# **SEVIR Tutorial, TEAM 5**

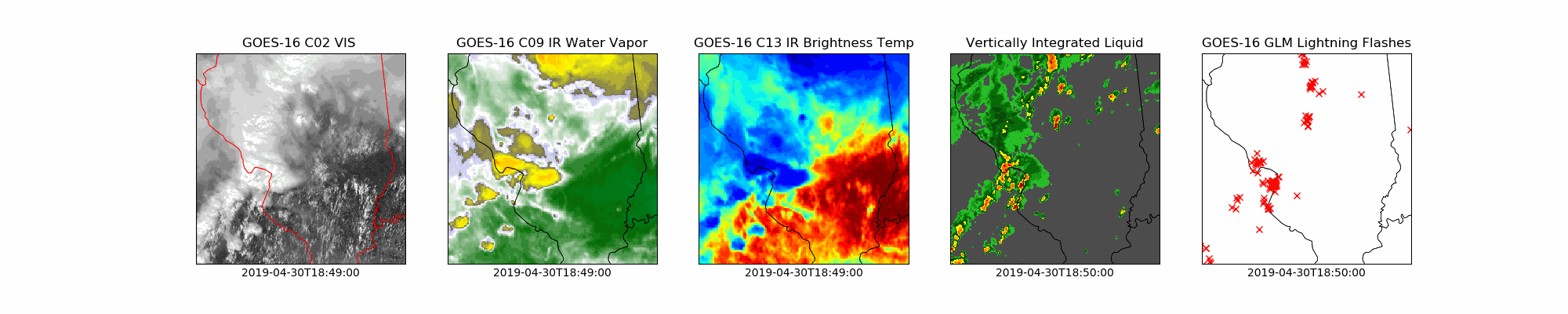
| **NAME** | **NUID** |
| --- | --- |
| SAI VENKATA SAMANTH KODURU | 2983840 |
| SHANTAN DADI | 2927718 |

**Abstract**

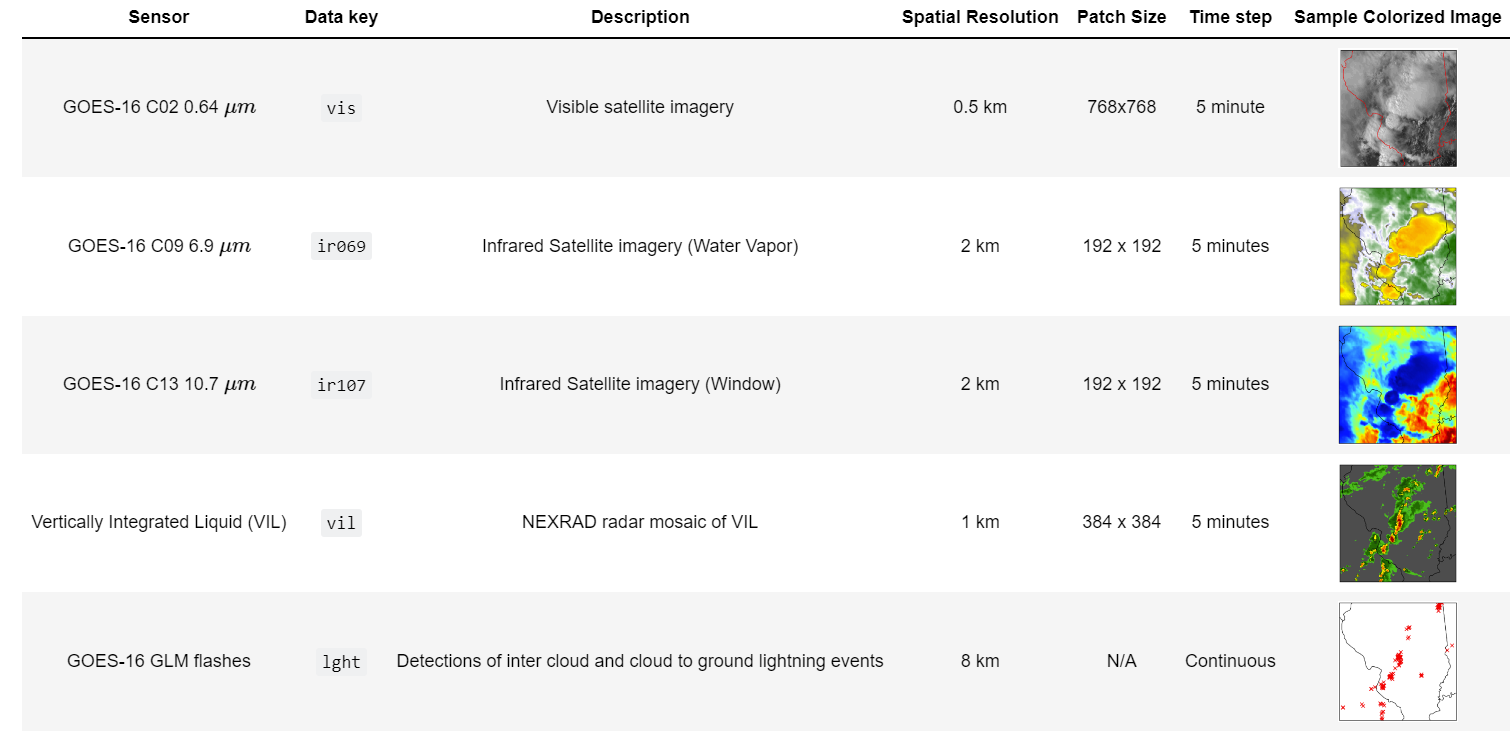
Modern deep learning approaches have shown promising results in meteorological applications like nowcasting, disaster prediction, weather anomalies and several others. In order to successfully train and validate these complex algorithms, large and multi-modal datasets containing high-resolution imagery are required. Petabytes of weather data, such as from the Geostationary Environmental Satellite System (GOES) and the Next-Generation Radar (NEXRAD) system, are available to the public; however, the size and complexity of these datasets is an issue to develop and training deep models. To help address this problem, we introduce the Storm EVent ImagRy (SEVIR) dataset - a single, rich dataset that combines spatially and temporally aligned data from multiple sensors, along with baseline implementations of deep learning models and evaluation metrics, to accelerate new algorithmic innovations.

