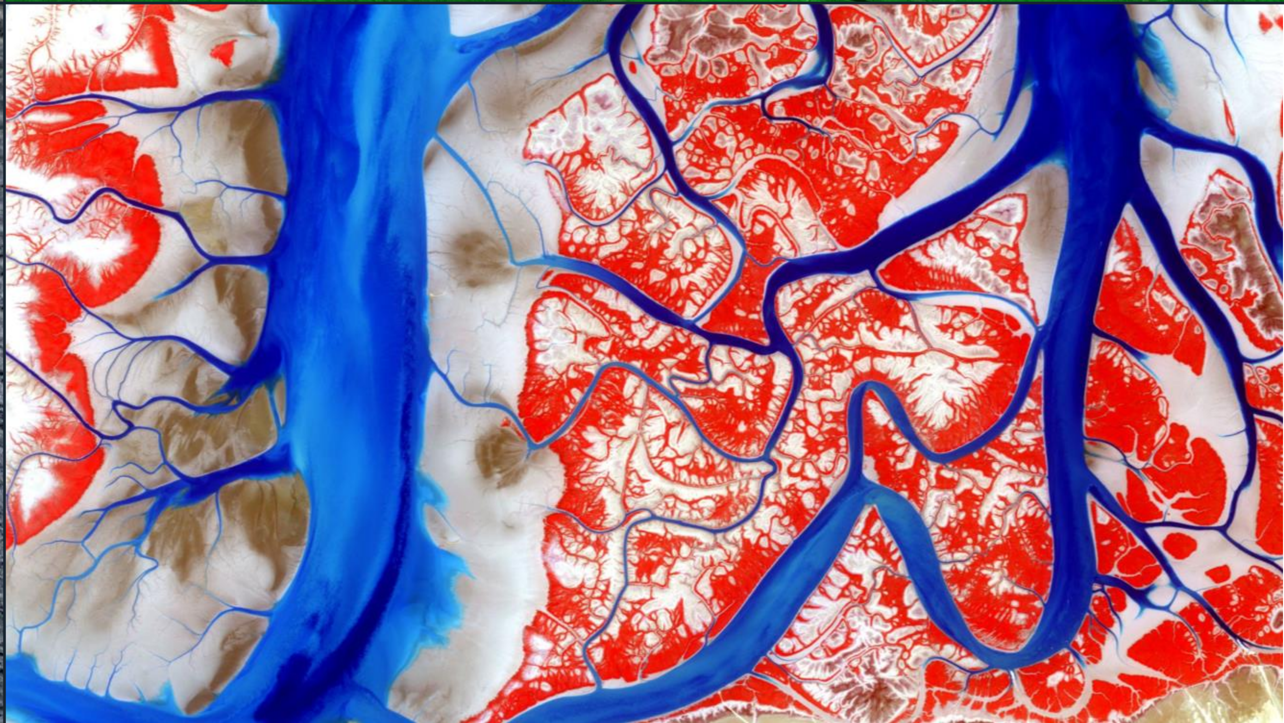
L3HARRIS™

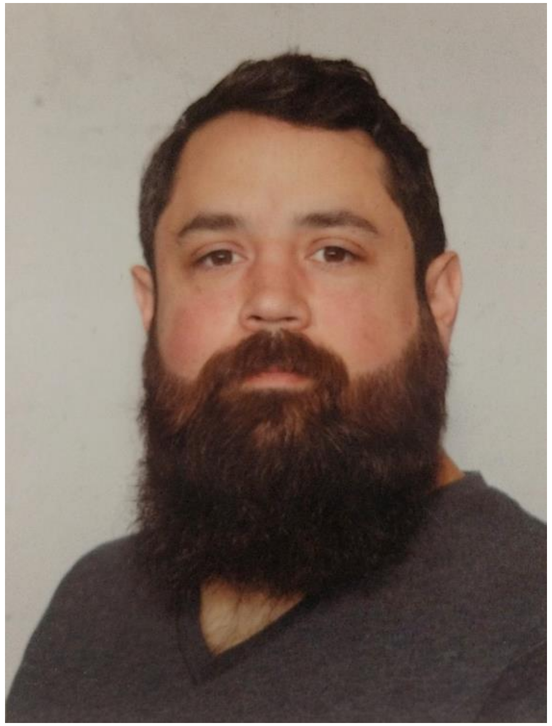
# TAKING ENVI TO THE NEXT LEVEL VIA MACHINE LEARNING

October 2022


JP Metcalf – Solutions Engineer
Zach Norman – Product Manager

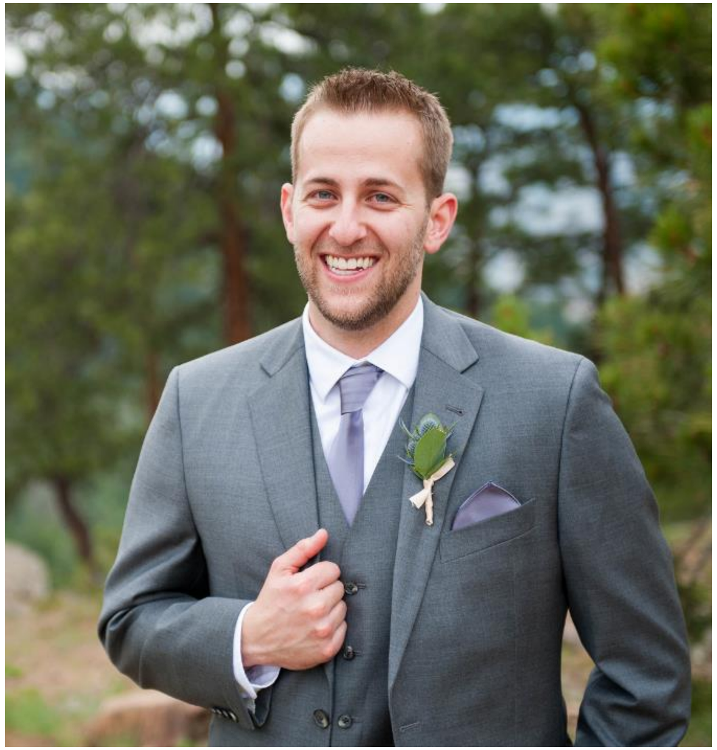
## Contact Information and Introductions




**JP Metcalf**

*Solutions Engineer*

jp.metcalf@l3harris.com


**Zachary Norman**

*Product Manager*

zachary.norman@l3harris.com

# Agenda

- Introduction
- Machine Learning vs Deep Learning: Installation and Hardware
- Accessing the Machine Learning Algorithms
- High-level Features and Use-cases
- Machine Learning Workflow in ENVI
- Machine Learning and Hyperspectral Data
- Machine Learning vs Deep Learning
- Traditional Approaches vs Machine Learning
- Use Cases
- Questions and Discussion

# Introduction: What is Machine Learning?

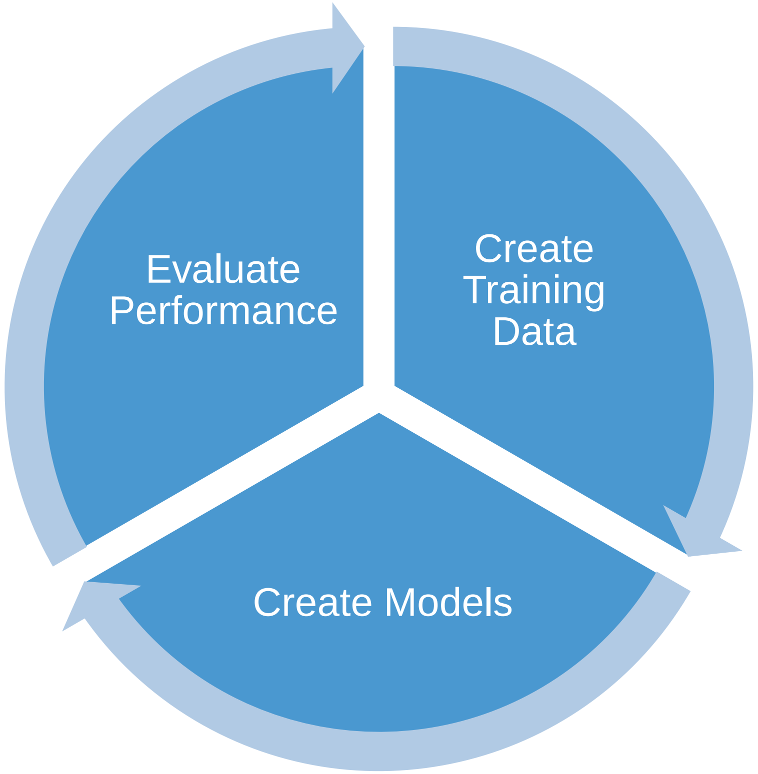
From Wikipedia:

- Machine learning (ML) is the scientific study of algorithms and statistical models that computer systems use to effectively perform a specific task without using explicit instructions, relying on patterns and inference instead.

