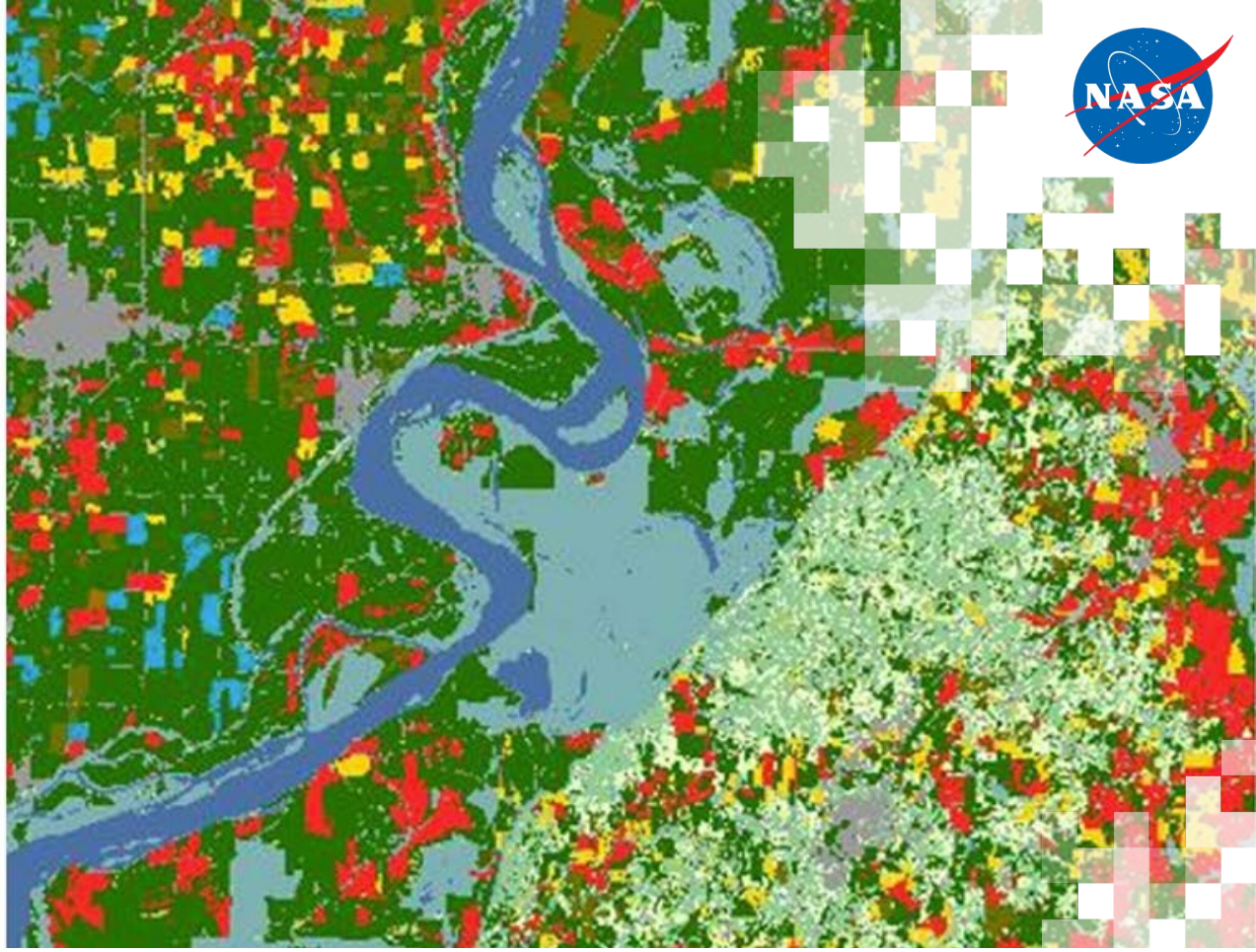
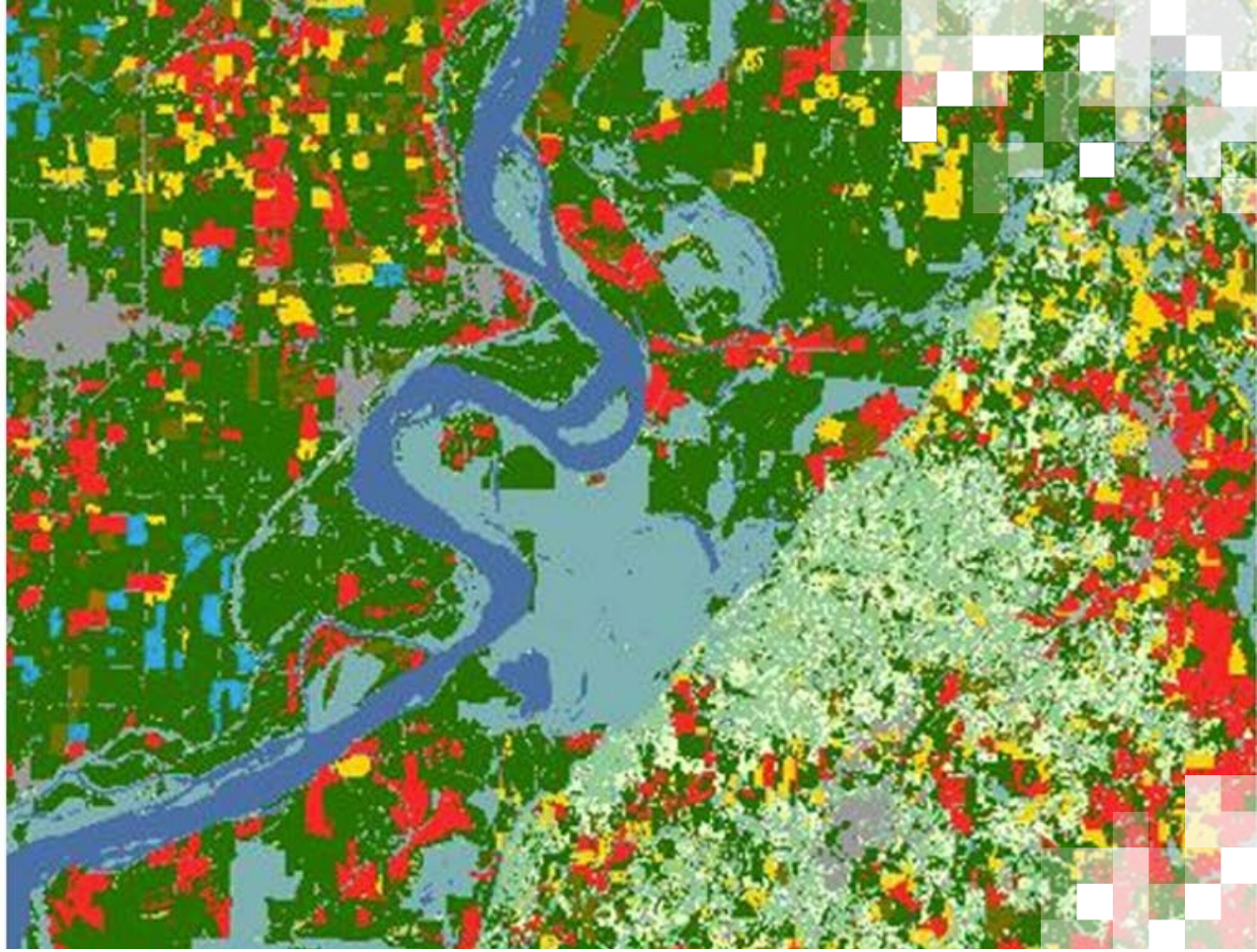


Large Scale Applications of Machine Learning using Remote Sensing for Building Agriculture Solutions

Part 1: Data Preparation of Imagery & Labels for Large-Scale ML Modeling

John Just (Deere & Co., Iowa State University), Erik Sorensen (Deere & Co.)

March 5, 2024



About ARSET

About ARSET

- **ARSET provides accessible, relevant, and cost-free training on remote sensing satellites, sensors, methods, and tools.**

- Trainings include a variety of applications of satellite data and are tailored to audiences with a variety of experience levels.


AGRICULTURE


CLIMATE & RESILIENCE


DISASTERS


ECOLOGICAL CONSERVATION

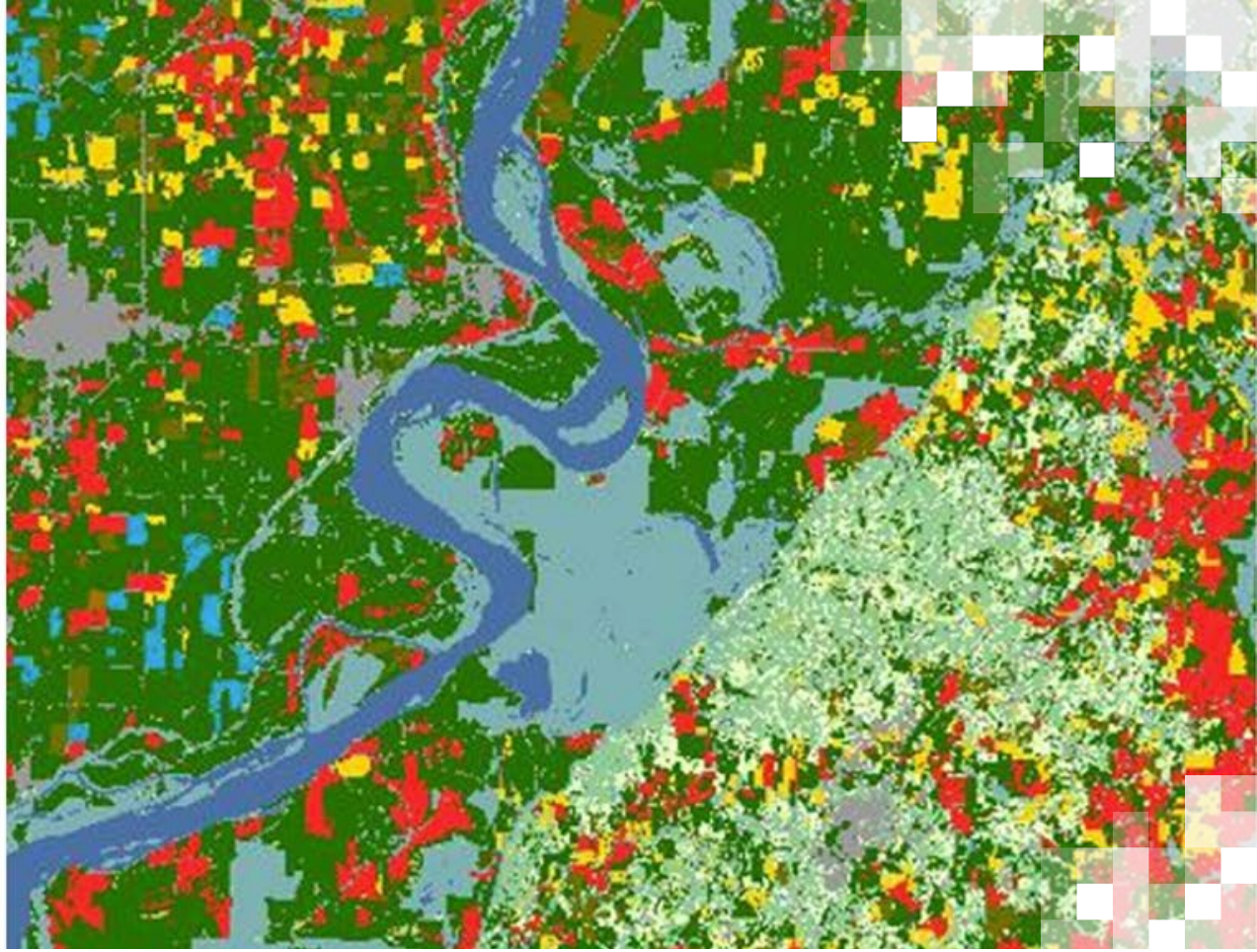

HEALTH & AIR QUALITY


WATER RESOURCES

About ARSET Trainings

- Online or in-person
- Live and instructor-led or asynchronous and self-paced
- Cost-free
- Bilingual and multilingual options
- Only use open-source software and data
- Accommodate differing levels of expertise

- Visit the ARSET website to learn more.

