.

The user will immediately be prompted to enter a nickname for the file being downloaded. This is purely for convenience and is optional. This nickname will be stored and displayed in parentheses alongside the image name in the Scan Library list in the Image Viewer module.

For the user’s convenience, a Google Maps view is shown initially in the USGS Search panel before a search is performed. The user may easily copy latitude and longitude coordinates by right clicking an area on the map, then paste them in the search bar of the panel to search the USGS online database for scans of that location.

## 4.2 SQLite Database

HyperGix stores information about the user’s hyperspectral image library, as well as custom classes and pixel spectral information, in a SQLite database file titled **data.db**. This database consists of three primary tables described below, plus a fourth automatically generated for the auto-increment function.

**scans -** Contains the unique id, nickname, and file path information of all hyperspectral images imported into the HyperGix software, along with several metadata values about the image that may be referenced in future software updates.

**materials** – Stores user-created material classes used in the Spectra Manager module

**pixels** – Stores pixel spectra values selected by the user along with their assigned material class and source information referenced by the Pixel Viewer panel in the Spectra Manager module.

### 4.2.1 Table Schemas

Described below are the schemas used for the tables in HyperGix’s database. The schemas were designed with future expansion in mind and include some data points that are not yet used in the software.

CREATE TABLE "pixels" (

"pid" INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,

"source" INTEGER NOT NULL,

"row" INTEGER NOT NULL,

"col" INTEGER NOT NULL,

"material" INTEGER NOT NULL,

"spectra" TEXT,

FOREIGN KEY("source") REFERENCES "scans"("id"),

FOREIGN KEY("material") REFERENCES "materials"("mid")

);

CREATE TABLE "materials" (

"mid" INTEGER PRIMARY KEY AUTOINCREMENT UNIQUE,

"name" TEXT NOT NULL UNIQUE,

"spectra" TEXT

);

CREATE TABLE "scans" (

"id" TEXT,

"nickname" TEXT,

"filepath" TEXT NOT NULL UNIQUE,

"rows" INTEGER NOT NULL,

"samples" INTEGER NOT NULL,

"bands" INTEGER NOT NULL,

"sensor" TEXT,

"interleave" TEXT,

"date" TEXT,

"time" TEXT,

"ul\_lat" REAL,

"ul\_lon" REAL,

"ur\_lat" REAL,

"ur\_lon" REAL,

"ll\_lat" REAL,

"ll\_lon" REAL,

"lr\_lat" REAL,

"lr\_lon" REAL,

"c\_lat" REAL,

"c\_lon" REAL,

PRIMARY KEY("id")

);

# 5. Code Structure Details

## 5.1 Subclassed Classes within qViewer

Most of the work of reading and displaying hyperspectral data within HyperGix was done by the creators of the *Spectral Python* (SPy) library, however some classes and methods had to be modified to be compatible with a *PyQt* desktop application.

***MyImageView*** is a subclass of Spectral Python’s **ImageView** class used to display hyperspectral scans on screen. It includes a new method to record the last pixel that has been viewed by the user, in the case the user decides to assign it to a material class and store it in the database. It also includes modifications to the **open\_zoom()** method that allow for HyperGix’s proprietary Pixel Viewer feature to trace back the source of a stored pixel and show a thumbnail preview of where it was taken from. The keyword of **brokenaxes**, which defaults to True, determines if the line graph should include a break in the x axis for the gap of bad bands known to be common to Hyperion images. **brokenaxes** is set to False when displaying hyperspectral images from other sensors.

**MyMouseHandler** subclasses Spectral Python’s **ImageViewMouseHandler** class and modifies the code to allow for the line graph of pixel spectra to appear in the same main HyperGix window alongside the image, as well as include the red, green, and blue color bars overlayed on the graph to indicate areas of the visible light spectrum.

**NavigationToolbar** is a subclass of **NavigationToolbar2QT** and simply removes some of the unnecessary toolbar buttons included by default.