It's not terminator

# Context: The ENVI Deep Learning Module

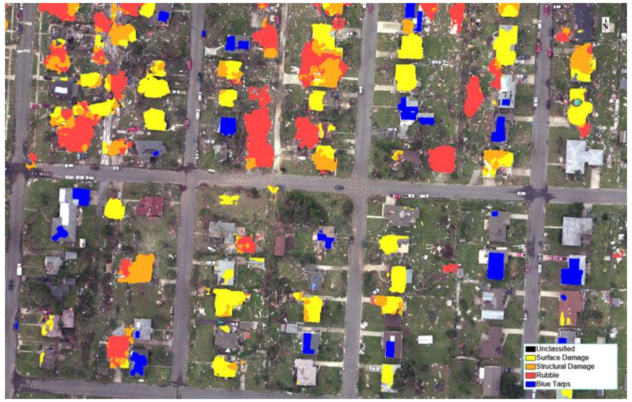
Applied deep learning for geospatial imagery in ENVI, the leading remote sensing and image analysis software



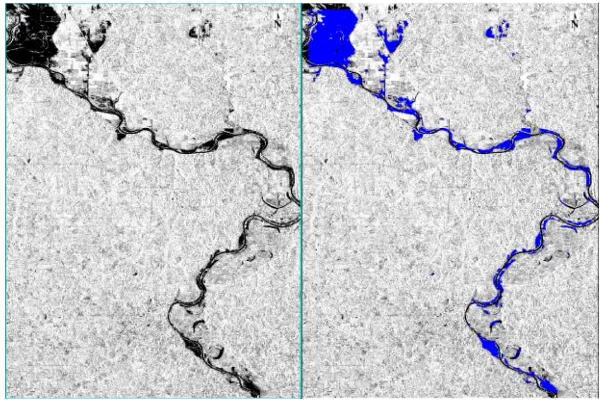

Deep learning workflow in ENVI, built on TensorFlow and Keras

**Without** needing to program, the capabilities include:

- Segmentation (i.e. cloud masking)
- Object detection (i.e. cars or ships)
- Linear feature extraction (i.e. roads)
- Support for nearly any image format and data modality



Assess building damage after hurricanes and tornadoes

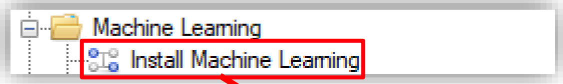


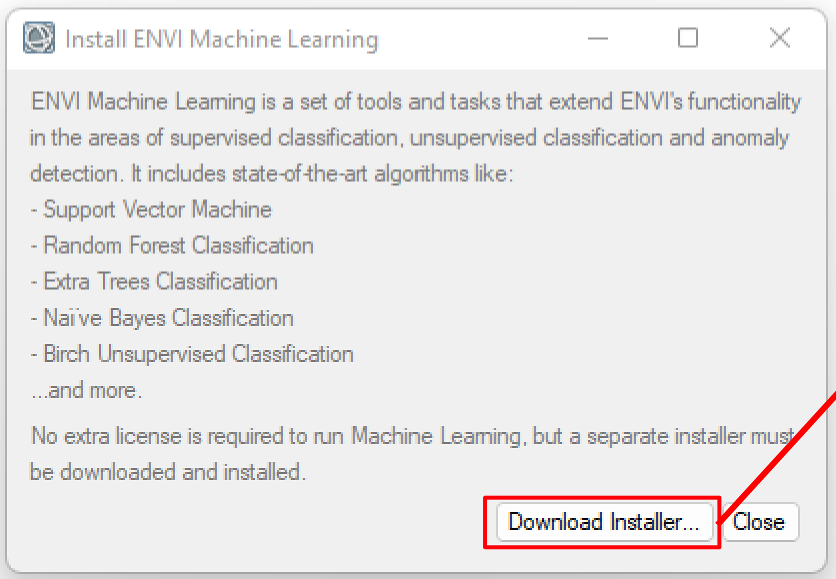
Automated flood detection using SAR

# Machine Learning vs Deep Learning: Installation and Hardware

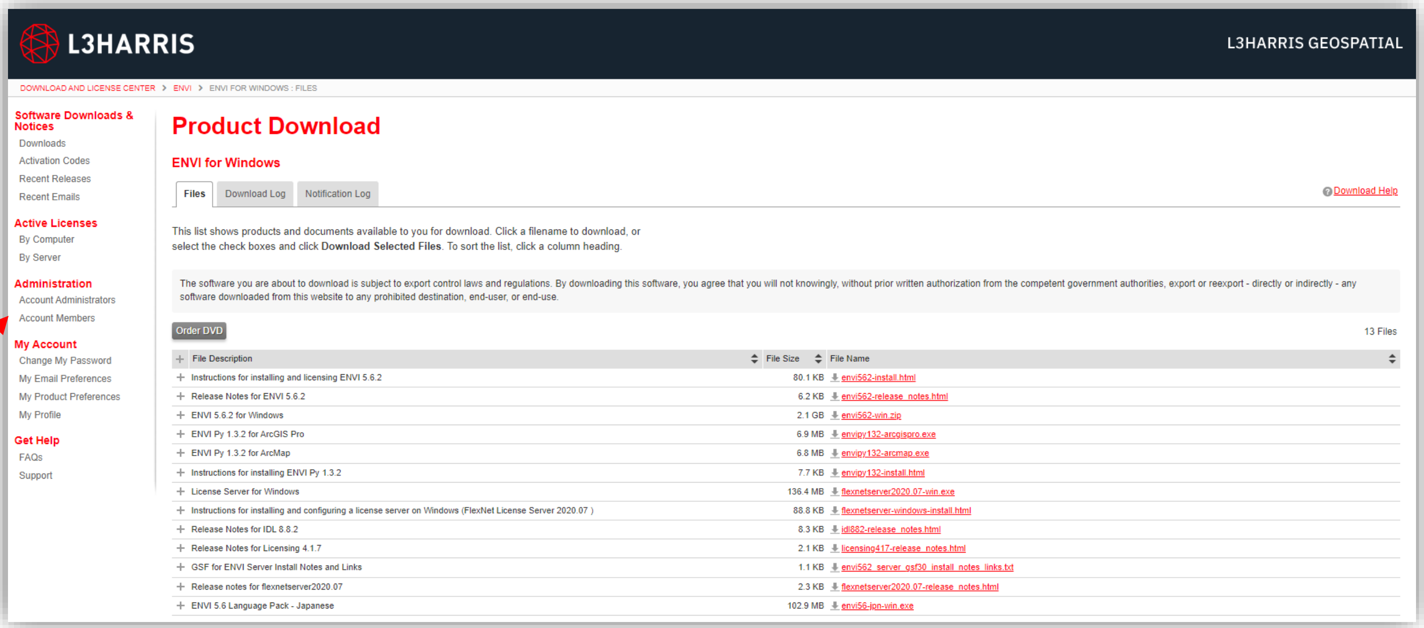
| MACHINE LEARNING | DEEP LEARNING |
|---|---|
| • FREE and unlicensed, packaged with “ENVI Deep Learning”<br>• Available for ENVI 5.6.3<br>• Uses the CPU<br>• Recommended 16 GB RAM<br>• Libraries optimized for Intel, but AMD also supported<br>– Some algorithms have significant performance improvements with Intel | • Need to purchase the “ENVI Deep Learning” module to use<br>• Latest version with be 2.0 and will have machine learning<br>• Requires a GPU<br>• Minimum 8 GB GPU RAM<br>• Only NVIDIA GPUs are supported<br>– TensorFlow requirement |


# Machine Learning: How to Install




Signing into the Download & Licensing portal will show the ENVI Deep Learning 2.0 installer under the ENVI 5.6.3 download



# Machine Learning: Algorithm Types

## SUPERVISED

- Description
  - Similar to traditional classification algorithms in ENVI that produce a raster with a color table and class lookup
  - Uses one or more rasters and one or more ROIs per raster
- Algorithms by recommended use
  - ★★★ Random Forest
  - ★★★ Extra Trees
  - ★★☆ Naïve Bayes
  - ★★☆ Nearest Neighbors (KNeighbors)
  - ★☆☆ SVM (Linear, RBF)
    - Not recommended to use, better and faster alternatives
    - *For good user experience with RBF SVM, requires intel CPU*

## ANOMALY DETECTION

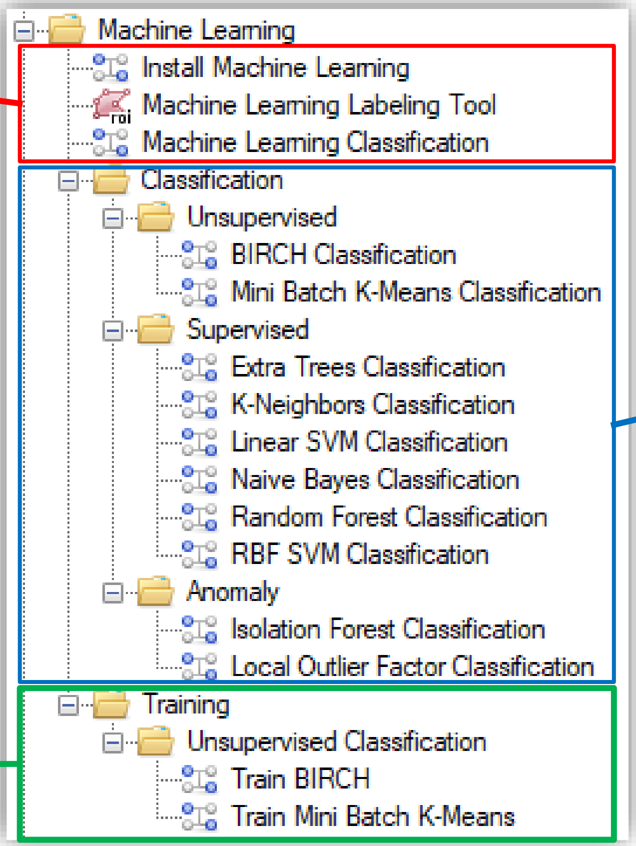
- Description
  - NEW algorithms that, based on examples of non-anomalous spectra, give you a binary 1/0 flag if a pixel is an anomaly and different from what it was trained on
  - Uses one or more rasters with example spectra from each raster that represent normal pixels
- Algorithms
  - Isolation Forest
  - Local Outlier Factor
- Both algorithms have similar performance

## UNSUPERVISED

- Description
  - Similar to ISODataClassification, generates a classifier that finds “X” number of classes
  - Uses one or more rasters without any training data as input (i.e. no ROIs/labels need to be specified)
- Algorithms by recommended use
  - ★★★ KMeans
  - ★★☆ BIRCH