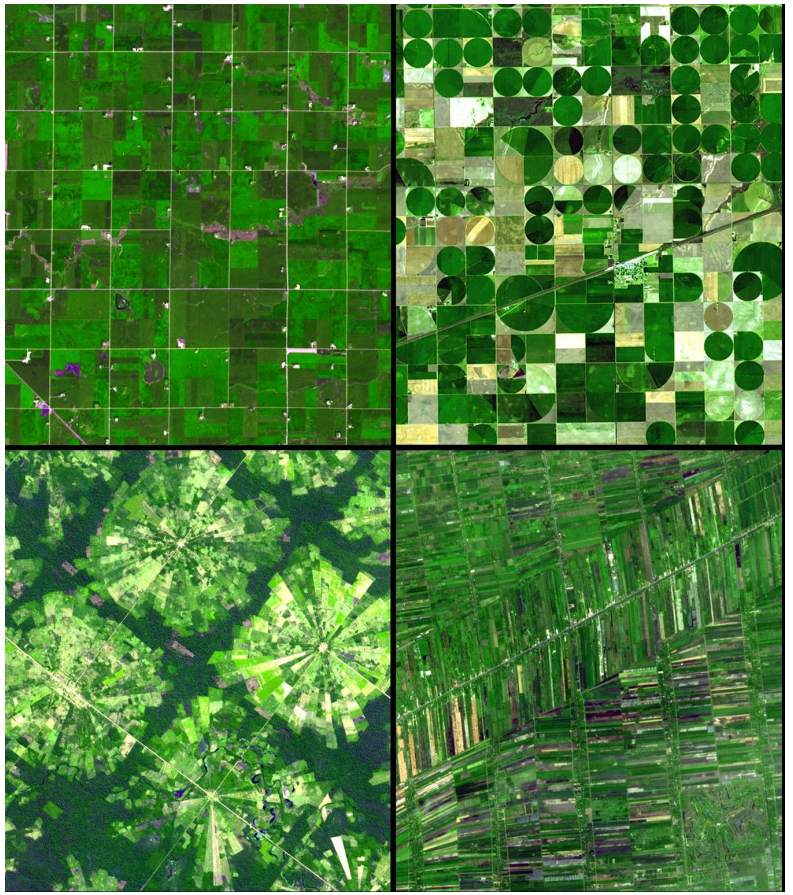


Large Scale Applications of Machine Learning using Remote Sensing for Building Agriculture Solutions

Overview

Motivation for Training

- Timely and accurate in-season crop maps at local to regional scales is crucial for agricultural decision-making and management.
- Irregularly-spaced time-series are common with optical satellite images.
- Training robust models on remote sensing data often requires very large data, but processing and training is complex.
- The Cropland Data Layer (CDL, USDA–NASS) only gives estimates of the types of crops released to the public a few months after the end of the growing season, and not their sequence or timing (e.g., for double crops)



Montage of images shows differences in field geometry and size in different parts of the world. Image credit: NASA (Instrument: Terra – ASTER)

Training Learning Objectives

By the end of this training series, participants will be able to:

- Use recommended techniques to download and process remote sensing data from Sentinel-2 and the Cropland Data Layer (CDL) at large scale (> 5GB) with cloud tools (Amazon Web Services [AWS] Simple Storage Service [S3], Databricks, Spark/Pyspark, Parquet)
- Produce interactive plots of maps, tables, time series, etc. for investigation & verification of data and models
- Filter data from both the measured (satellite images) and target (CDL) domains to serve modeling objectives based on quality factors, land classification, area of interest (AOI) overlap, and geographical location.
- Build training pipelines in TensorFlow to train machine learning algorithms on large scale remote sensing/geospatial datasets for agricultural monitoring
- Utilize random sampling techniques to build robustness into a predictive algorithm while avoiding information leakage across training/validation/testing splits

Prerequisites

- Fundamentals of Remote Sensing
- Crop Classification with Time Series, Part 2
- Sign up for and access Databricks Community Edition

Training Outline

Part 1

Data Preparation of Imagery & Labels for Large-Scale ML Modeling

March 05, 2024

Part 2

Data Loaders for Training ML Models on Irregularly-Spaced Time-Series of Imagery

March 12, 2024

Part 3

Training & Testing ML Models for Irregularly-Spaced Time Series of Imagery

March 19, 2024

Homework

Opens March 19 – **Due April 1** – Posted on Training Webpage

A certificate of completion will be awarded to those who attend all live sessions and complete the homework assignment(s) before the given due date.

How to Ask Questions

- Please put your questions in the Questions box and we will address them at the end of the webinar.
- Feel free to enter your questions as we go. We will try to get to all the questions during the Q&A session after the webinar.
- The remainder of the questions will be answered in the Q&A document, which will be posted to the training website about a week after the training.

Part 1 – Trainers

John Just

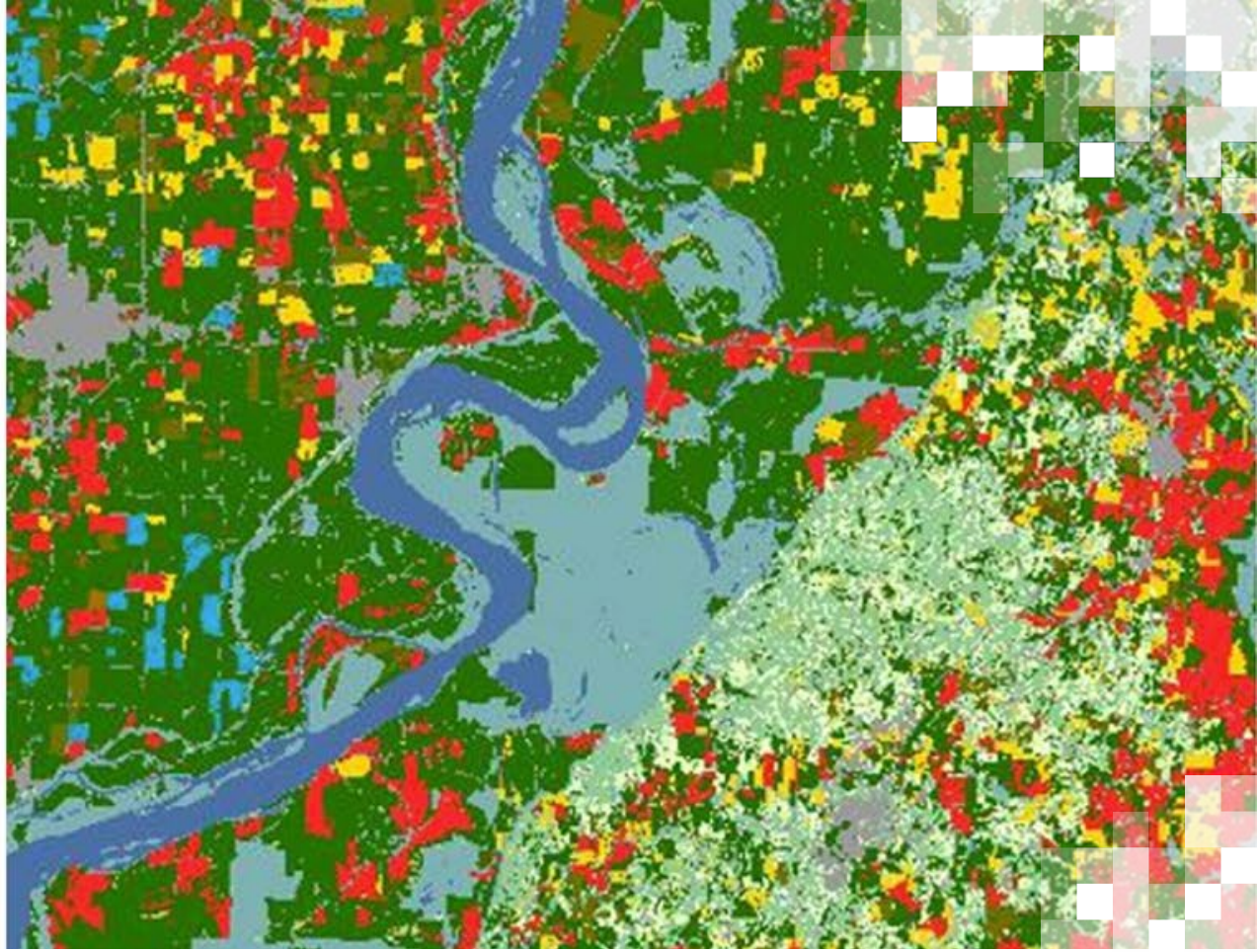
Principal Data Scientist

John Deere

Erik Sorensen

Senior Data Scientist

John Deere



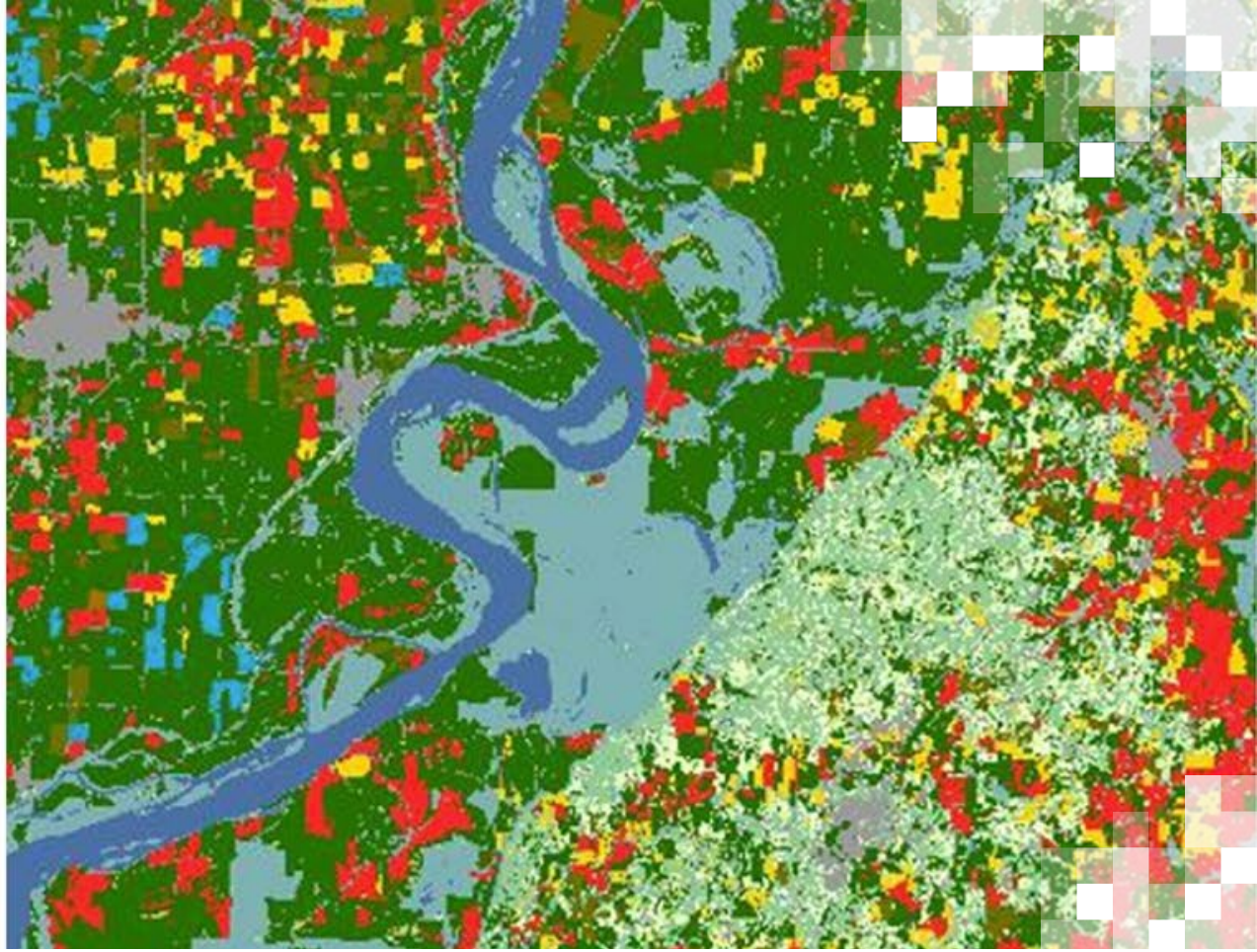
Large Scale Applications of Machine Learning using Remote Sensing for Building Agriculture Solutions