## 5.2 Use of PyQt Signals and Slots

Diagram

Description automatically generatedSignals and slots are a unique feature of the Qt framework that allow objects within an application to communicate [29]. This replaces the typical method of callbacks used by other GUI toolkits. How it works is simple: an object may have one or more signals that can be emitted, like a beacon, when necessary. This signal can be linked up to a “slot” which is a member function of another object. Thus, for example, clicking a “Delete from List” button may emit a signal which triggers a method in the separate list object to remove the selected entry from the list. PyQt widgets include many default signals, but custom ones can be created as well. A signal can also carry a piece of data in several supported data types.

Figure Graphical depiction of signals and slots [29]

HyperGix uses signals and slots extensively to communicate between its different modules, and with its backend worker classes. Thus a developer may notice methods such as ***report\_scans*()** or ***requestPixels()*** in the HyperGix source codewhich rather than performing an action, simply broadcast a piece of data or request information from another object in the program. Signals and slots must be connected explicitly using a line of code, and most of these connections can be found in the ***main\_pyqt.py***file. The signals themselves are usually defined at the top of each class before any methods are defined. As mentioned previously, any class looking to use signals and slots must subclass PyQt’s **QWidget** or **QObject** classes.

## 5.3 Worker Classes

In addition to the primary GUI classes of **qViewer, qHypBrowser,** and **qManager,** HyperGix utilizes three “worker” classes that perform some of the intensive tasks behind the scenes. These classes and their methods are defined in the ***workers.py*** file.

**Databaser Class** - This class is responsible for interfacing with the SQLite database, including reading and writing information to it and reporting information to the GUI objects. In HyperGix, the object **databoy** is an instance of this class.

**Downloader Class** - This class handles the downloading and unzipping of Hyperion files from the USGS Search module, as well as the fix to the header offset value mentioned earlier in the report. An instance of this class (called **downloader**) is created and moved to a separate thread whenever the user initiates a file download within HyperGix. This prevents the main program from locking up while the file is downloaded.

**LogIner Class** - This class corresponds with the USGS API server to log in with appropriate credentials, receive and store an API key, and perform API queries. In HyperGix, the object **server** is an instance of this class.

## 5.4 Creating Custom Material Classes

Spectral libraries are databases of spectral signatures for materials such as minerals, soils, rocks, liquids, plants, or microorganisms that can be used as a reference for identifying unknown samples [30]. Several spectral libraries are available to the public, including the USGS Spectral Library and the ECOSTRESS Spectral Library. Although HyperGix does not yet interface with existing libraries, it allows the user to create their own small spectral library by isolating individual pixels and assigning them to a material class. Because different sensors may vary in sensitivity across the EM spectrum, it is best to use the same sensor for labeled and unlabeled data. Thus, HyperGix takes a semi-supervised approach to pixel classification. By taking 5-10 sample pixels of a known material, an average spectral signature can be determined for that material and stored in the database.

To create a new material class, first navigate to the Spectra Manager tab and click **Create New Profile** on the lower left side. A pop-up window will appear asking the user to name the new material class. Now back in the Image Viewer module, after double-clicking a pixel in an image and clicking the **Add Pixel to Profile** button, the newly created class should appear in the drop-down list of stored profiles.

To inspect a class, simply double click its name in the list of saved profiles in the Spectra Manager module. Every stored pixel will be charted as a different color, with a legend on the right side of the graph. The thick navy-blue line represents the average value of all the stored pixels for that class and is the spectral signature that will represent that class in pixel classification tasks. The Pixel Viewer panel below the graph allows the user to examine each pixel’s source to verify that it is classified correctly. Deleting a pixel can be done easily by the click of a button if a mistake has been made.

## 5.5 Pixel Classification

Classifying pixels by material is a common goal of hyperspectral imagery work. HyperGix currently utilizes a spectral angle mapper (SAM) algorithm to match image pixels to the reference spectra stored in the Spectra Manager. To classify a pixel, the SAM algorithm finds the angle in n-dimensional space between the pixel’s spectral signature and that of each class stored in the Spectra Manager. Figure 10 shows the conceptual visualization of the spectral angle between two 3-dimensional vectors. Given two vectors **x** and **y**, the angle α between them can be computed as:

(1)

where < , > refers to the dot product and || || refers to the vector norm

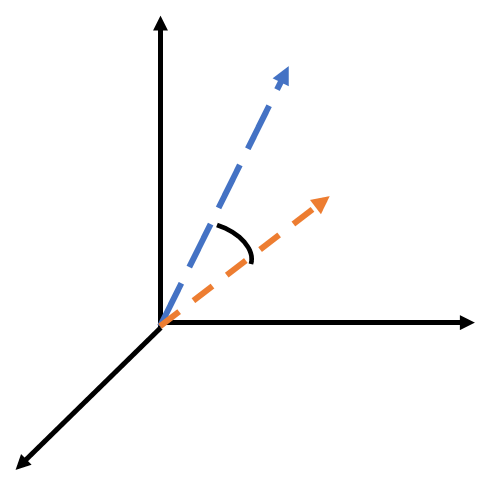


Figure 10 Visualization of vector angle as used in Spectral Angle Mapper

The smaller the angle, the higher the probability that the materials associated to these spectra are similar. This method corrects for any difference in illumination and albedo effects in the scene, but at the cost of much more processing time than other algorithms [31]. Chart

Description automatically generated

Figure A full spectral signature. Purple dots indicate the 15 select bands used to represent the signature in pixel classification.

Calculating spectral angle requires additional computation and processing time increases with higher numbers of spectral bands or with more reference classes. To alleviate this issue, HyperGix uses only 15 bands in its spectral angle calculations. These 15 bands are spread across the frequency range that most hyperspectral sensors are sensitive to - from roughly 400 to 2500 nanometers. Despite this major reduction in the number of calculations performed, spectral angle pixel classification for a Hyperion image can take several minutes.

# Conclusions and Future Work

To DO list

- Right-click to rename nickname

- Implement Python Logging library

- improve status bar messages

- replace config file with QT settings

- plot multiple spectras on top of each other with shift+doubleclick

- add a legend to spectral angle classification output

- plot all profile averages at once

- improve hypbrowser search result navigation so it doesn’t lock up program

- add a little sun icon that when clicked brightens a dark spectral scan by adjusting RGB stretching

- improve accuracy of x axis on Spectra Manager module plot (don’t use np.linspace)

- RGB value presets including (accurate, USGS default, and custom)