# Machine Learning Toolbox

Quick access tools



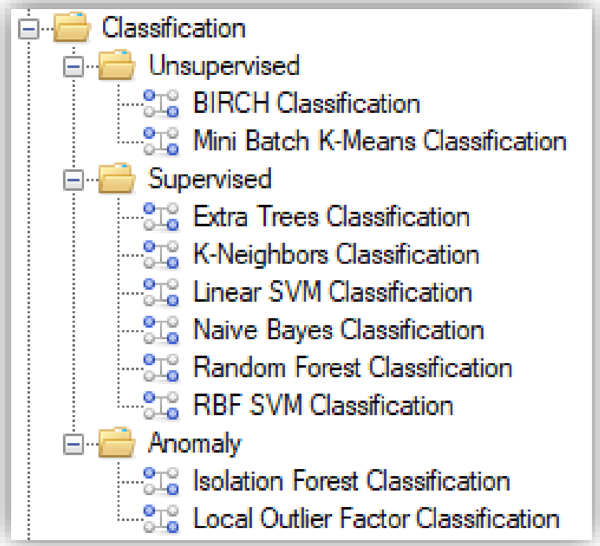

"One shot" classification tools

Limited direct training options that don't require labels

# Tools by Use Case

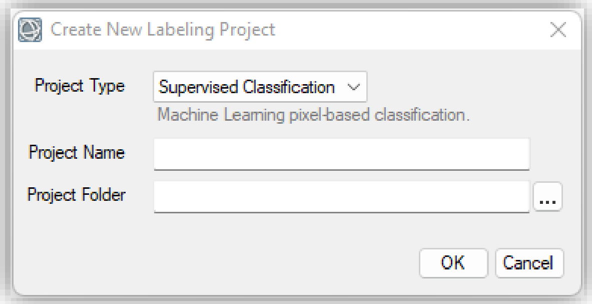
## EASY-PEASY

- Toolbox entries to “find more pixels like this” using spectral information
  - Input: Raster + ROIs
  - Output: Classification Image



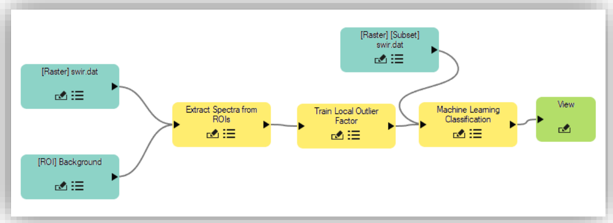

## HAND HOLDING

- Labeling tool to manage your data and automate machine learning classifier generation with the click of a button
  - Input: Raster (optional) ROIs
  - Output: Machine learning model
- Two new project types
  - Supervised Classification (2+ classes)
  - Anomaly Detection (single class)




## FREE FORM

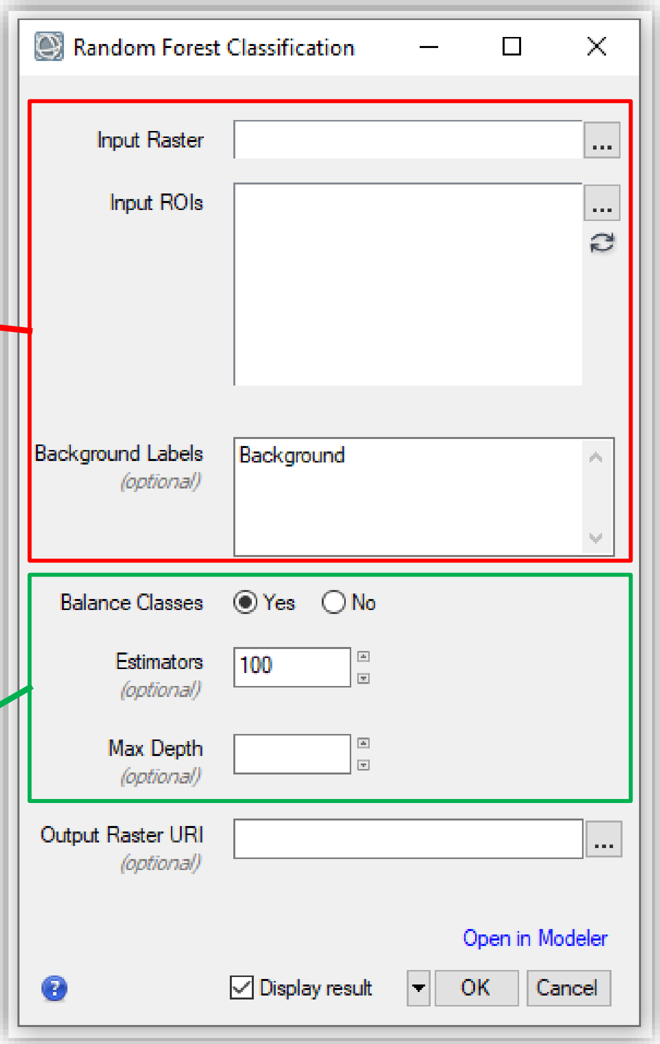
- For advanced users and custom use cases, the ENVI Modeler and ENVI API help you automate and create your own machine learning workflow




- Fun fact: Zach likes to build “machine learning spaceships” using the ENVI Modeler

# Easy-peasy Classification Tools

Data inputs, differ slightly by algorithm type (i.e. Anomaly Detection, Supervised, Unsupervised)

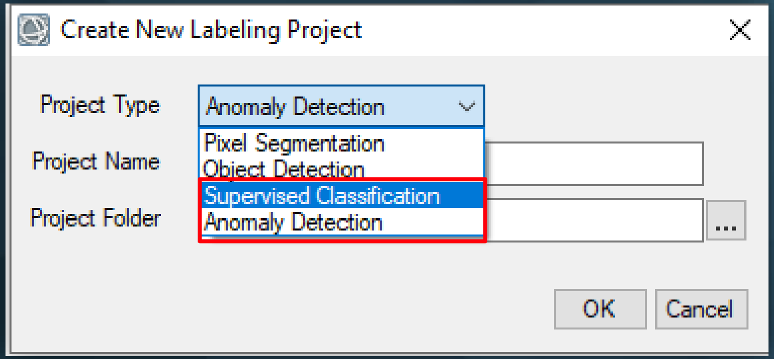



Custom parameters for each type of model

Goal: Easy-to-use tools when you are working with a single image

Performs: data preparation, training data extraction, training, and classification

# Labeling Tool: Updated for Machine Learning

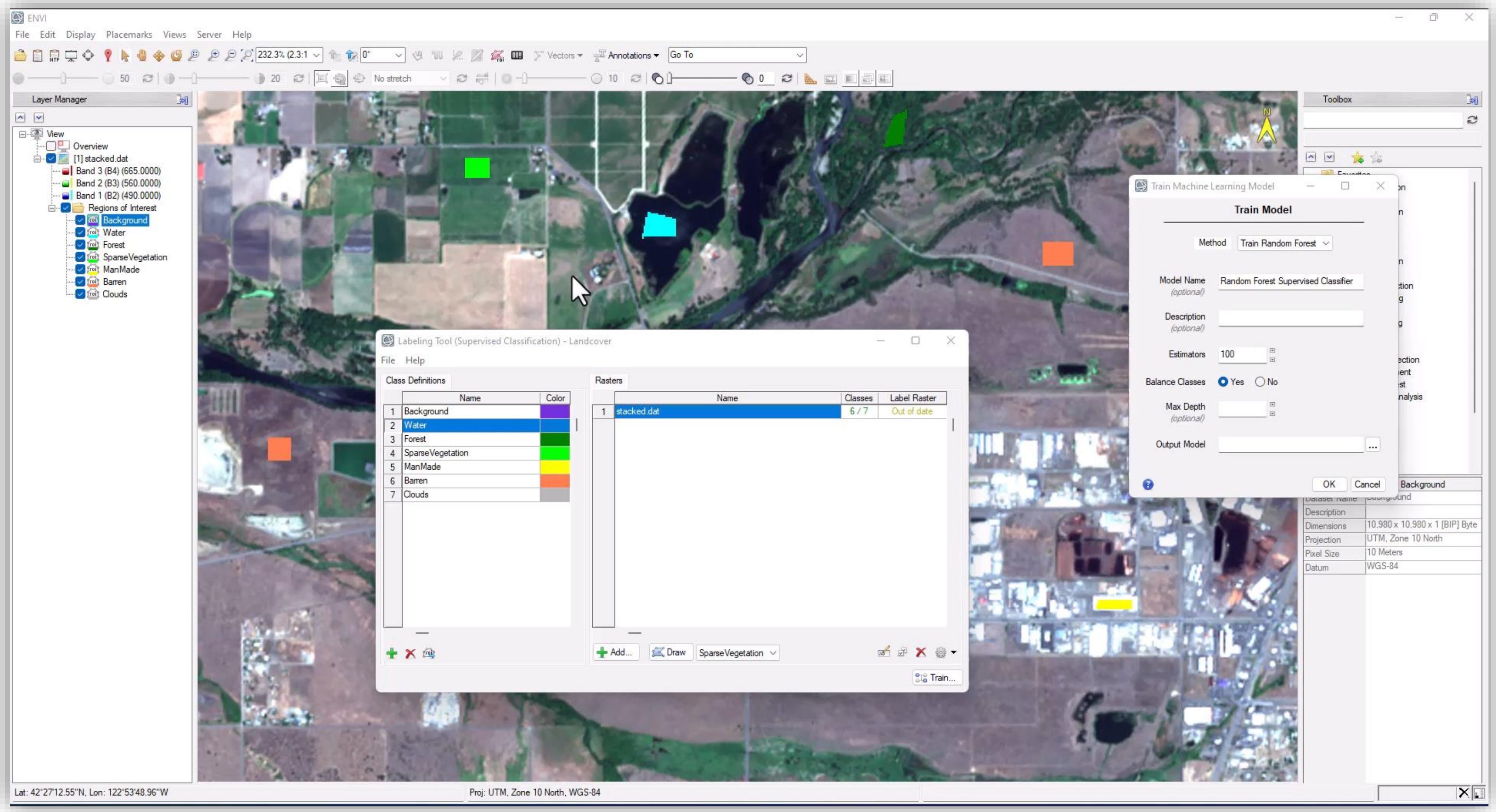



When you create a new project using the Deep Learning Labeling Tool, there are two new options added for machine learning.