Part 1: Data Preparation of Imagery & Labels for Large-Scale ML Modeling

Part 1 Objectives

By the end of Part 1, participants will be able to programmatically:

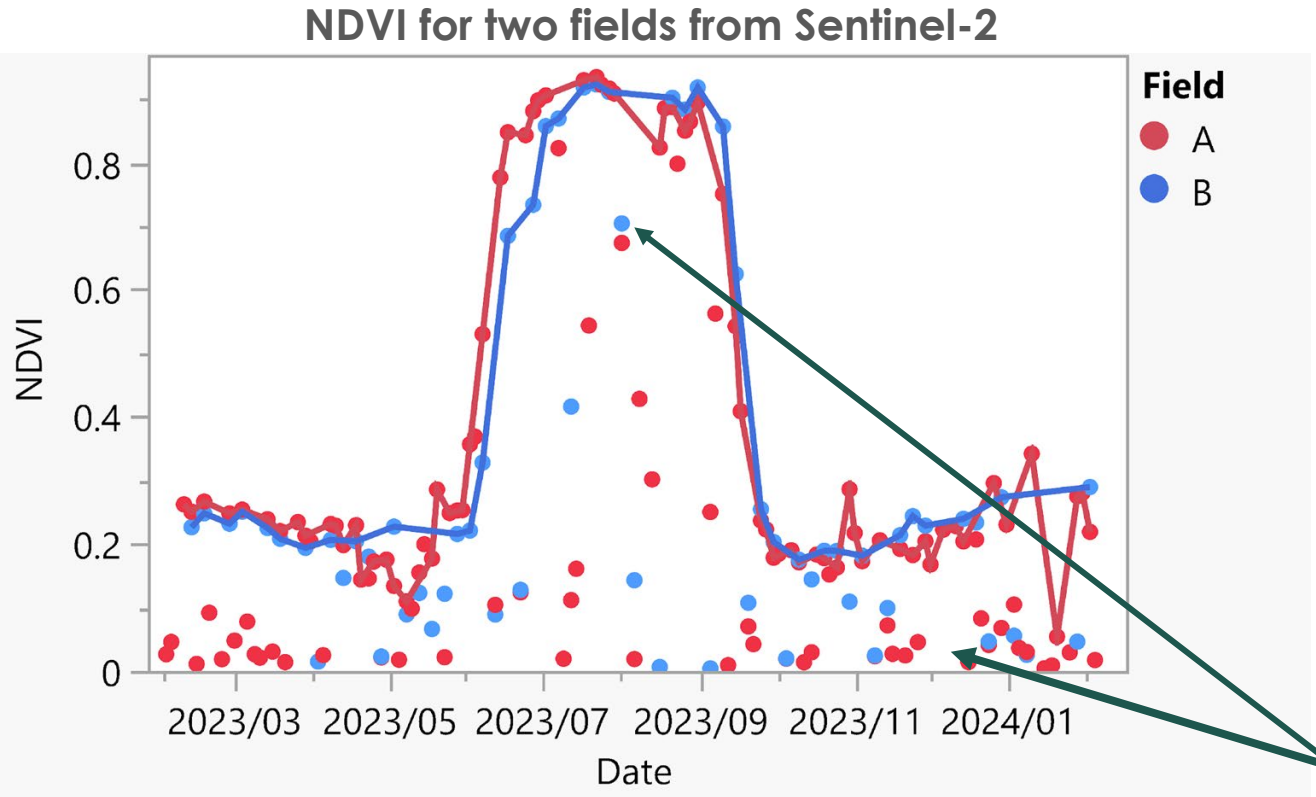
- Submit lists of boundaries to the NASS API and retrieve CDL rasters back.
- Subsample and visualize retrieved data from CDL with interactive spatial images and other statistical plots.
- Obtain Sentinel-2 raster files for a given area and timeframe corresponding to the retrieved CDL data and manipulate the Sentinel-2 rasters into tables in preparation for analysis and model training.
- Verify correct processing of data via various interactive plots (e.g. time series of pixels of various land covers).



Part 1 Section 1:
Irregularly Spaced Time Series Modeling

Irregularly Spaced Time Series Modeling

Common due to: orbital geometry, variations in exact orbit timing/geolocation/image extents, and atmospheric disturbances due to clouds/smoke or other random events.



Irregular Spacing/timing

Field	Interval (days)	# Scenes
A	2	65
A	3	65
A	5	8
B	5	67
B	10	3

Points not fit by lines are scenes with cloud cover

A & B are 2 km apart
But A has 2x scenes (coverage) due to orbit path overlap

Motivation for this Example

We propose a **real-time prediction** for the Cropland Data Layer (CDL) as the working example for this tutorial.

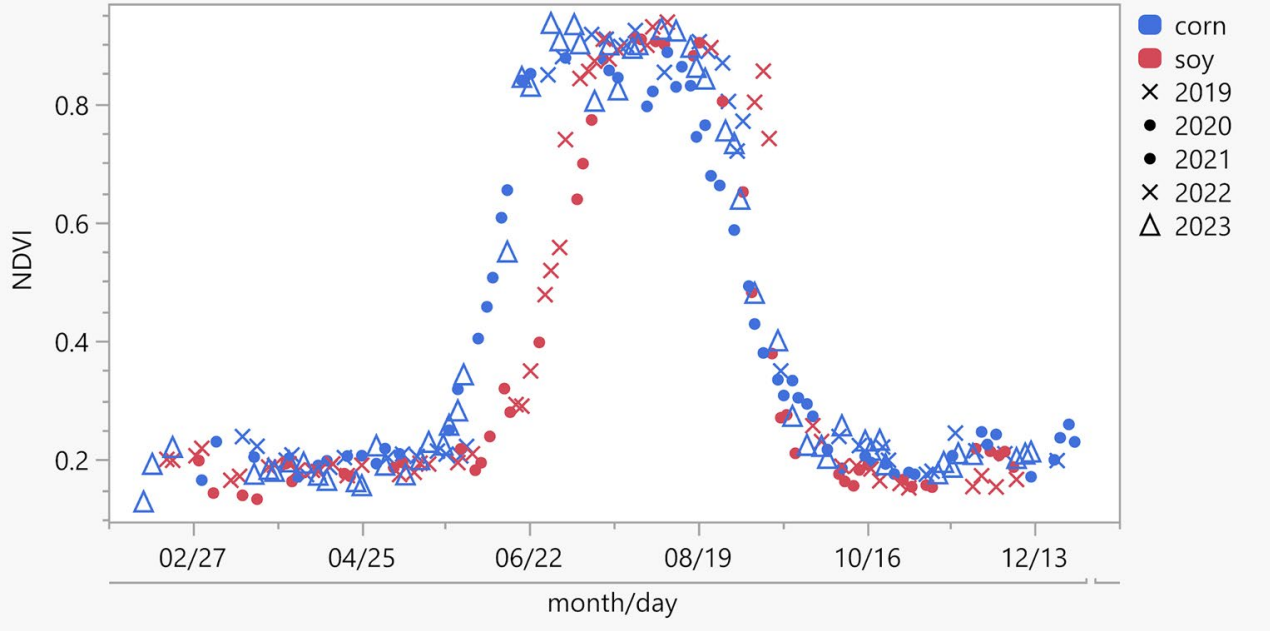
- The CDL algorithm is already using the same (or similar) satellite data sources and irregular spacing/timing to make predictions (documented example of success)
- Labels are readily available via API calls (highly scalable/available).
- Accuracy is well studied & documented
- ***Resulting code & methods are highly transferable to other problems/use cases***

Predicting the CDL in Real-Time

According to the CDL FAQs,

- "The CDL Program uses medium spatial resolution (30 meter) satellite imagery because it's too costly to use higher resolution satellites to perform crop acreage estimation over large areas."
- The CDL is considered confidential and market sensitive during the growing season and cannot be released until after the official NASS year end area county estimates are published in late January/early February following the end of the typical US growing season
- The CDL only gives estimates of the types of crops, but not their sequence or timing (e.g. for double crops)

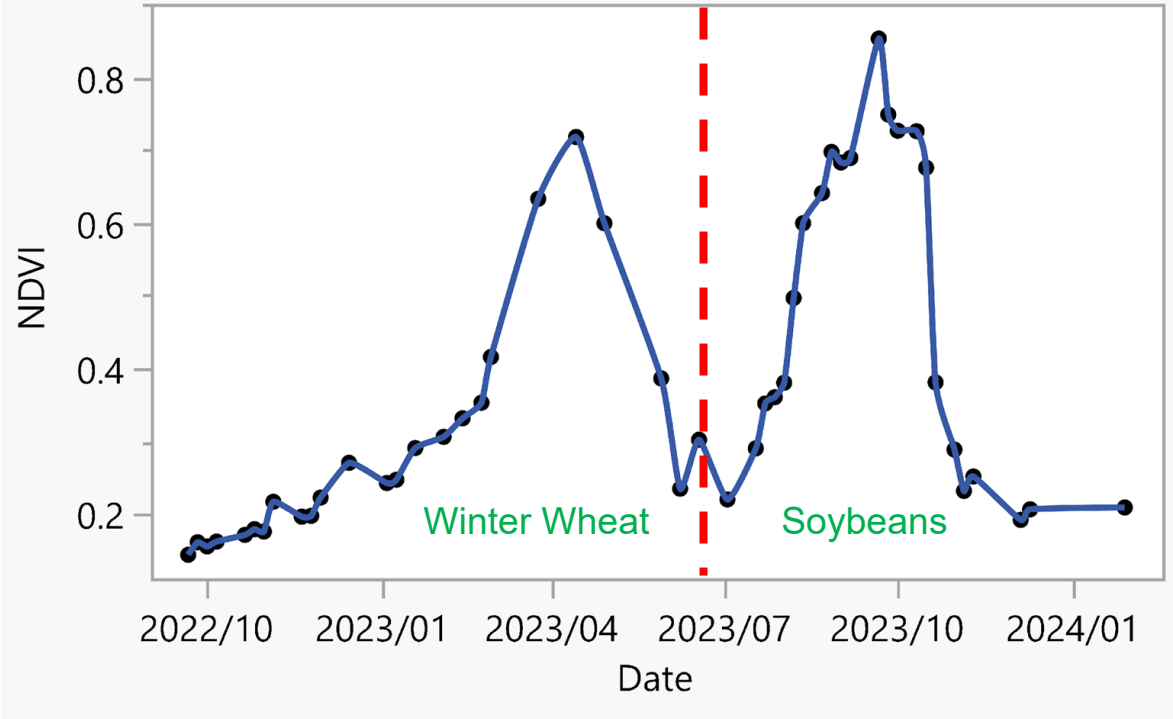
However… do we really need to wait until the following year for accurate estimates?



"By early July we probably are pretty confident in the crop type just from NDVI in this area (probably even more so by using multispectral time series). By late August we are really confident (6 months prior to when CDL is typically released)."