The notebook here is a walkthrough of the following:

* Downloading SEVIR files
* Accessing SEVIR
* Data Visualization SEVIR
* Data Georeferencing SEVIR Events



*Figure 1: Sample event in the SEVIR dataset. SEVIR contains over 10,000 sets of image sequences similar to the one shown here.*



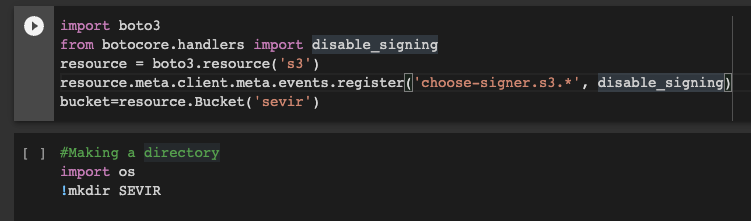
## 

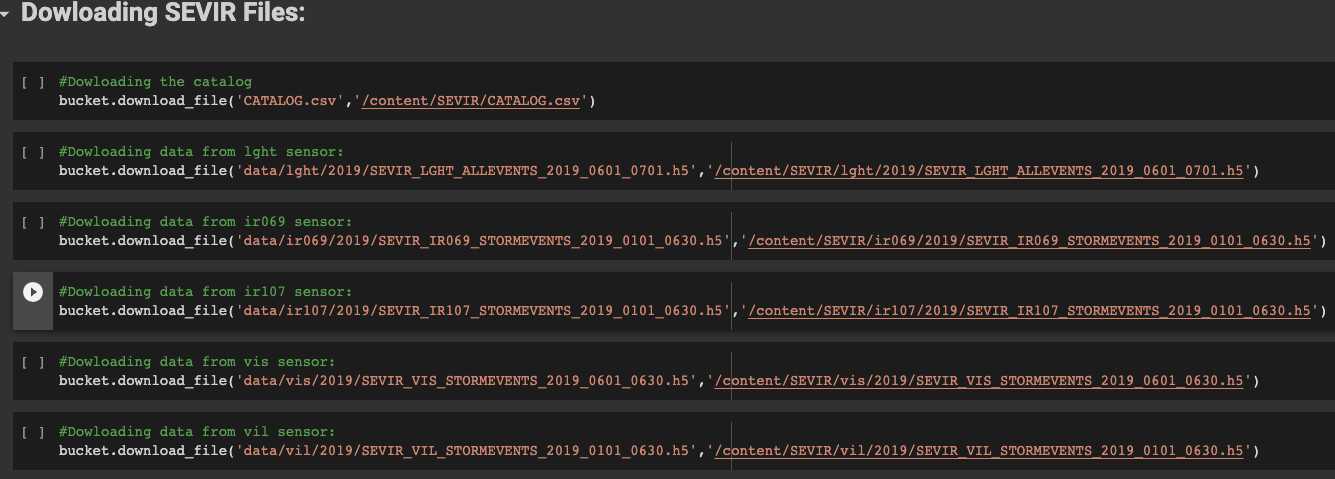
# Downloading SEVIR Data & Files

Duration: 10:00

The SEVIR dataset consists of 5 modalities from different sensors. We have used AWS SDK for Python (Boto3) to create, configure, and manage AWS services, such as Amazon Elastic Compute Cloud (Amazon EC2) and Amazon Simple Storage Service (Amazon S3).