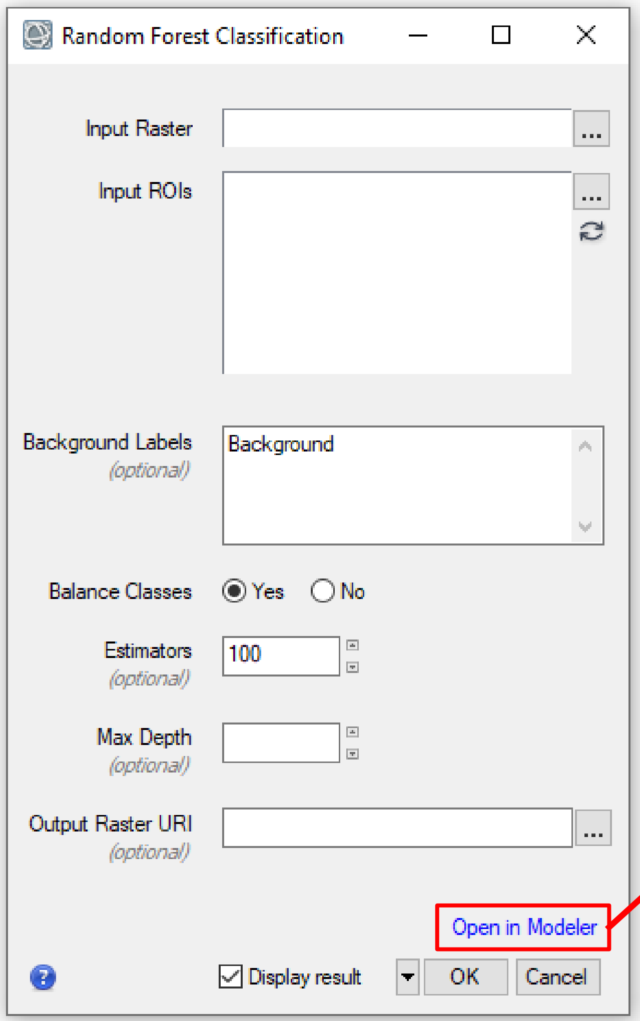
Supervised Classification: Label pixels to find, by class, and specify background data

Anomaly Detection: Label non-anomalous pixels

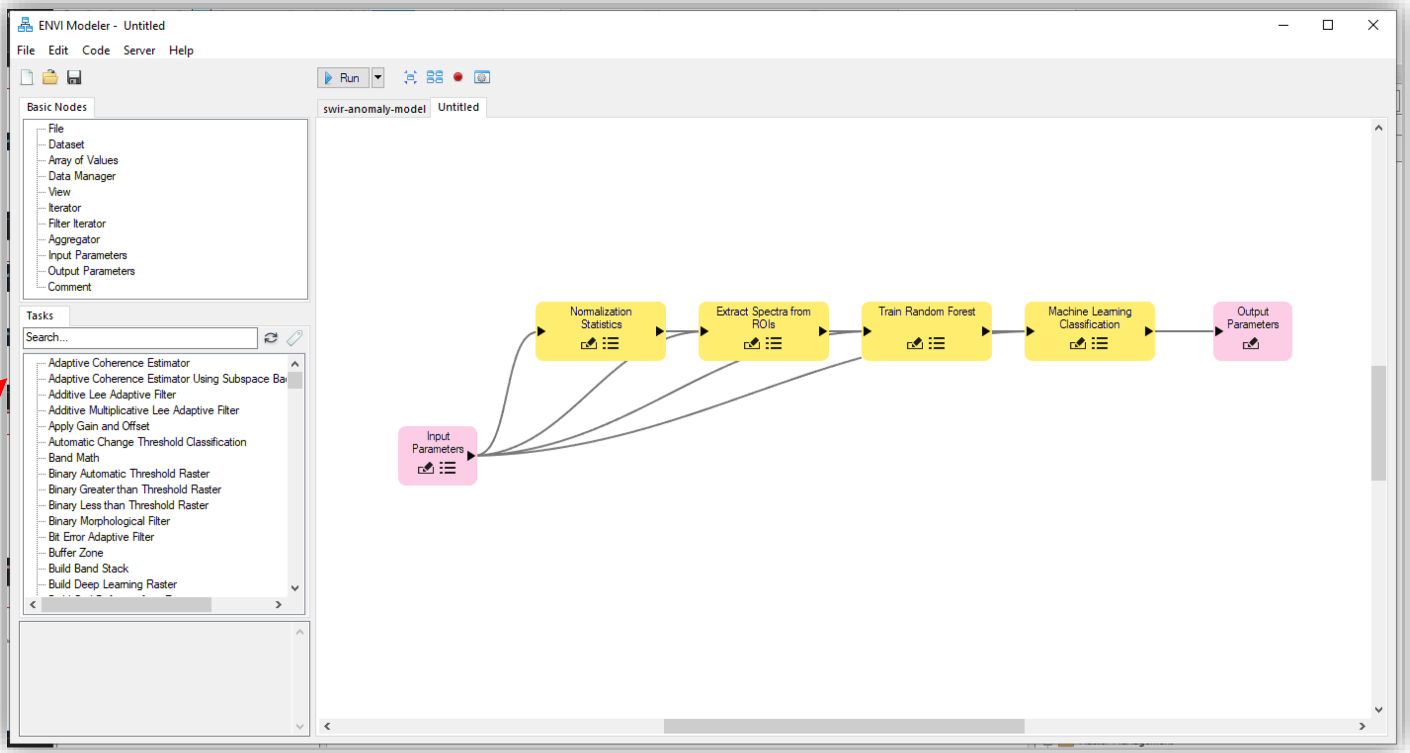
A short video showing the training dialog in the labeling tool allowing you to select any supervised classification algorithm for landcover classification



# Classification Tools: Bonus ENVI Modeler Workflows




Helps make the Machine Learning algorithms more accessible to ENVI Modeler and ENVI API users



# Scenario: Active Volcano!

RGB

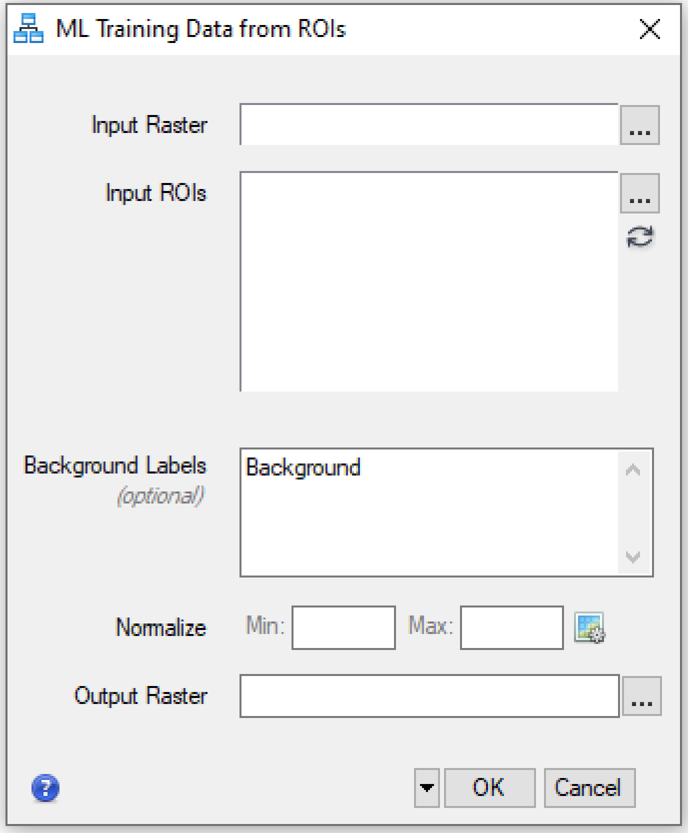
NIR2, NIR1, RedEdge

Data credits:
WorldView 8 band

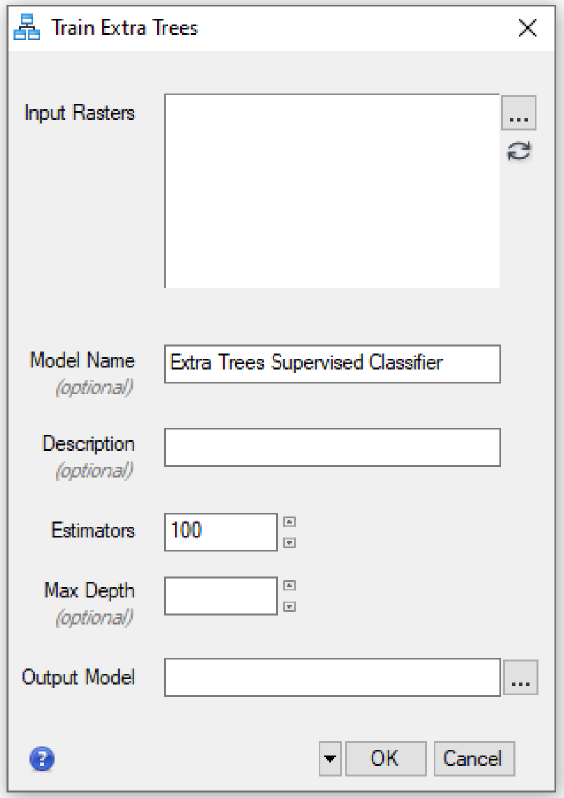
# Step-by-Step: Machine Learning Workflow

Goal: Show how to use the machine learning tools to detect hot lava, cool lava, and gas plumes