Generalizable Approach

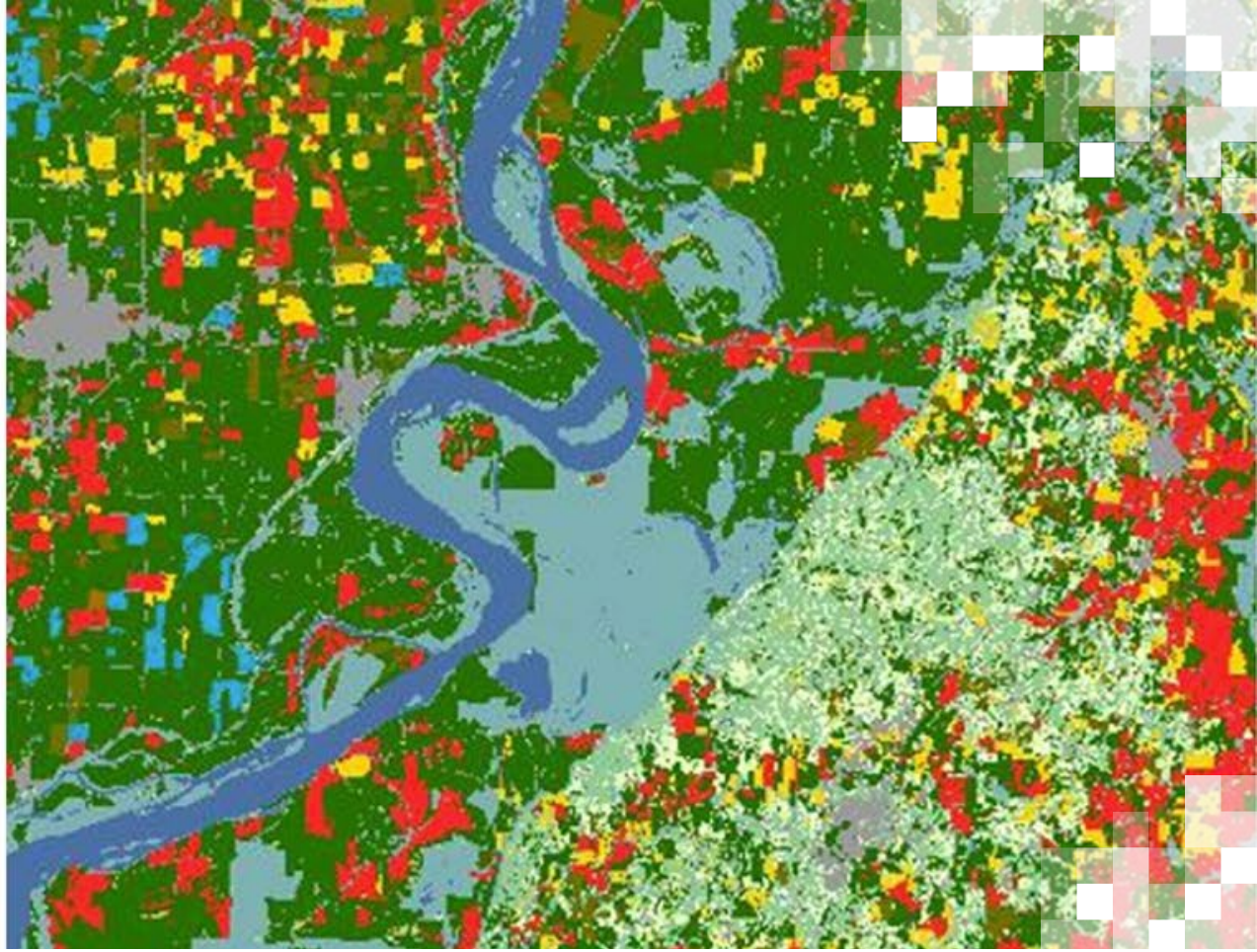
- Robust models require large-scale data management tools and approaches. Leveraging multi-core and multi-machine parallel computing is a necessary step to scale. We demonstrate these tools & approaches with this series.
- Note that a similar approach to crop modeling with the time-series of imagery could be used for estimating the crop health or other time-dependent factors as well (simply substitute the label/target).



By early Spring we are pretty confident in the crop type for areas that planted winter crops and will likely have a good estimate of double cropping by early Fall.

Irregularly-Spaced Time-Series Modeling

- There's a dearth of statistical theory around unevenly-spaced time series, and thus not much for out-of-the-box methods to apply *directly* for such situations.
  - Most common solution is to manipulate the data into a regularly-spaced time-series, then apply standard methods. E.g. interpolation or interval-binning (*the latter being what the CDL algorithm does*).
  - **Note that resulting data format from this part-1 demo will support any modeling approach**.
    - We follow a similar approach as the CDL (binned intervals) for parts 2 & 3 of this demo for data loaders & model training due to simplicity.
- Newer ML sequence models such as transformers ("self-attention") accept positional encodings of inputs/outputs and learn meaningful absolute and relative input (and output) information. This could facilitate direct modeling of unevenly-spaced satellite data, but due to increased complexity we do not incorporate it here.



Part 1 Section 1:
Cropland Data Layer (CDL)

Cropland Data Layer (Contiguous United States)

Best place to find information about it is the USDA NASS FAQs & metadata.

Some relevant info:

- **Model**: decision tree classifier (handles missing, non-continuous, non-normal, nonlinear data, efficient computation). Probabilistic output (argmax to class)
- **Input**: Landsat 8 and 9 OLI/TIRS, ISRO ResourceSat-2 LISS-3, and ESA SENTINEL-2A and -2B. Imagery is downloaded daily with the objective of obtaining at least one cloud-free usable image every two weeks throughout the growing season
- **Ground truth**: FSA Common Land Unit (CLU) for ag/crops & National Land Cover Database (NLCD) for non-ag areas
- **Accuracy**: Generally, 85% to 95% correct for the major crop-specific land cover categories. 30m resolution

CDL Accuracy

As noted earlier, the CDL accuracy is well studied and documented. Below is an excerpt from the USDA quality checks for Arkansas 2022.

- All states from all years can be found USDA NASS Cropland metadata.
- Some crops like sorghum (in this case) may have low accuracy, but they also represent a tiny proportion of the farmland. Training a model specifically focused on certain classes could boost accuracy for those classes, at the expense of others.

USDA National Agricultural Statistics Service, 2022
Arkansas Cropland Data Layer

STATEWIDE AGRICULTURAL ACCURACY REPORT

Crop-specific covers only	*Correct	Accuracy	Error	Kappa
OVERALL ACCURACY**	482475	87.30%	12.70%	0.817

Cover Type	*Correct Pixels	Producer's Accuracy	Omission Error	Kappa	User's Accuracy	Commission Error	Cond'l Kappa
Corn	55159	86.40%	13.60%	0.855	94.10%	5.90%	0.937
Cotton	50682	88.00%	12.00%	0.873	93.20%	6.80%	0.928
Rice	87048	90.30%	9.70%	0.893	96.10%	3.90%	0.957
Sorghum	156	22.70%	77.30%	0.227	77.20%	22.80%	0.772
Soybeans	254301	93.60%	6.40%	0.91	89.60%	10.40%	0.857

*Correct Pixels represents the total number of independent validation pixels correctly identified in the error matrix.
**The Overall Accuracy represents only the FSA row crops and annual fruit and vegetables

Rough calculations of data size for CDL on Contiguous US

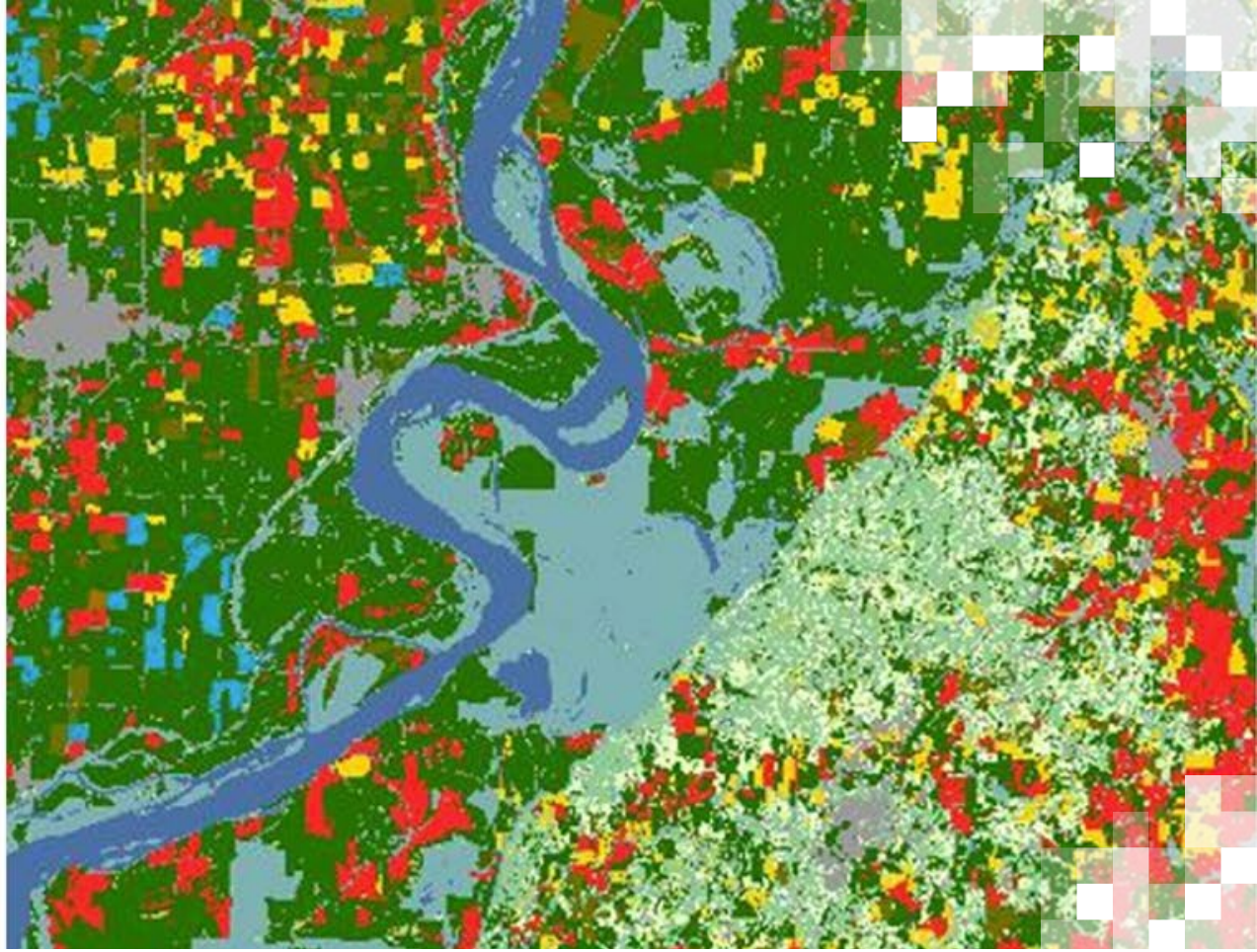
Even though only about 20% of the US is specifically used for crop land, any given model must run across all the US area to classify the land.

Assuming:

- Use Sentinel-2 for land classification at 10m resolution
- 1 cloud-free image every 2 weeks (26 images total per 100m^2)
- 12 bands at 2 bytes (16 bits) per pixel (26 images total per 100m^2)
- ~7,500,000 km^2 of land

≈44 TB of data post-processed to run a predictive model. While not trivial, 10TB drives only cost $200. using 30$m^2$ resolution drops this down to ~5TB to work with.

If only using Sentinel-2, roughly 28TB of raster data must be downloaded and processed over that land since Sentinel-2 tiles are 110 x 110 km^2 and have 12 bands, amounting to roughly 640MB per scene, that occur every 5 days.