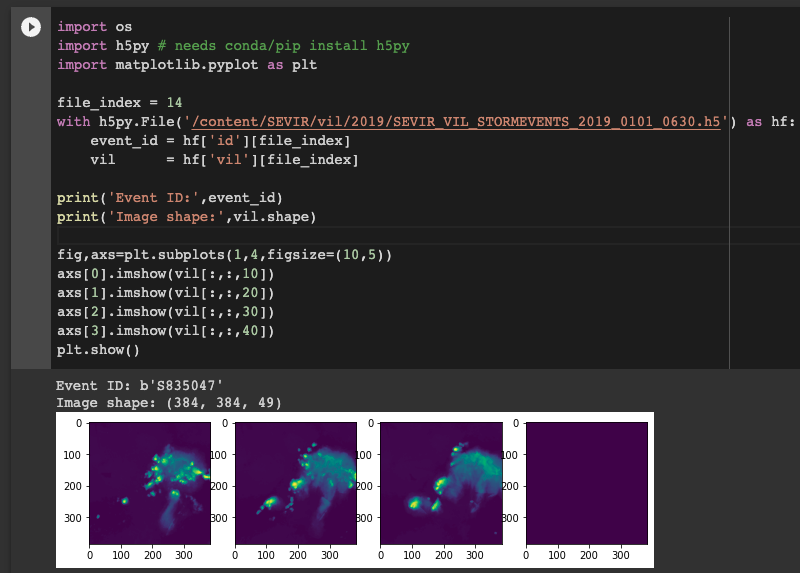
Once we have access to AWS S3 bucket, we find and retrieve the vis, ir069, ir107, vil, and lght files for the specific event ID of interest. Our investigation refers to Event ID=835047.





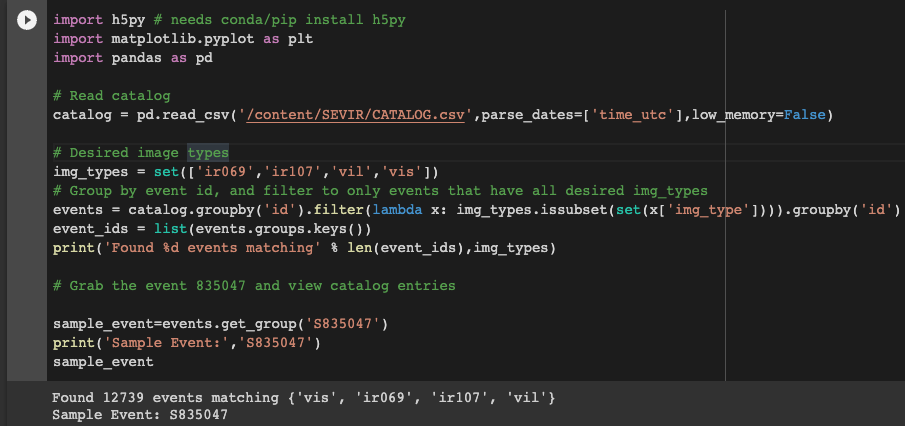
# Acessing a SEVIR event

The following code shows how to open a data file containing a catalog and use h5py: Disk to visualize event frames captured from a single type as if they were real NumPy arrays . Access and visualize event 835047 from the vil sensor data..



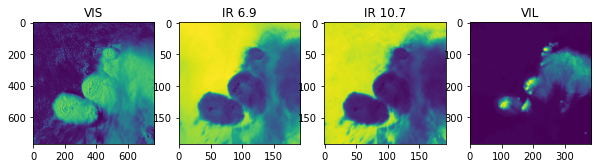
You can use the catalog to access event data collected from multiple sensor types. Find all available sensors for each event by grouping by the ID column. Some events in SEVIR are not covered by all sensors, so we also need to filter out events that are not covered by all requested image types.

The following code shows how to extract events observed by four image types: vis, vil, ir069, and ir107. Again, we're only interested in one of the long list of events available in the CATALOG.csv file, so we filter to find the vis, vil, ir069, and ir107 images for that event.



# Visualizing SEVIR Data

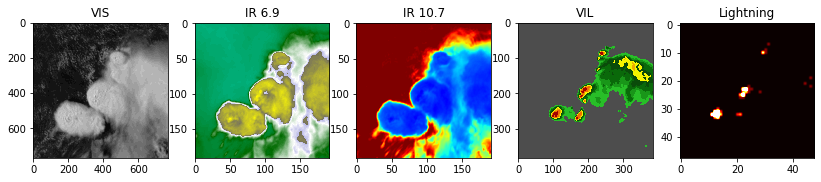
So far, only data from vis, vil, ir069, and ir107 have been recorded. To display the flashlight among other image types in the data collected in the previous results, you must first convert the flashlight to a raster type. To achieve this, the flash data is converted to flash counts per pixel per 5 minute frame with the following code.