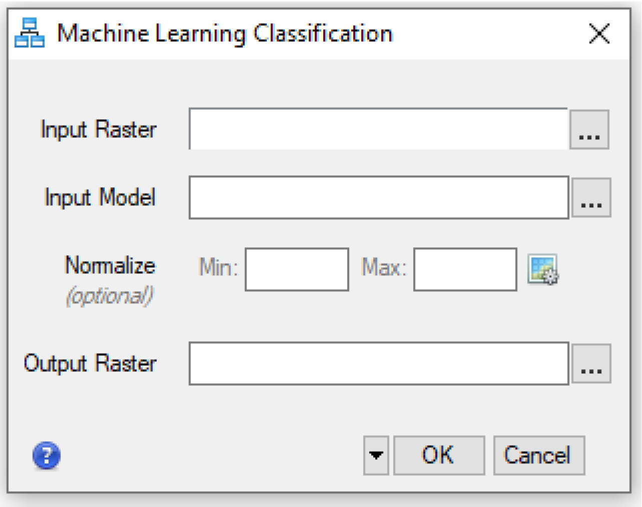
**Training Data**




**Generate Classifier**

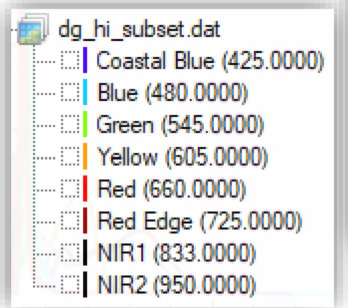



**Classify Raster**



# Data Preparation for Machine Learning

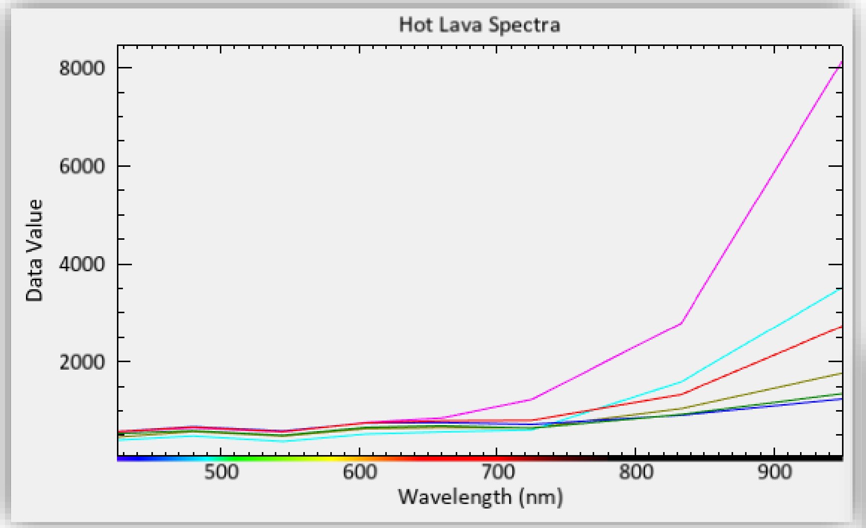
- Assumptions about data for machine learning
  - Your data is analysis ready with pixel values that represent surface reflectance
  - As surface reflectance, your data falls between two values (likely 0 and 10,000 but could be different)
  - If you use more than one image, then the datasets need to be apples-to-apples and should both represent surface reflectance



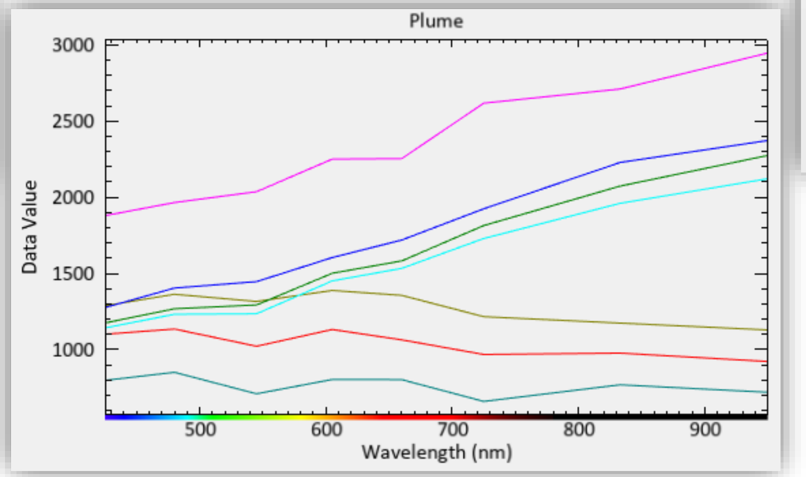

Full WorldView scene

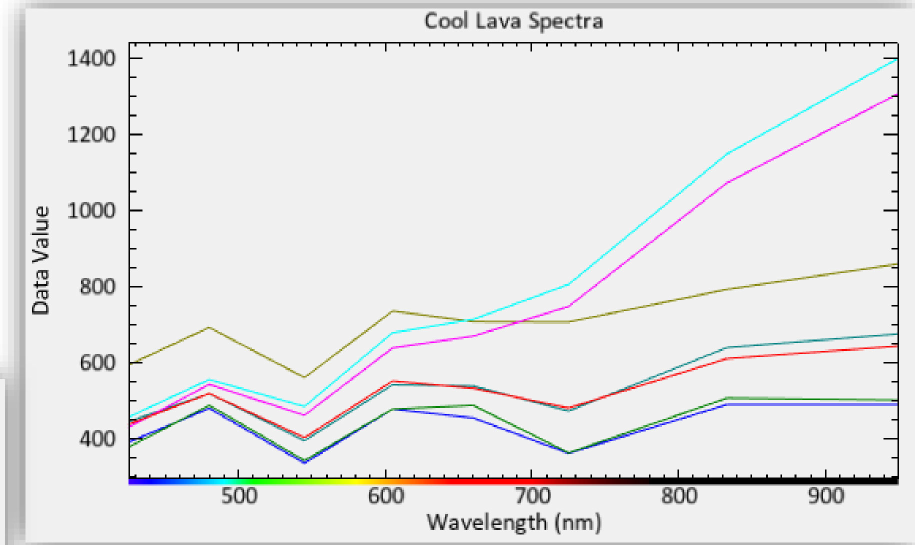
# Training Data for Machine Learning

- We use all the bands for a given pixel as training data with machine learning and there is no spatial component to it
- Whereas deep learning uses a byte-scaled/stretched version of 3-8 bands and mostly spatial context




Sample spectral for our three classes that we want to identify



# Fundamental Concept: Background Data

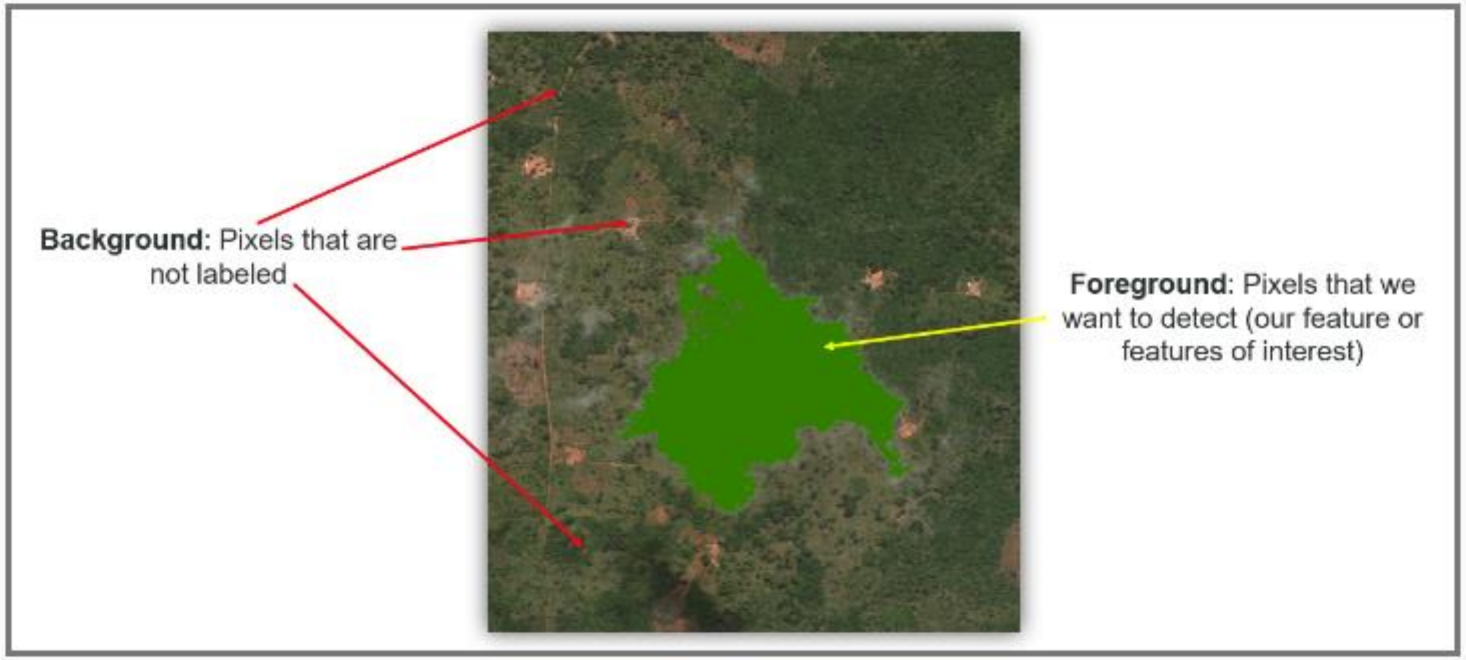
The biggest difference between machine learning and deep learning is that, for machine learning, you need to specify the pixels that you *don’t* want to classify.

Think about this as a generic “Background” class with pixels you don’t care about.