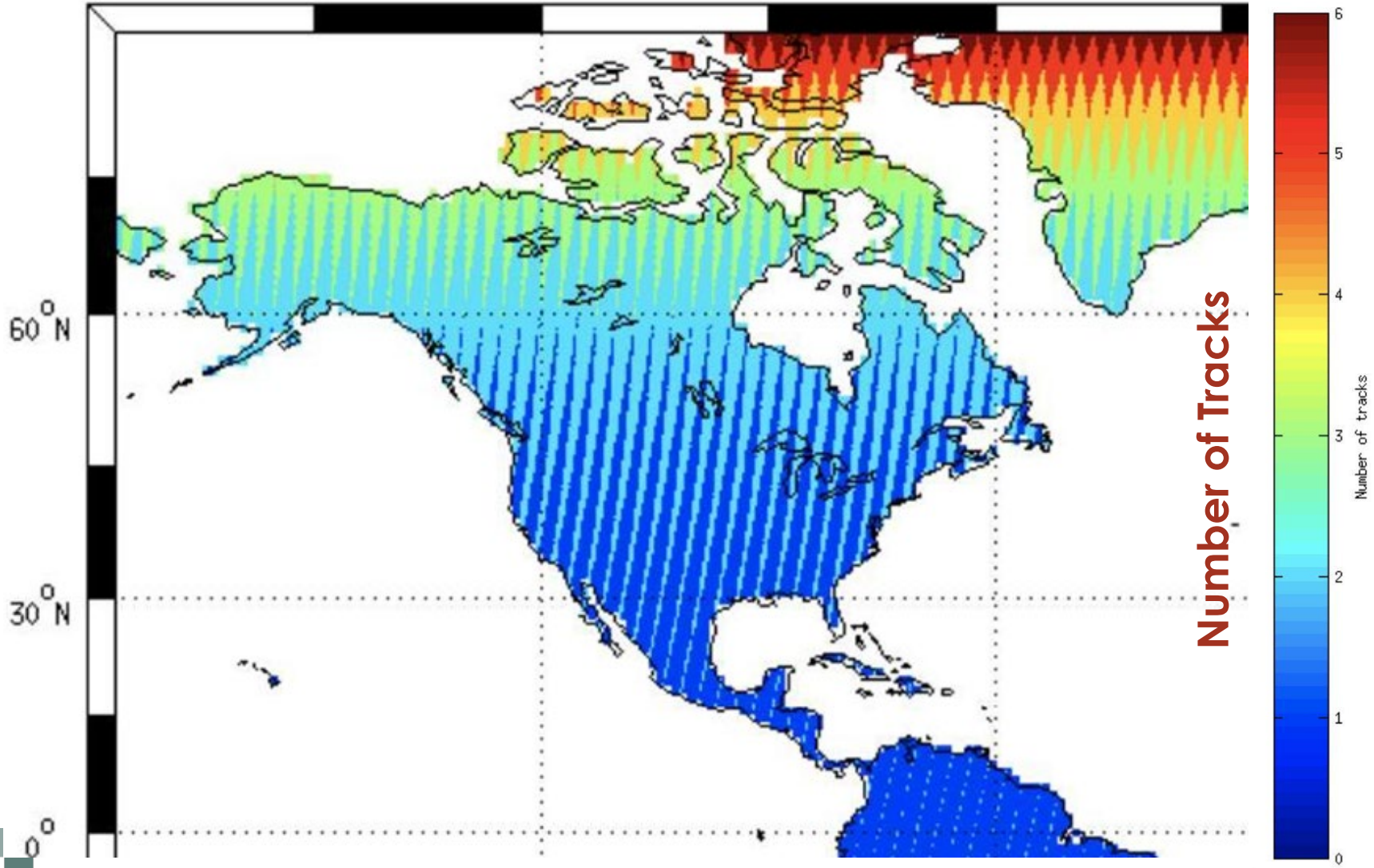
Part 1 Section 2:
Sentinel-2 Optical Data

Factors Affecting Quality & Temporal Spacing of Satellite Data

Example factors affecting data:

- Irregularity
  - **Orbit path overlap** – closer to poles leads to increased coverage
  - **Orbit path/image capture repeatability** – the exact position of the image can vary east/west and result in areas on the edge of scenes to have more uncertainty in coverage.
  - **Thick cloud cover** – when present and identified (thus ignored) there is a gap in coverage.
  - **SCL errors** – when we ignore data from scenes due to SCL category but it's wrong, we introduce unnecessary gaps in coverage.
- Quality
  - **Thin cloud haze** – cloud cover isn't Boolean, it's a gradient. Sometimes hard to identify when thin (it alters the reflectance values and may not be caught)
  - **Tiling system overlap** – at the edge of tiles (which is how the data is stored & queried) there is overlap and slight differences between the values for the same scene and location in different tiles.
  - **Geolocation/georeferencing** – location of pixels can be incorrect and vary by more than the size of a pixel (resulting in wrong information for a point location)

Sentinel-2 Orbit Path Overlap



Reference: https://sentinels.copernicus.eu/web/sentinel/user-guides/sentinel-2-msi/revisit-coverage

Nominal availability from Sentinel-2 is 1 image every 5 days. However, overlap between adjacent orbits increases further from the equator. Thus, certain parts of US on the borders of orbits get up to 2x coverage per satellite and results in intervals of 2 or 3 days instead of 5.

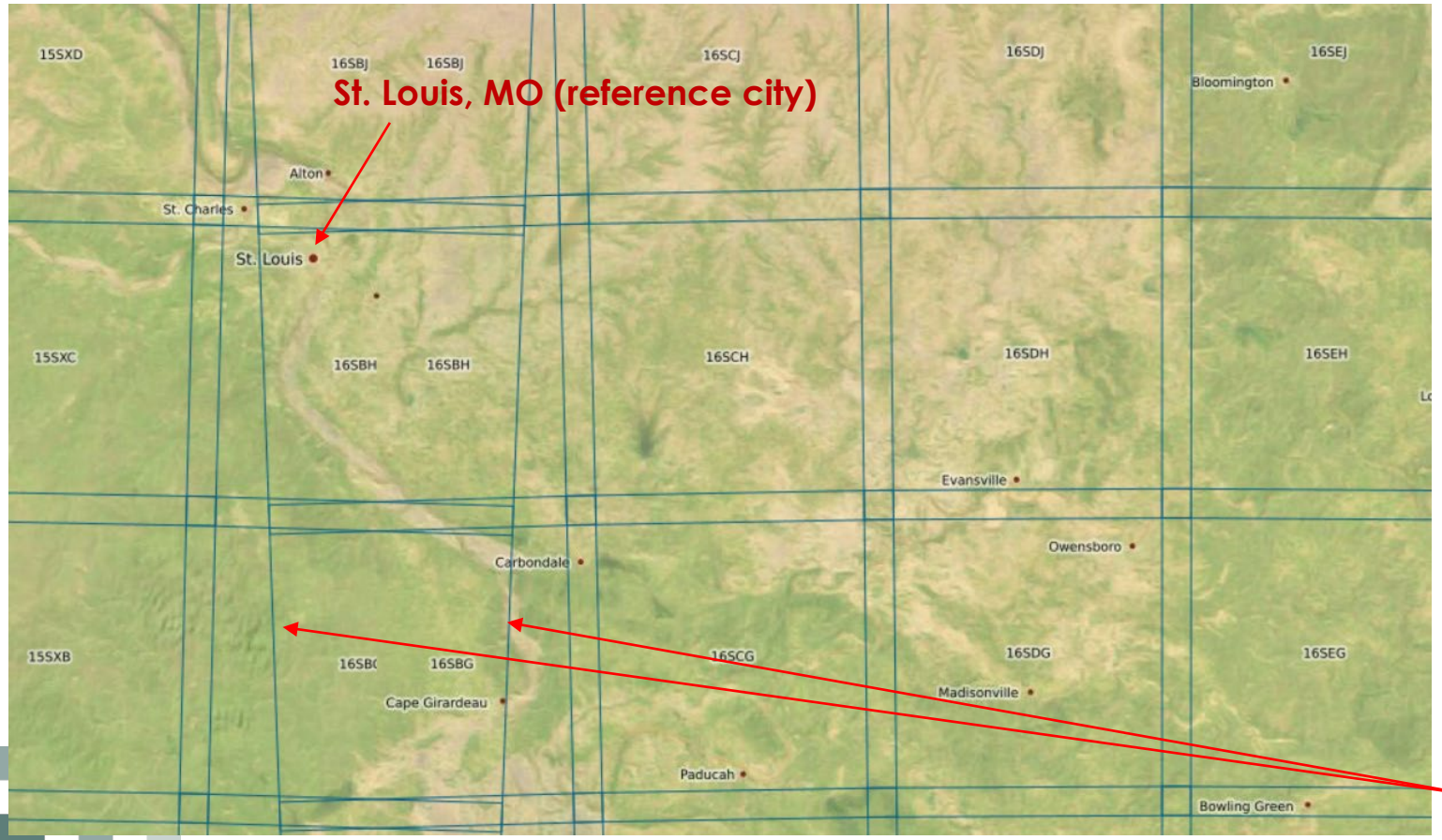
Orbit Path Repeatability

Within a distance of 1km, the same nominal orbit path has several actual orbit path variations. This can again affect availability of imagery.

Sentinel-2 Tile System Overview

Some considerations to be aware of when processing S2 data for analysis & modeling:

- Scenes captured from Sentinel-2 are processed and made available in a unique tiling system that is a slightly modified version of the military grid reference system (MGRS).

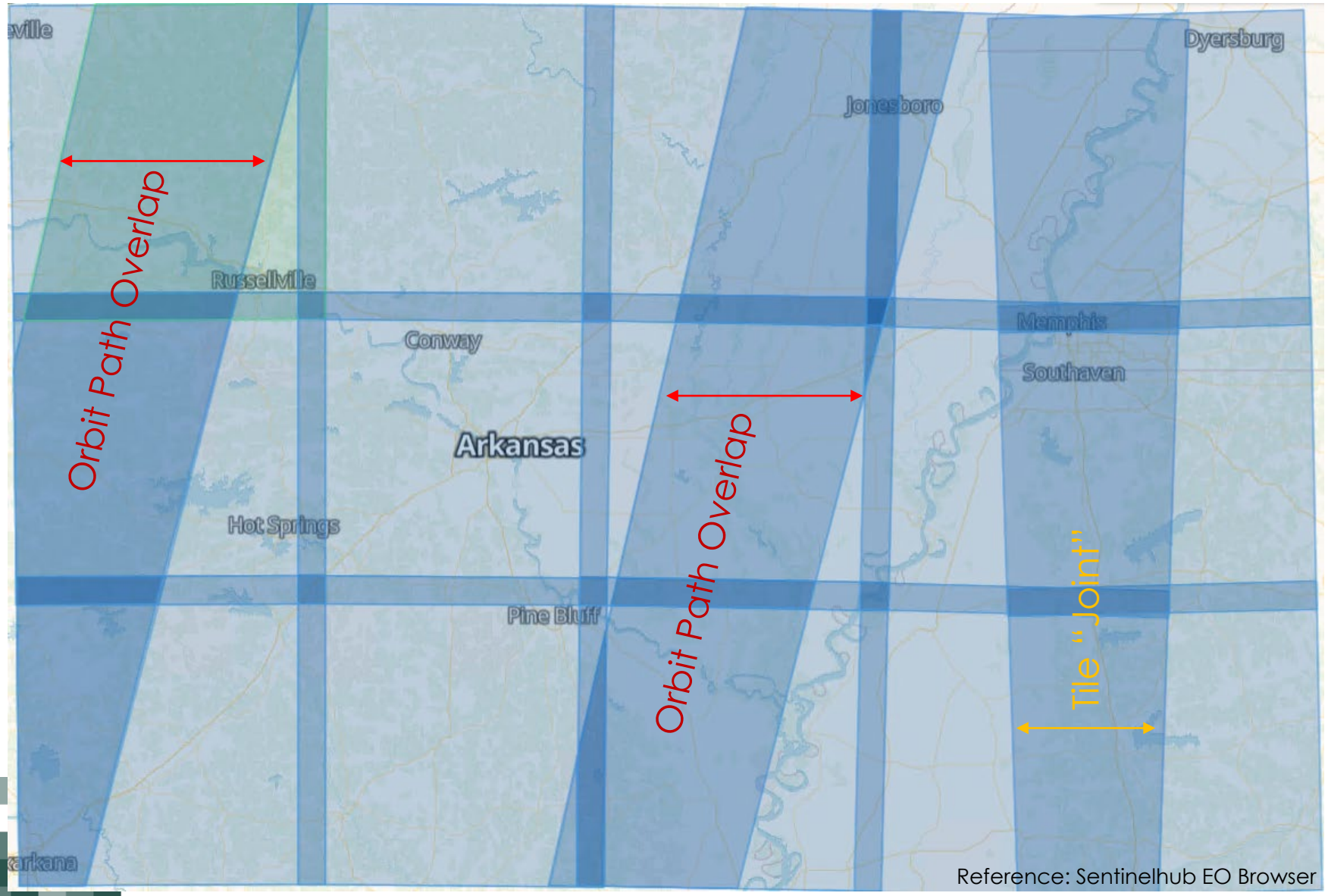


The tiles overlap and can result in values from the same scene in up to four different tiles. There also exists "joints" in the grid to tie it together due to overlaying a grid on a sphere.