References

|  |  |
| --- | --- |
| [1] | Nasa.gov, "Tour of the Electromagnetic Spectrum - Gamma Rays," [Online]. Available: https://science.nasa.gov/ems/12\_gammarays. |
| [2] | Nasa.gov, "Earth Observing-1: Ten Years of Innovation," [Online]. Available: https://earthobservatory.nasa.gov/features/EO1Tenth/page3.php. |
| [3] | "Hyperspectral imaging," [Online]. Available: https://en.wikipedia.org/wiki/Hyperspectral\_imaging. |
| [4] | USGS, "USGS - Earth Observing One (EO-1) - Hyperion," [Online]. Available: https://www.usgs.gov/centers/eros/science/usgs-eros-archive-earth-observing-one-eo-1-hyperion?qt-science\_center\_objects=0#qt-science\_center\_objects. [Accessed May 2021]. |
| [5] | USGS, "EarthExplorer," [Online]. Available: https://earthexplorer.usgs.gov/. |
| [6] | "Hyperspectral Applications," [Online]. Available: https://resonon.com/applications. [Accessed May 2021]. |
| [7] | ArcGIS Developers, "Landcover mapping using hyperspectral imagery and deep learning," [Online]. Available: https://developers.arcgis.com/python/sample-notebooks/landcover-classification-using-hyperspectral-imagery-and-deep-learning/. |
| [8] | Resonan, "Resonon Product Catalog," September 2021. [Online]. Available: https://resonon.com/content/files/Resonon-Product-Catalog\_20Sept2021.pdf. |
| [9] | "Earth Data Analytics Online Certificate," [Online]. Available: https://www.earthdatascience.org/courses/earth-analytics/multispectral-remote-sensing-data/introduction-multispectral-imagery-r/#spatial-resolution. |
| [10] | "Spectral Unmixing - An Overview," [Online]. Available: https://www.sciencedirect.com/topics/earth-and-planetary-sciences/spectral-unmixing. [Accessed May 2021]. |
| [11] | "ArcGIS for Desktop," [Online]. Available: https://desktop.arcgis.com/en/arcmap/10.3/manage-data/raster-and-images/bil-bip-and-bsq-raster-files.htm. |
| [12] | IEEE, "GRSS/SC/Hyperspectral Working Group," May 2021. [Online]. Available: https://sagroups.ieee.org/sagroups-hyperspectral/. |
| [13] | Resonon, "Hyperspectral Software," 2021. [Online]. Available: https://resonon.com/software. |
| [14] | "Prediktera - Hyperspectral Imaging Software," 2021. [Online]. Available: https://prediktera.com/. |
| [15] | L3 Harris Geospatial, "ENVI - The Leading GeoSpatial Image Analysis Software," [Online]. Available: https://www.l3harrisgeospatial.com/Software-Technology/ENVI. |
| [16] | QGIS, "Welcome to the QGIS Project!," 2021. [Online]. Available: https://www.qgis.org/en/site/. |
| [17] | "GDAL," 2021. [Online]. Available: https://gdal.org/. |
| [18] | Purdue, "MultiSpec," 2021. [Online]. Available: https://engineering.purdue.edu/~biehl/MultiSpec/. |
| [19] | J. J. &. E. Angelopoulou, 2021. [Online]. Available: http://gerbilvis.org/. |
| [20] | NASA.gov, "Landsat Science," [Online]. Available: https://landsat.gsfc.nasa.gov/landsat-9/landsat-9-overview. [Accessed 29 November 2021]. |
| [21] | E. O. C. o. DLR, "EnMAP," [Online]. Available: https://www.enmap.org/. [Accessed 29 11 2021]. |
| [22] | "ASI | Agenzia Spaziale Italiana," [Online]. Available: https://www.asi.it/en/earth-science/prisma/. [Accessed 29 November 2021]. |
| [23] | Geoinformers, "All about HySIS of ISRO," [Online]. Available: https://www.geoinformers.com/post/2018-12-02-all-about-hysis-hyperspectral-imaging-satellite-of-isro. [Accessed 29 November 2021]. |
| [24] | D. Li, M. Wang and J. Jiang, "China’s high-resolution optical remote sensing satellites and their mapping applications," *Geo-spatial Information Science,* vol. 24, no. 1, pp. 85-94, 2021. |
| [25] | A. Jones, "Space.com," 10 September 2021. [Online]. Available: https://www.space.com/china-long-march-4c-launches-gaofen-5-02. |
| [26] | S. Chen, "The hidden man fuelling China’s military ambitions: Xiang Libin honoured for work on ‘super camera’ to aid spy satellites," *South China Morning Post,* 12 January 2016. |
| [27] | J. Lin and P. Singer, "China To Launch Powerful Civilian Hyperspectral Satellite," *Popular Science,* 25 January 2016. |
| [28] | USGS.gov, "NDVI, the Foundation for Remote Sensing Phenology," [Online]. Available: https://www.usgs.gov/core-science-systems/eros/phenology/science/ndvi-foundation-remote-sensing-phenology?qt-science\_center\_objects=0#qt-science\_center\_objects. |
| [29] | The Qt Company, "Signals & Slots," [Online]. Available: https://doc.qt.io/qt-5/signalsandslots.html. [Accessed 1 December 2021]. |
| [30] | USGS.gov, "USGS High Resolution Spectral Library," 2021. [Online]. Available: https://www.usgs.gov/centers/gggsc/science/usgs-high-resolution-spectral-library?qt-science\_center\_objects=0#qt-science\_center\_objects. |
| [31] | S. Rashmi, S. Addamani and A. Ravikiran, "Spectral Angle Mapper Algorithm for Remote Sensing Image Classification," *IJISET - International Journal of Innovative Science, Engineering & Technology,* vol. 1, no. 4, pp. 201-205, 2014. |

## Appendix I – Installing HyperGix

## Appendix II – Installing GDAL

A helpful guide to installing GDAL on Windows 10 can be found at the following link:

<https://towardsdatascience.com/python-and-gdal-installation-automated-for-windows-10-f22686595447>

## Appendix III – HyperGix Source Code

Source code for HyperGix is available at <https://github.com/johngallino/HyperGix>