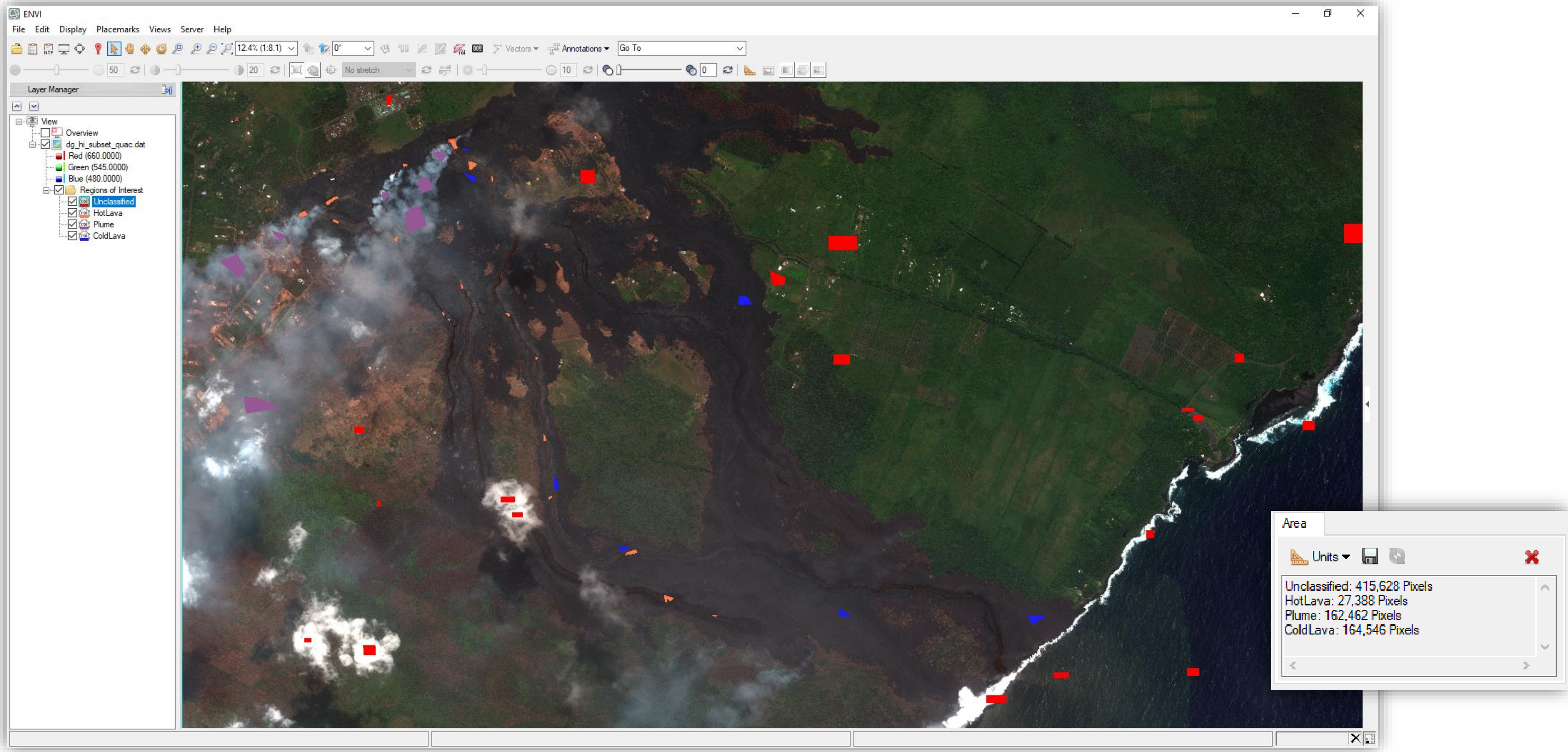
This is because machine learning, like deep learning, must assign a value/class to a pixel.

UNets or object detectors do not require you to label background data because anything not labeled is already considered that.

**Figure 5: Anything not labeled for a UNet is automatically considered a background pixel**

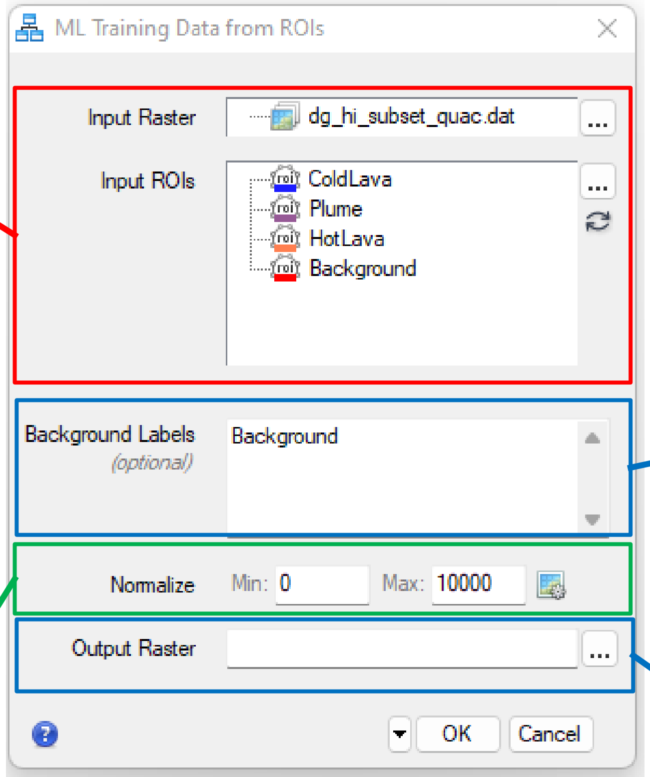


# Volcano Training Data



# Back to Extracting Spectra from ROI

The rasters and ROIs to extract sample spectra from




Which ROIs contain examples that represent background pixels or features we don't want to identify

Data ranges between 0 and 10,000 from QUAC.

Select the button on the right to calculate this for you. You can manually calculate in the ENVI Modeler as well

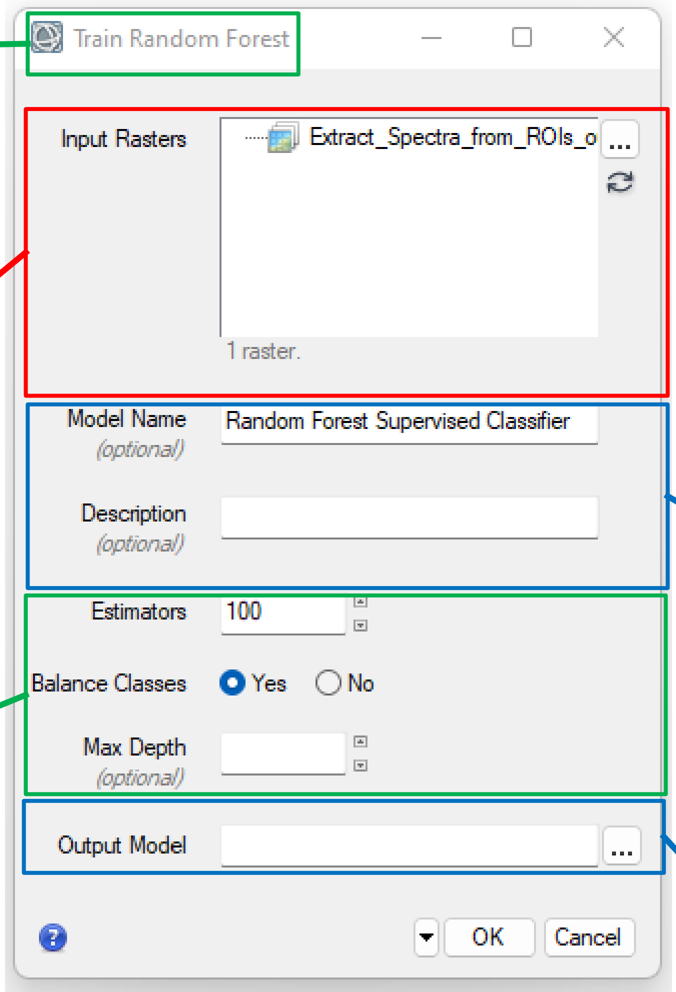
Special ENVI Rasters are used to store training data

# Step Two: Train

Title indicates the algorithm that we are generating a model for

Output rasters from “Extract Spectra from ROI”

Model-specific parameters, varies by the model we are training

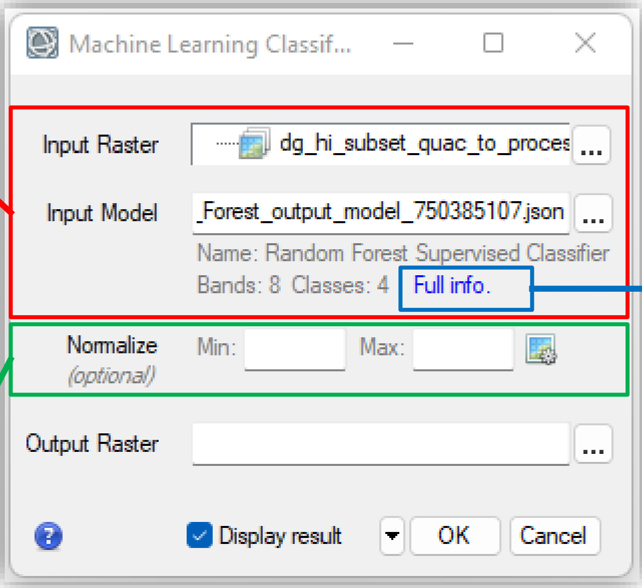



Customize the name and description of the model being generated

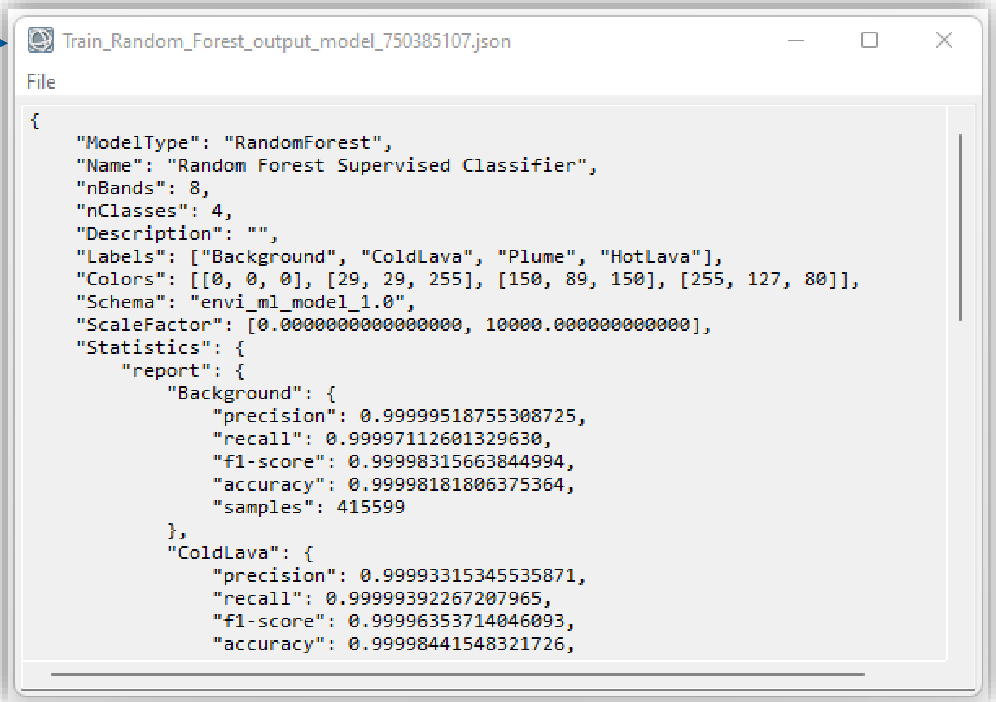
Output format is a JSON file that contains model, more details on training slide