For whatever reason, (perhaps slightly different referencing or calibration for each tile) ***band values for the same scene/location in different tiles can be slightly different ("false" differences).***

Reference: https://maps.eatlas.org.au/

Sentinel-2 Orbit Paths VS Tiling Grid



Reference: Sentinelhub EO Browser

Sentinel-2 scenes are much larger than tiles (swath width of 290km, vs 110km width of a tile). The overlap at the edges of orbit paths thus can cover an entire tile at certain latitudes.

Scene Classification Layer (SCL)

The SCL band is useful for rapid identification of data of interest. While not a proper land-cover classifier like the CDL, it facilitates rapid classification of per-scene pixels into 12 [mostly potentially transient] categories. Some highlights:

- Most common use case is identification of cloud cover, and there is a separate cloud mask available with probabilities (using the Sen2Cor algorithm). We also use for identifying vegetation in this demo
- 60m resolution, using single pixel from a single scene for prediction
- Can be error prone

Cloud and cirrus cloud detection rates and land, water, snow and shadow misclassification rates as clouds as determined using 108 Sentinel-2 scenes hand labeled by Hollstein et al.
Reference

Fraction of classifications as clouds

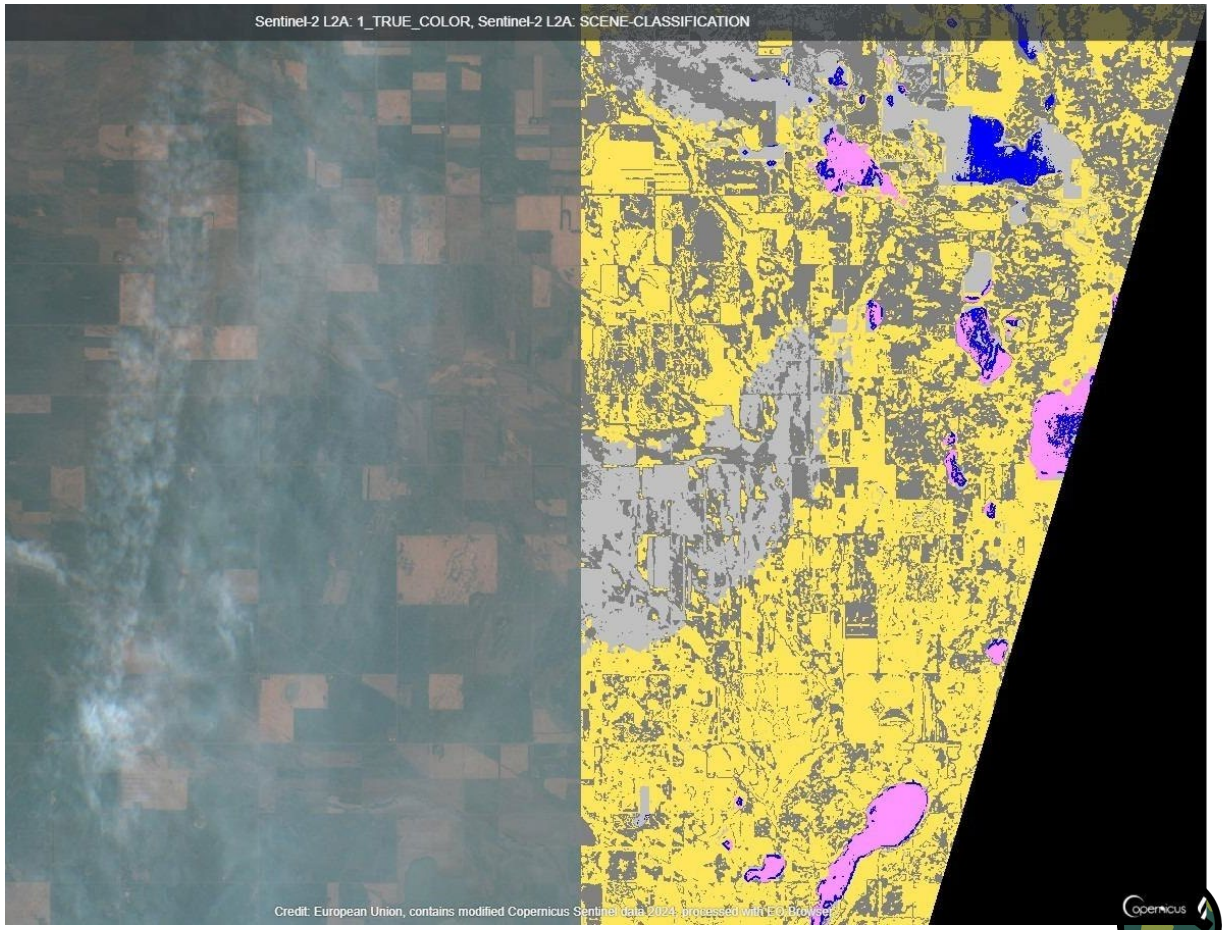
True Label \ Algorithms	Fmask	Sen2Cor	Sentinel Hub
Cloud	89.0%	97.5%	99.4%
Cirrus	88.3%	87.7%	83.8%
Land	7.2%	5.7%	2.2%
Water	2.0%	0.0%	0.1%
Snow	39.2%	30.7%	13.5%
Shadow	3.9%	3.9%	5.8%

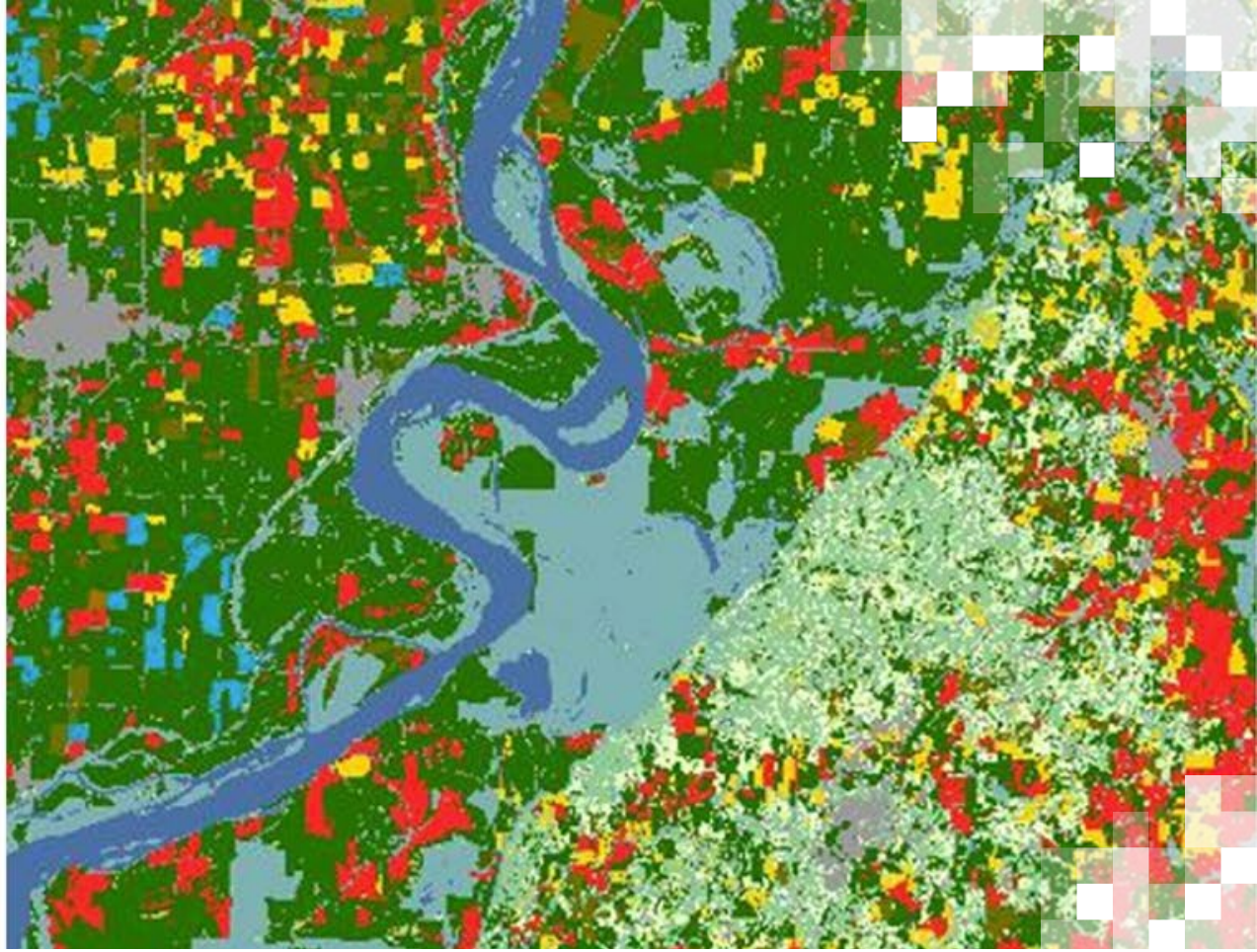
Common Issues/Limitations

Inconsistent geolocation/georeferencing of pixels and default scene classification labels from providers (e.g., SCL layer from Sentinel-2) aren't always accurate.

Inconsistent geolocation (10-15m disagreement between subsequent images 5 days apart)

Poor scene classification, mislabeling clouds as bare ground and water as snow due to cloud haze, etc.





Part 1 Section 3:

Databricks Procedural Demo (Run the code)

Databricks Community Edition Overview

Link to instructions to signup for Databricks Community Edition

- Jupyter Notebook style coding
- Databricks Community Edition allows up to 10GB persistent storage on the "FileStore"
  - Can store generic files, tables, and code.
  - Notebooks are stored in the "workspace" area
- Can spin up small instances with 2 cpus,15GB RAM, 130GB local storage, Spark enabled out of the box
- Anything stored on the local machine is lost when the instance shuts down
- Running notebook code longer than ~60 minutes will cause the node shutdown. However, as long as you are interacting with the notebook (writing and running code manually) it usually will stay up longer

Demo Information and Notes

Available materials for this demo:

- Three data processing scripts for Part-1 (CDL acquisition, Sentinel-2 acquisition, final data manipulation)
- Generalization: The CDL table could be any ground truth label + point location + timeframe, and the rest of the data acquisition & modeling can remain the same. E.g., crop health, vegetative stage, or other types of land cover classes.
- Any systematic error from the CDL will likely pass to models trained on it.
- To get data as it would exist after running the scripts in this training demonstration, download the zip files located with the other training materials.

How to download the resulting files from your Databricks account FileStore:

- Is somewhat not intuitive. To download, you should navigate to the path of **YOUR** file using the below format.
  - https://community.cloud.databricks.com/files/path/to/folder/filename.extension
    - 'path/to-folder' is the directory path where your file exists on the file store.