In previous visualizations, data was displayed with a default color map. The repository at https://github.com/MIT-AI-Accelerator/eie-sevir contains color maps that can be used to display color images similar to those expected to appear in SEVIR data. I'm here. The sevir module from this repository must be in your path for this code to work. To access the included libraries, clone the repository.

**Visualizing the five sensor images using the SEVIR specific colormap, we have the resultant images as below:**



# **Georeferencing SEVIR Events**

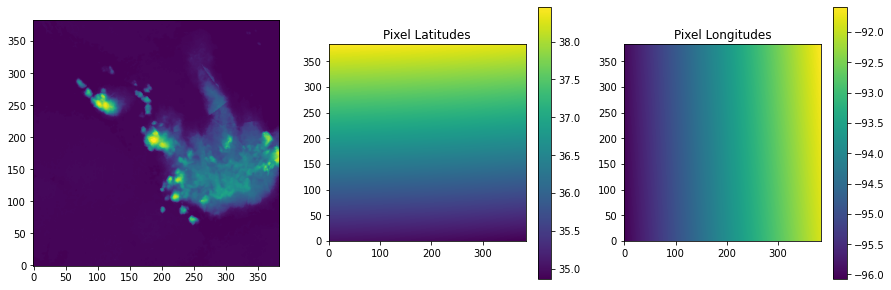
In many applications it's necessary to know the lat/lon positions of pixels in a SEVIR patch. The catalog provides the required information to perform the georeferencing the data.

Currently, all patches in SEVIR use the same map projection, a Lambert Equal Area projection centered at (lat,lon)=(38,-98). This specifics of the projection are stored in the proj column:

The projection formula for this particular instance is encoded in the following class. With this, you can change the coordinates for each event in SEVIR between pixel coordinates and latitude/longitude coordinates.

**Georeferencing an event**

Using this class, we can get a grid of latitudes and longitudes for our event\_id=835047.

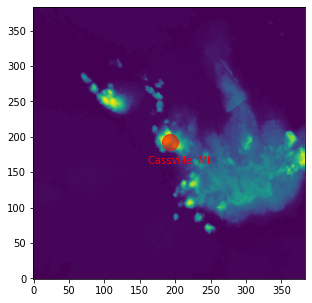


This particular domain overlaps with Southwestern Missouri. We can add markers for particular locations by converting from lat/lon coordinates into the image pixel coordinates:

**Visualizing a satellite image by are co-ordinates:**

The code below helps in representing the vil(Vertically Integrated Liquid) over Cassaville, MI, this is a sattelite image using the NEXRAD radar mosiac sensor.

Vertically integrated liquid (VIL) is an estimate of the total mass of precipitation in the clouds. The measurement is obtained by observing the reflectivity of the air which is obtained with weather radar.



You can further improve the image above by adding details such as state borders. Thanks to a set of Python libraries that compute these transformations, you can display georeferenced results with features such as coastlines, state boundaries, and other attractive map features. A popular module for computing geo-transformations is pyproj. The Basemap and Cartopy plotting libraries are excellent. The cell below shows how to add state boundaries using the basemap library instead of the old georeferencing method.

**Conclusion**

This notebook provides detailed information on the SEVIR dataset, from how to retrieve SEVIR data from an AWS S3 bucket using boto3, to how to visualize all types of sensor data (vil, ir069, ir107, lght). I explained the research. We also used the CATALOG.csv file to georeference the SEVIR data and plot the actual event locations as Cassville, Michigan. A basemap projection was used to create a more readable representation using state boundaries.

# **References**

* NOAA's Storm Event Database: <https://www.ncdc.noaa.gov/stormevents/ftp.jsp>
* https://nbviewer.org/github/MIT-AI-Accelerator/eie-sevir/blob/master/examples/SEVIR\_Tutorial.ipynb
* <https://github.com/MIT-AI-Accelerator/sevir_challenges>
* <https://raw.githubusercontent.com/MIT-AI-Accelerator/eie-sevir/master/CATALOG.csv>
* <https://www.ncei.noaa.gov/pub/data/swdi/stormevents/csvfiles/>
* https://www.ncei.noaa.gov/pub/data/swdi/stormevents/csvfiles/Storm-Data-Bulk-csv-Format.pdf
* https://sevir.mit.edu/sites/default/files/About\_SEVIR.pdf
* https://proceedings.neurips.cc/paper/2020/file/fa78a16157fed00d7a80515818432169-Paper.pdf