# Step Three: Classify

The raster to process and previously generated model file

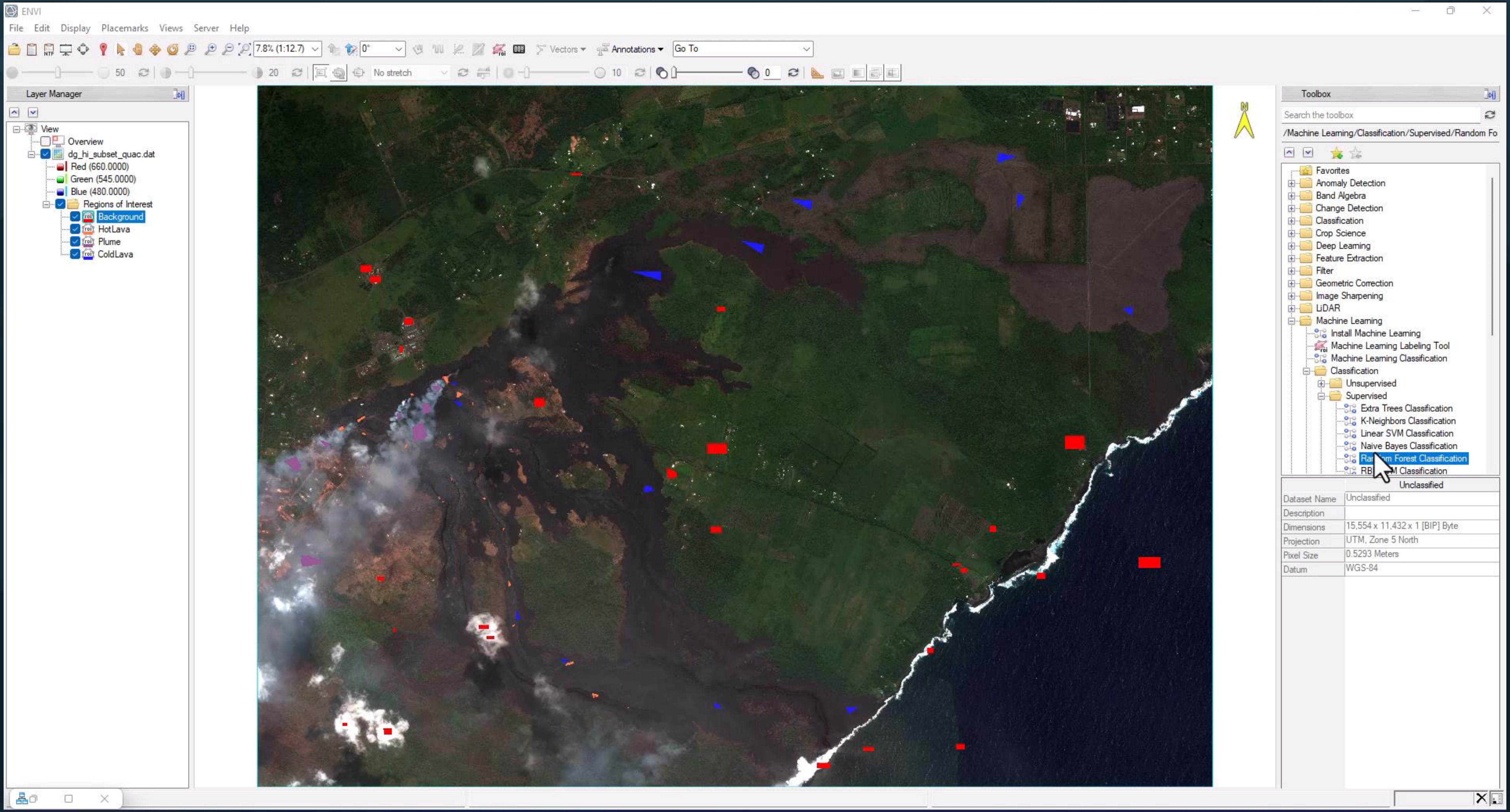



Machine learning models are stored as JSON files and contain the metadata and model

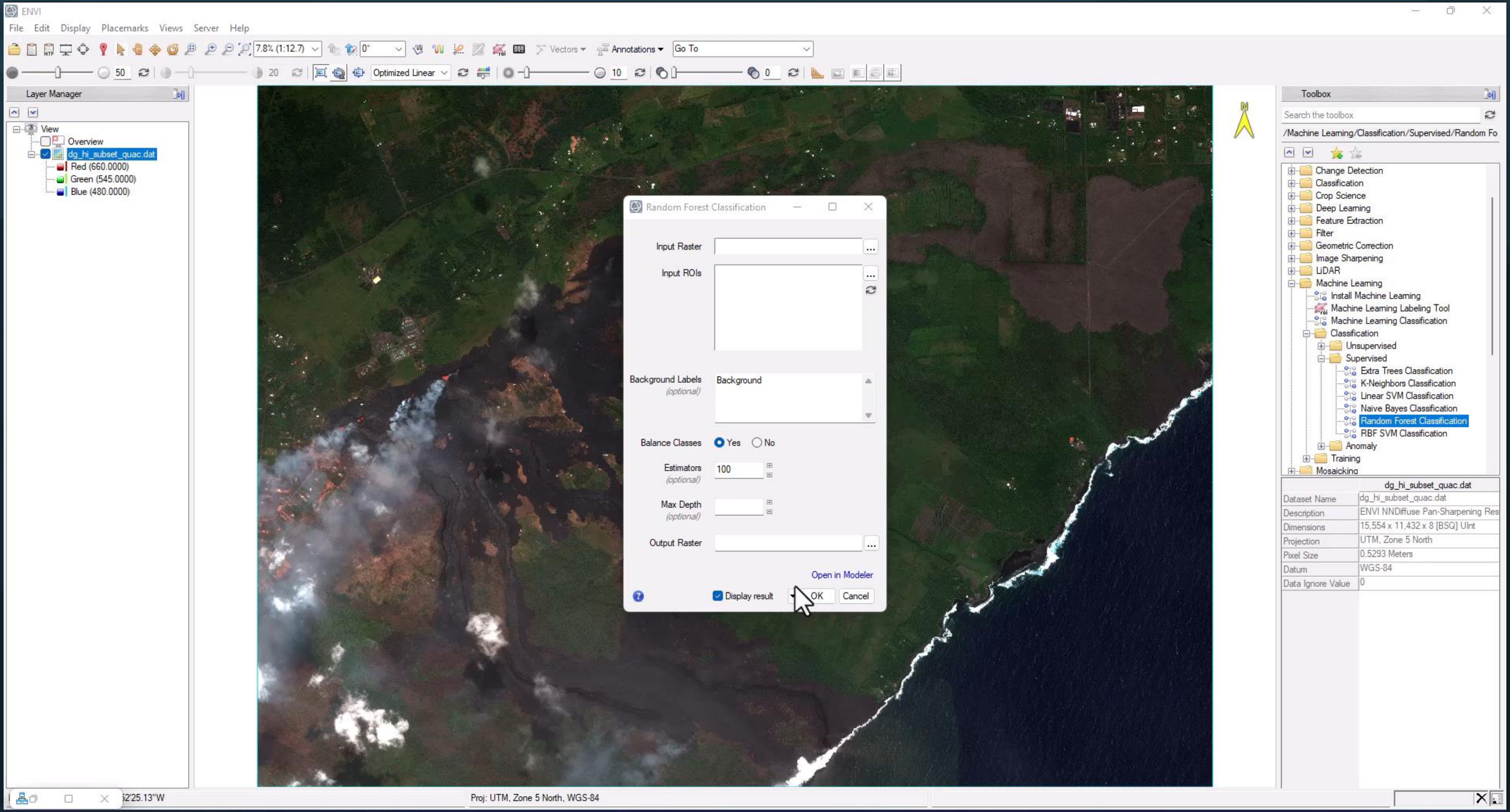



During classification, normalization information is optional and bounds from training data will be used

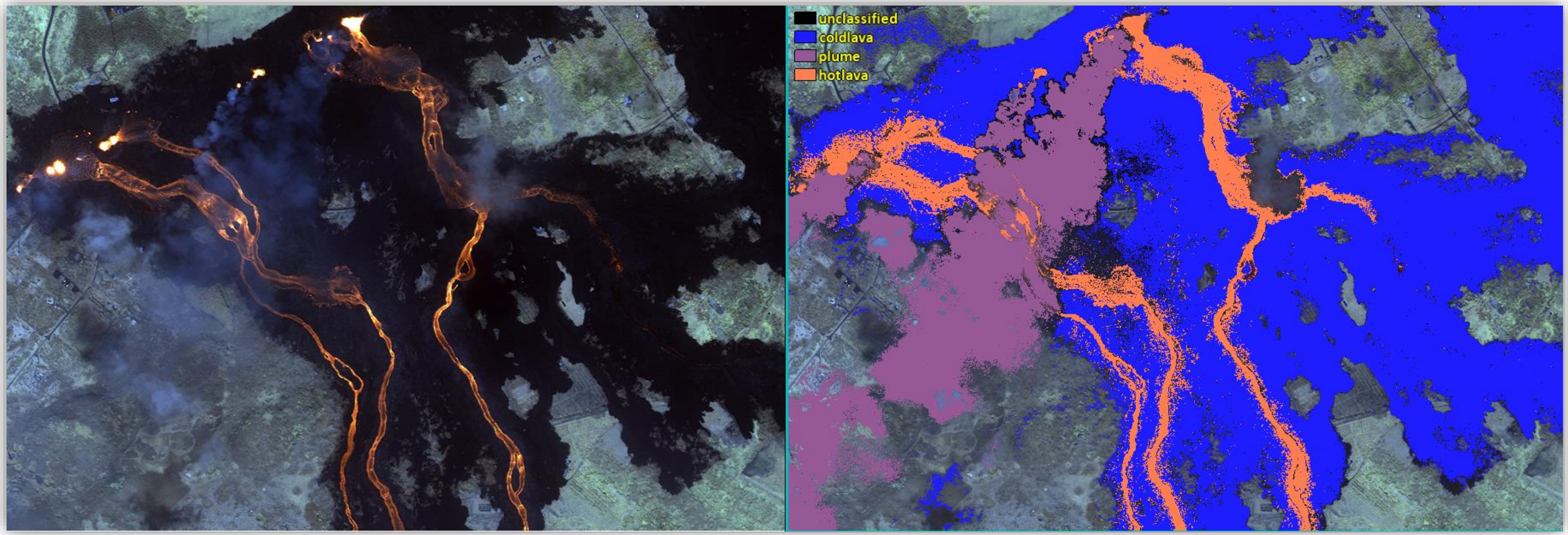
# Quick Classify



# ENVI Modeler Workflow



# Classification Results



# Machine Learning Model Files

Model files are human-readable JSON and contain useful performance metrics by class.