Code Steps

Strategy for processing and storing this data:

- Step 1: Define areas of interest (AOIs)
- Step 2: Acquire corresponding CDL data
- Step 3: Search and filter for available satellite data
- Step 4: Acquire corresponding satellite data
- Step 5: Rearrange Parquet file of satellite band values data into a single row per pixel location & season, with columns for time series components in the form of lists of values (e.g., band values, scene dates) to support modeling

APIs Brief Overview

In place of manually retrieving data, APIs make data acquisition & processing significantly more scalable by providing a consistent interface to search for and retrieve large amounts of data via web requests. We primarily rely on two APIs in this demo for data acquisition.

- CDL API from NASS geo data (link)
- AWS STAC API for sentinel-2 imagery searches.
  - Sentinel-2 image raster data can be downloaded via web URL download links that we can access directly once known from the imagery search. Since these are large though and slow down processing, we will do our best to minimize the downloading of any superfluous or low-impact scenes.

Area of Interest (AOI) & Boundary Creation

The only prior step needed typically before this part is to define AOIs. For this work we used the nassgeodata web gui to draw 7 boxes and export them as ESRI shapefiles. Then convert them to bounds (left, bottom, right, top) in the EPSG:5070 CRS (as required by the nassgeodata API). We provide bounds already in the CDL acquisition code.

Example python code to get bounds from [zipped] ESRI shapefiles from NASSGEO:

```
import  geopandas as gpd
from pyproj import CRS
import pandas as pd
root_path = 'C:/Users/myname/Downloads/'
# List of file paths for esri shapefile boundaries (exported into zips from nassgeo)
paths = [root_path + 'CDL_12345.zip', root_path + 'CDL_6789.zip']
gdf_list = [] # Create a list to hold the GeoPandas dataframes
# Read each shapefile into a GeoPandas dataframe and append it to the list
for path in paths:
    gdf = gpd.read_file("zip://" + path)
    gdf_list.append(gdf)
# Concatenate all the dataframes into a single GeoPandas dataframe
combined_gdf = gpd.GeoDataFrame(pd.concat(gdf_list, ignore_index=True))
target_crs = CRS("EPSG:5070") # Define the target CRS (EPSG:5070)
gdf_5070 = combined_gdf.to_crs(target_crs) # Convert the GeoDataFrame to the target CRS
print(gdf_5070.bounds.apply(lambda row: ', '.join(map(str, map(int, row))),
axis=1).to_string(index=False))
```

CDL Acquisition Code Summary

The results of this first part are a spatially down-sampled version of the CDL for the user specified AOIs and years.

- This code part executes quite rapidly (few minutes) and results in a parquet table with a single $30m^2$ pixel/year per row (with associated CDL estimate).
- The below table summarizes the top 5 CDL categories across all AOIs, and % of entire dataset per year that each CDL category represents. E.g., 2021 Soybeans represented ~36.6% of the land cover for that year.

year	CDL	count	total_count	percentage
2021	Soybeans	29416	80427	36.57
2020	Soybeans	28063	80151	35.01
2019	Soybeans	25422	80695	31.5
2019	Woody Wetlands	11916	80695	14.77
2020	Woody Wetlands	11725	80151	14.63
2021	Woody Wetlands	11765	80427	14.63
2020	Rice	10809	80151	13.49
2021	Corn	9331	80427	11.6
2021	Rice	8599	80427	10.69
2019	Cotton	8510	80695	10.55
2019	Rice	8360	80695	10.36
2019	Corn	7775	80695	9.64
2020	Corn	6930	80151	8.65
2020	Cotton	6907	80151	8.62
2021	Cotton	6847	80427	8.51

Sentinel-2 Acquisition Code Results

Summary: For each pixel/year from CDL acquisition code, this code acquires associated Sentinel-2 data for that entire year and saves in a Parquet table. ***Note that this part takes a long time to execute.*** It will time-out the free version of Databricks after an hour.

"Duplicate" data has the same date & tile, but perhaps due to updated processing has slightly different values. These are left in the data for this work and removed in data loader, but **probably best to only keep the one with largest number at the end of the tile (latest processing?)**. When duplicate values are due to tile overlap, choosing one randomly can be fine.

lon	lat	CDL	scl	coastal	blue	green	red	rededge1	rededge2	rededge3	nir	nir08	nir09	swir16	swir22	bbox	year	tile	scene_date
-90.6448304	36.46194644	Corn	5	371	604	802	1024	1266	1396	1536	1770	1764	1784	1988	1261	465073, 1479393, 583994, 1504168	2021	15SYA_1	1/5/2021
-90.6448304	36.46194644	Corn	5	114	422	678	951	1214	1344	1495	1744	1739	1770	2011	1267	465073, 1479393, 583994, 1504168	2021	15SYA_0	1/5/2021
-90.6448304	36.46194644	Corn	5	381	489	670	904	1037	1169	1245	1420	1444	1421	1864	1267	465073, 1479393, 583994, 1504168	2021	15SYA_1	1/13/2021
-90.6448304	36.46194644	Corn	5	230	364	568	855	997	1136	1221	1412	1436	1400	1883	1273	465073, 1479393, 583994, 1504168	2021	15SYA_0	1/13/2021
-90.6448304	36.46194644	Corn	3	422	483	438	493	542	567	628	696	695	633	855	680	465073, 1479393, 583994, 1504168	2021	15SYA_1	1/15/2021
-90.6448304	36.46194644	Corn	3	255	350	319	411	489	519	585	646	651	589	858	683	465073, 1479393, 583994, 1504168	2021	15SYA_0	1/15/2021
-90.6448304	36.46194644	Corn	5	570	782	1068	1376	1741	1881	2014	2092	2188	2177	3433	2999	465073, 1479393, 583994, 1504168	2021	15SYA_2	1/23/2021
-90.6448304	36.46194644	Corn	5	580	800	1100	1434	1758	1892	2024	2132	2197	2174	3466	3022	465073, 1479393, 583994, 1504168	2021	15SYA_0	1/23/2021
-90.6448304	36.46194644	Corn	5	1093	1023	1222	1584	1917	2068	2166	2220	2428	2476	2400	1853	465073, 1479393, 583994, 1504168	2021	15SYA_1	1/28/2021
-90.6448304	36.46194644	Corn	3	785	948	1138	1400	1676	1775	1936	2162	2134	1941	2729	1925	465073, 1479393, 583994, 1504168	2021	15SYA_0	2/2/2021
-90.6448304	36.46194644	Corn	5	539	668	847	1064	1343	1427	1553	1676	1780	1800	2412	1670	465073, 1479393, 583994, 1504168	2021	15SYA_1	2/24/2021
-90.6448304	36.46194644	Corn	5	476	623	827	1050	1327	1403	1526	1648	1753	1770	2410	1670	465073, 1479393, 583994, 1504168	2021	15SYA_0	2/24/2021
-90.6448304	36.46194644	Corn	5	565	706	933	1214	1449	1557	1678	1874	1863	1936	2656	1986	465073, 1479393, 583994, 1504168	2021	15SYA_1	3/4/2021
-90.6448304	36.46194644	Corn	5	577	718	946	1210	1453	1558	1684	1866	1862	1940	2658	1993	465073, 1479393, 583994, 1504168	2021	15SYA_0	3/4/2021

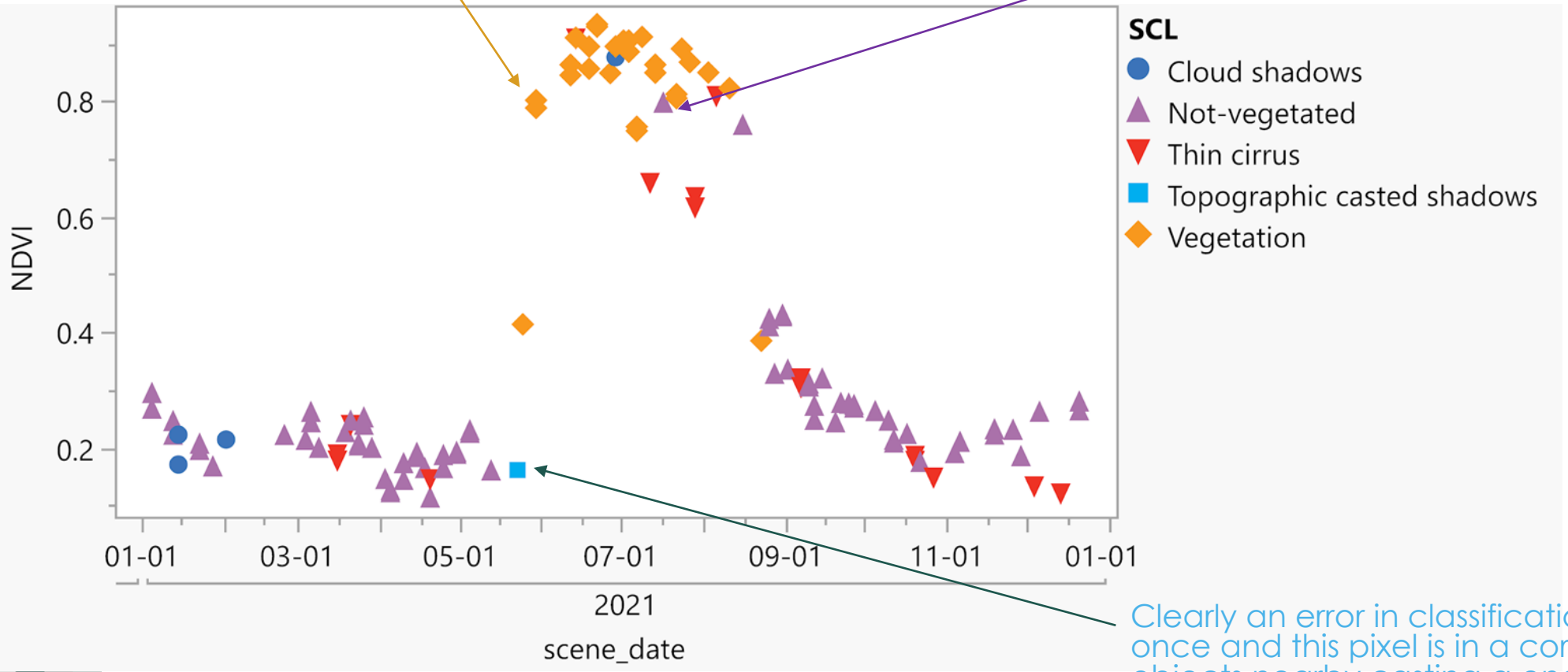
…123 rows total available for this particular pixel/year. Each row includes the band values for that location from a single scene/date.

Sentinel-2 Acquisition Code (Plotted)

Example data from a single pixel/year of corn. Includes duplicated data.

Example "duplicate" data (slightly different band values for same scene)

Clearly an error in classification (only happens once and this during middle of growing season).



Clearly an error in classification (only happens once and this pixel is in a corn field, so no objects nearby casting a one-time shadow).

Final Data Manipulation

A final rapid manipulation to the data combines all available scenes into a single row for each pixel/year.

- Several columns (bands, tiles, img dates, scl_vals) are all lists of values from the scenes for each row
- E.g., a pixel with 123 scenes would have 123*12 values in the list in the "bands" column.
- Lists of values are converted to binary strings for efficient storage (eliminating the "list" datatype, and commas for everything except the "tiles" column)

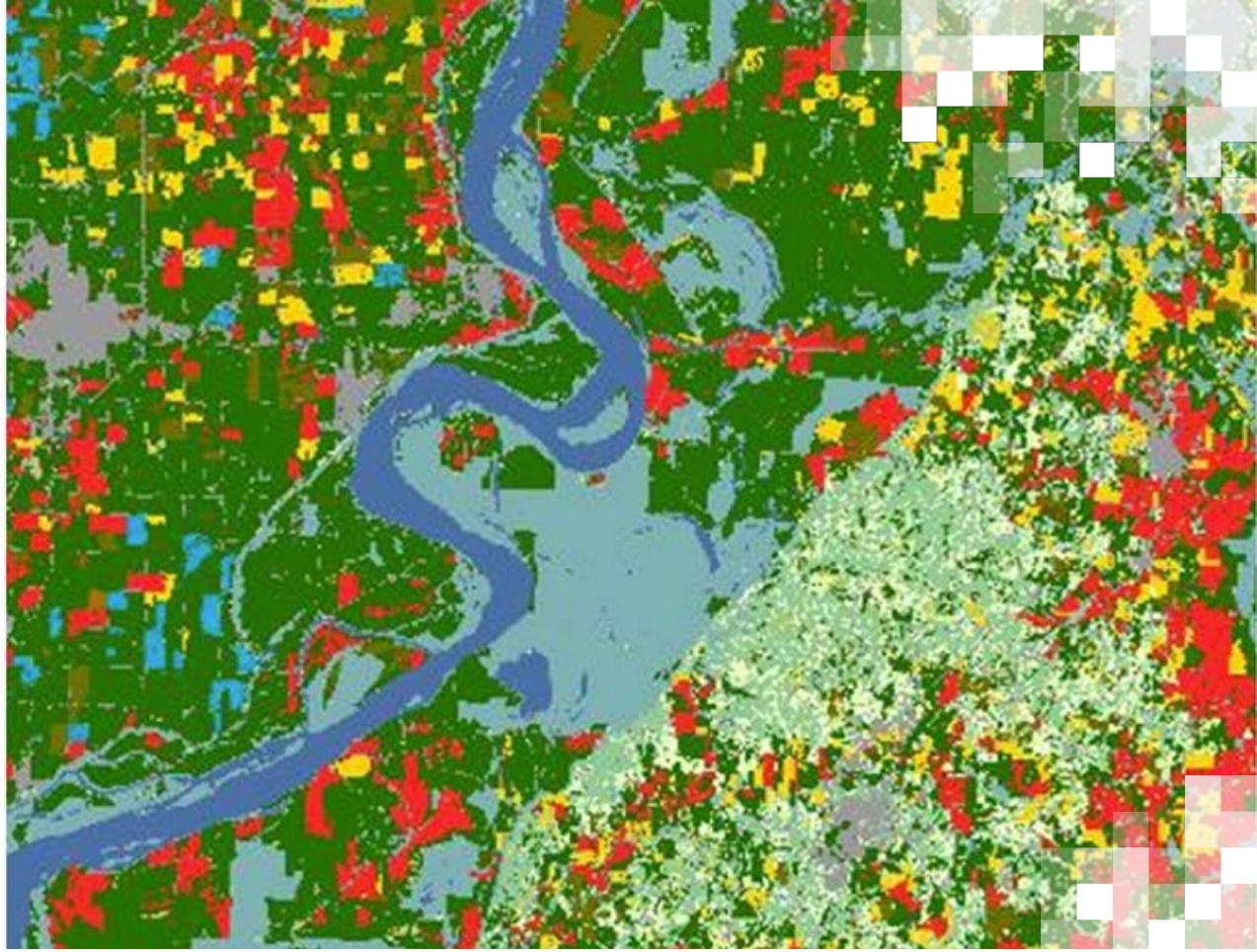
Final table

lon	lat	# scenes	bands	tiles	img dates	scl_vals	bbox	year	CDL
-89.2973	36.85473	38	AUECmAOEBM	MTZTQkZfMCw	Re5GAkYgRjRG	BQUKBQUFCgoF	549309,	2019	Corn
-89.3087	36.95602	71	AEYBMQHnAq4	MTZTQkZfMCw	Re5F7kYCRgJGI	BQUFBQUFBQUF	549309,	2019	Corn
-89.3206	36.88018	36	ATACXAOZBCw	MTZTQkZfMCw	Re5GAkYgRjRG	BQUKBQUFCgUF	549309,	2019	Corn
-89.3322	36.86472	33	ARcB4QKxA4M	MTZTQkZfMCw	Re5GIEY0RjIGP	BQoFBQUKBQUF	549309,	2019	Corn
-89.3382	36.79661	37	AQQBugJiA1ID	MTZTQkZfMCw	Re5GAkYgRjRG	AgIKBQUFCgoFB	549309,	2019	Corn
-89.3394	36.84096	37	ACgBcwHOArk	MTZTQkZfMCw	Re5GIEY0RjIGP	BQoFBQUKCgUF	549309,	2019	Corn
-89.3493	36.9019	37	AO4B5gJSAvgD	MTZTQkZfMCw	Re5GAkYgRjRG	BQUKBQUFCgUF	549309,	2019	Corn
-89.3552	36.95054	72	Ak8DYAQ4Bbg	MTZTQkZfMCw	Re5F7kYCRgJGI	BQUFBQUFBQUF	549309,	2019	Corn
-89.3577	36.74937	37	ANIAmQFTAfo	MTZTQkZfMCw	Re5GAkYgRjRG	BQUKBQUFCgoF	549309,	2019	Corn
-89.3632	36.97514	70	ACsBSwIVAIUC	MTZTQkZfMCw	Re5F7kYCRgJGI	BQUFBQUFBQUF	549309,	2019	Corn

Example decoded binary columns

decoded band vals	decoded tiles	decoded img dates	decoded scl_vals
321,664,900,1228,1415	16SBF_0,16SBF_0,16SE	2019-01-06,2019-01-26,201	5,5,10,5,5,5,10,10,
70,305,487,686,843,98	16SBF_0,16SBG_0,16S	2019-01-06,2019-01-06,201	5,5,5,5,5,5,5,5,5,5,
304,604,921,1068,1473	16SBF_0,16SBF_0,16SE	2019-01-06,2019-01-26,201	5,5,10,5,5,5,10,5,5
279,481,689,899,1075,	16SBF_0,16SBF_0,16SE	2019-01-06,2019-02-25,201	5,10,5,5,5,10,5,5,5
260,442,610,850,999,1	16SBF_0,16SBF_0,16SE	2019-01-06,2019-01-26,201	2,2,10,5,5,5,10,10,
40,371,462,697,744,81	16SBF_0,16SBF_0,16SE	2019-01-06,2019-02-25,201	5,10,5,5,5,10,10,5,
238,486,594,760,872,9	16SBF_0,16SBF_0,16SE	2019-01-06,2019-01-26,201	5,5,10,5,5,5,10,5,5
591,864,1080,1464,174	16SBF_0,16SBG_0,16S	2019-01-06,2019-01-06,201	5,5,5,5,5,5,5,5,5,5,
210,153,339,506,738,9	16SBF_0,16SBF_0,16SE	2019-01-06,2019-01-26,201	5,5,10,5,5,5,10,10,
43,331,533,597,755,11	16SBF_0,16SBG_0,16S	2019-01-06,2019-01-06,201	5,5,5,5,5,5,5,5,5,5,

Each row is a single 30m^2 CDL pixel from 1 year



Part 1:
Summary

Summary

- APIs allow us to automate and scale very large data processing pipelines in preparation for analysis and model building.
- Storing data in Parquet format and using Spark/Databricks to query/pivot or manipulate the data enables rapid investigation and transformation
  - The Parquet format has useful abstractions like partitions, which are also directories
- A convenient form for modeling time-series imagery data involves storing in parquet table format, with each row representing a pixel for a given time interval and having columns of:
  - Band values, scene dates, scene classification values over that time interval
  - Scalars for lat, lon representing the center point of the pixel (could substitute an Uber H3 hex or Google S2 cell instead)
  - A prediction target (ground truth)

Looking Ahead to Part 2

- Process data to prepare for model training using TensorFlow
- Properly split the data into train/val/test splits to avoid “data-leakage”
- Convert the irregularly-spaced time-series imagery into bucketed time-series to prepare for model training
- Modify the CDL labels to align with our training goals

Homework and Certificates

- **Homework:**
  - One homework assignment
  - Opens on March 19
  - Access from the training webpage
  - Answers must be submitted via Google Forms
  - **Due by April 1**

- **Certificate of Completion:**
  - Attend all three live webinars (attendance is recorded automatically)
  - Complete the homework assignment by the deadline
  - You will receive a certificate via email approximately two months after completion of the course.

Contact Information

Trainers:

- John Just (John Deere)
  - JustJohnP@JohnDeere.com
- Erik Sorensen
  - SorensenErik@JohnDeere.com
- Sean McCartney
  - Sean.McCartney@nasa.gov

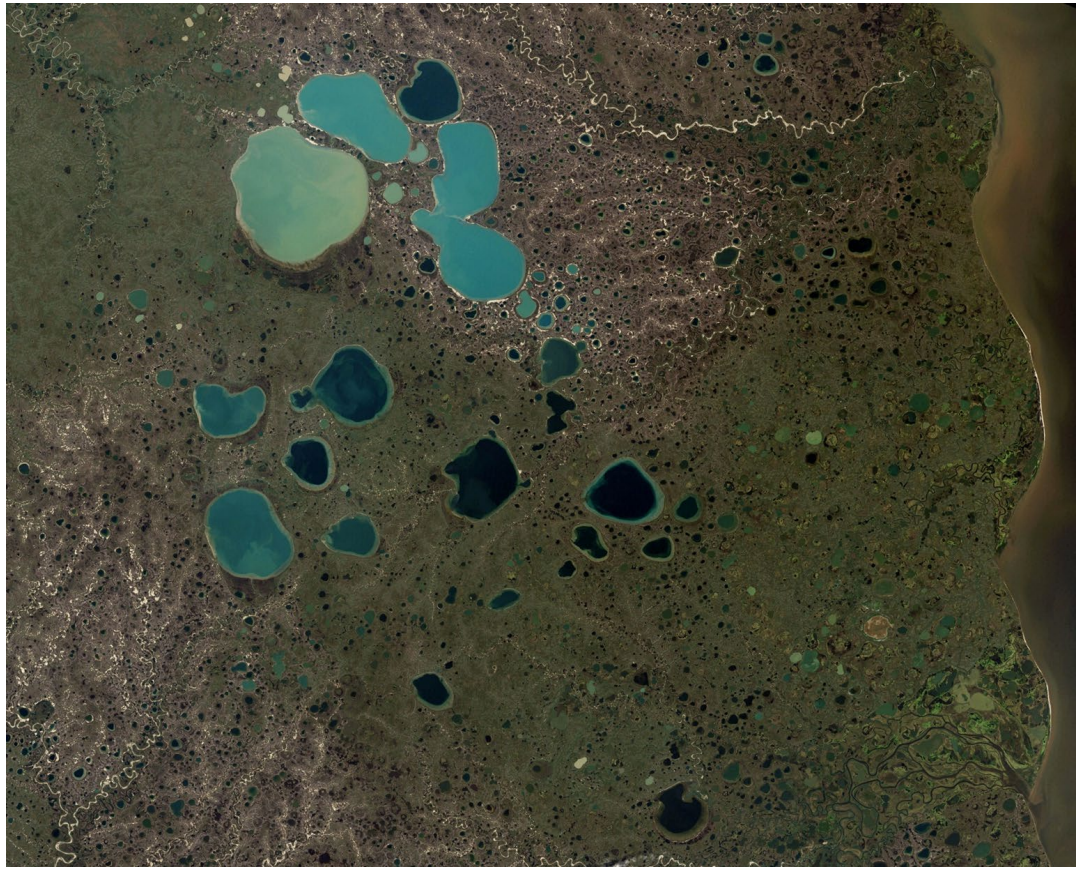
- ARSET Website
- Follow us on X (formerly Twitter)!
  - @NASAARSET
- ARSET YouTube

Visit our Sister Programs:

- DEVELOP
- SERVIR

Questions?

- Please enter your questions in the Q&A box. We will answer them in the order they were received.
- We will post the Q&A to the training website following the conclusion of the webinar.



https://earthobservatory.nasa.gov/images/6034/pothole-lakes-in-siberia


APPLIED REMOTE SENSING TRAINING PROGRAM
TRAIN - EMPOWER - ADVANCE

Thank You!