Metrics:

- Overall accuracy
- Precision, recall, accuracy by class
- Easy-to-read confusion matrix

You can edit/tweak class names and colors as desired.

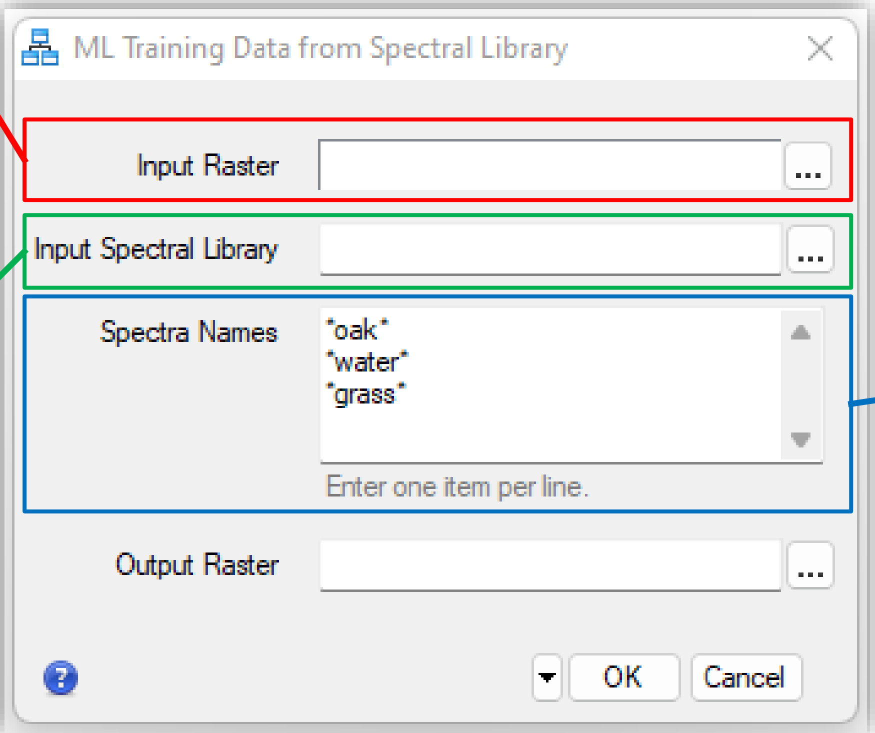

```
{
    "ModelType": "RandomForest",
    "Name": "Random Forest Supervised Classifier",
    "nBands": 8,
    "nClasses": 4,
    "Description": "",
    "Labels": ["Background", "ColdLava", "Plume", "HotLava"],
    "Colors": [[0, 0, 0], [29, 29, 255], [150, 89, 150], [255, 127, 80]],
    "Schema": "envi_ml_model_1.0",
    "ScaleFactor": [0.0000000, 10000.000],
    "Statistics": {
        "report": {
            "Background": {
                "precision": 0.99998556297191044,
                "recall": 0.99998315684108963,
                "f1-score": 0.99998435990505263,
                "accuracy": 0.99998311677348561,
                "samples": 415599
            },
            "ColdLava": {
                "precision": 0.99995745896309263,
                "recall": 0.99996353603247723,
                "f1-score": 0.99996049748855198,
                "accuracy": 0.99998311677348561,
                "samples": 164546
            },
            "Plume": {
                "precision": 0.99999384475234365,
                "recall": 1.0000000000000000,
                "f1-score": 0.99999692236670001,
                "accuracy": 0.99999870129026802,
                "samples": 162462
            },
            "HotLava": {
                "precision": 1.0000000000000000,
                "recall": 0.99996348765882881,
                "f1-score": 0.99998174349612057,
                "accuracy": 0.99999870129026802,
                "samples": 27388
            },
```

```
            },
            "accuracy": 0.99998181806375364,
            "macro_avg": {
                "precision": 0.99998421667183679,
                "recall": 0.99997754513309878,
                "f1-score": 0.99998088081410641,
                "samples": 769995
            },
            "weighted_avg": {
                "precision": 0.99998181810824638,
                "recall": 0.99998181806375364,
                "f1-score": 0.99998181806939224,
                "samples": 769995
            }
        },
        "confusion_matrix": [
            "               Background    ColdLava       Plume     HotLava",
            "   Background      415592           6           0           0",
            "     ColdLava           6      164540           0           1",
            "        Plume           1           0      162462           0",
            "      HotLava           0           0           0       27387"
        ]
    },
    "Classifier": "eF7UfQd8VMXz+IGFKl2KihxFmoIgSCe7FAEJJRQpguYCBEJNSEKV8gDpAUECIvXoVXo
}
```

# Spectral Libraries and Machine Learning

Raster with wavelength information used to resample the spectral library

Spectral library that we want to use for training data

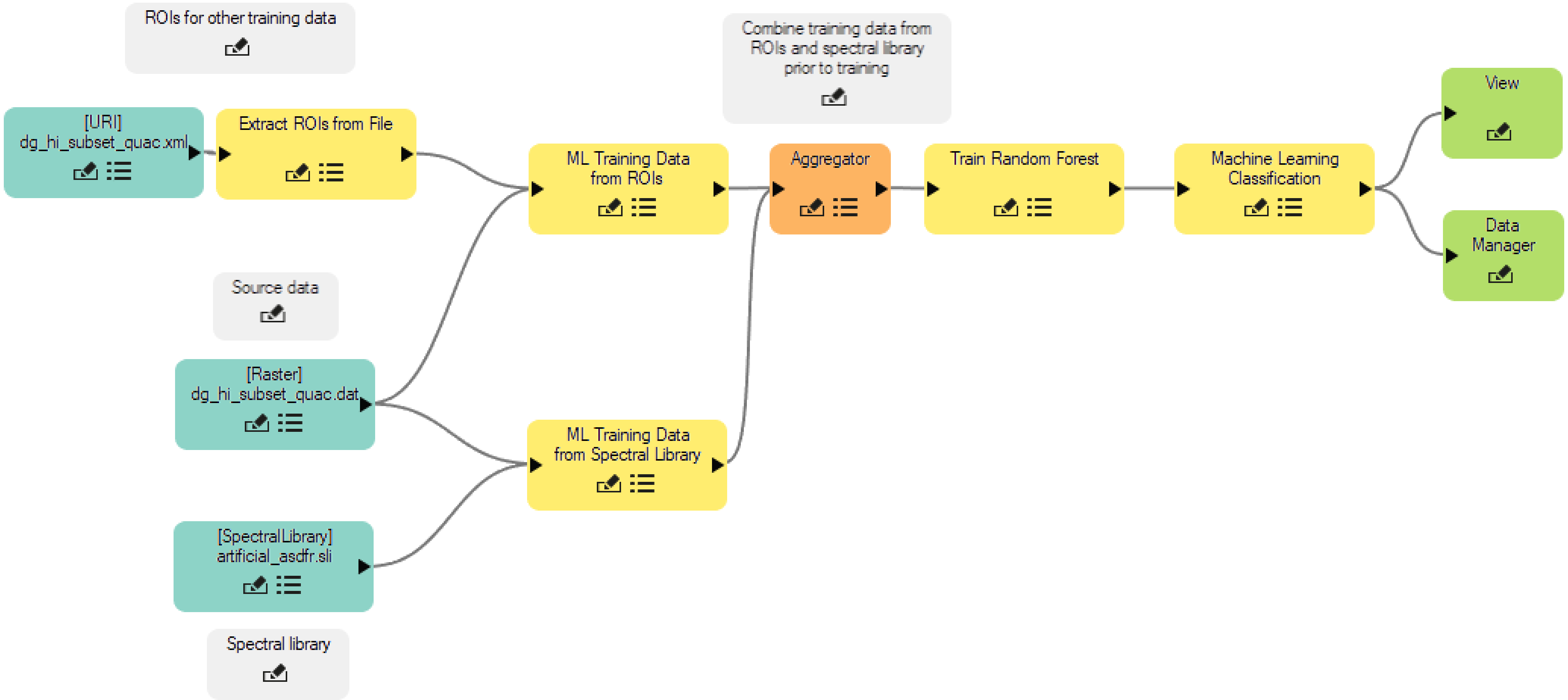



Specify the names of spectra to extract as individual classes for machine learning

Use “*” for wildcards

Any spectra that don’t match the wildcards are converted to background samples

# Example: Combine Spectral Libraries with ROIs



# Intel Optimizations

- Machine learning uses the CPU and not GPUs
- We ship an optimized version of the machine learning algorithms for Intel CPUs
- Some algorithms are dramatically faster on Intel and almost unusably slow without Intel CPUs

Sample algorithm performance improvements with Intel-optimized libraries.
Note that “RBF SVM Classify” is an approximate time as it never finished.

| Task | Time [s] | Time [s] w/ Intel | % Change |
|---|---|---|---|
| RBF SVM Train | 94.5 | 6.7 | -93% |
| RBF SVM Classify | 15000.0 | 69.4 | -100% |
| Nearest Neighbors Classify | 257.3 | 6.6 | -97% |
| Random Forest Train | 2.3 | 2.1 | -9% |
| Random Forest Classify | 18.5 | 13.1 | -29% |

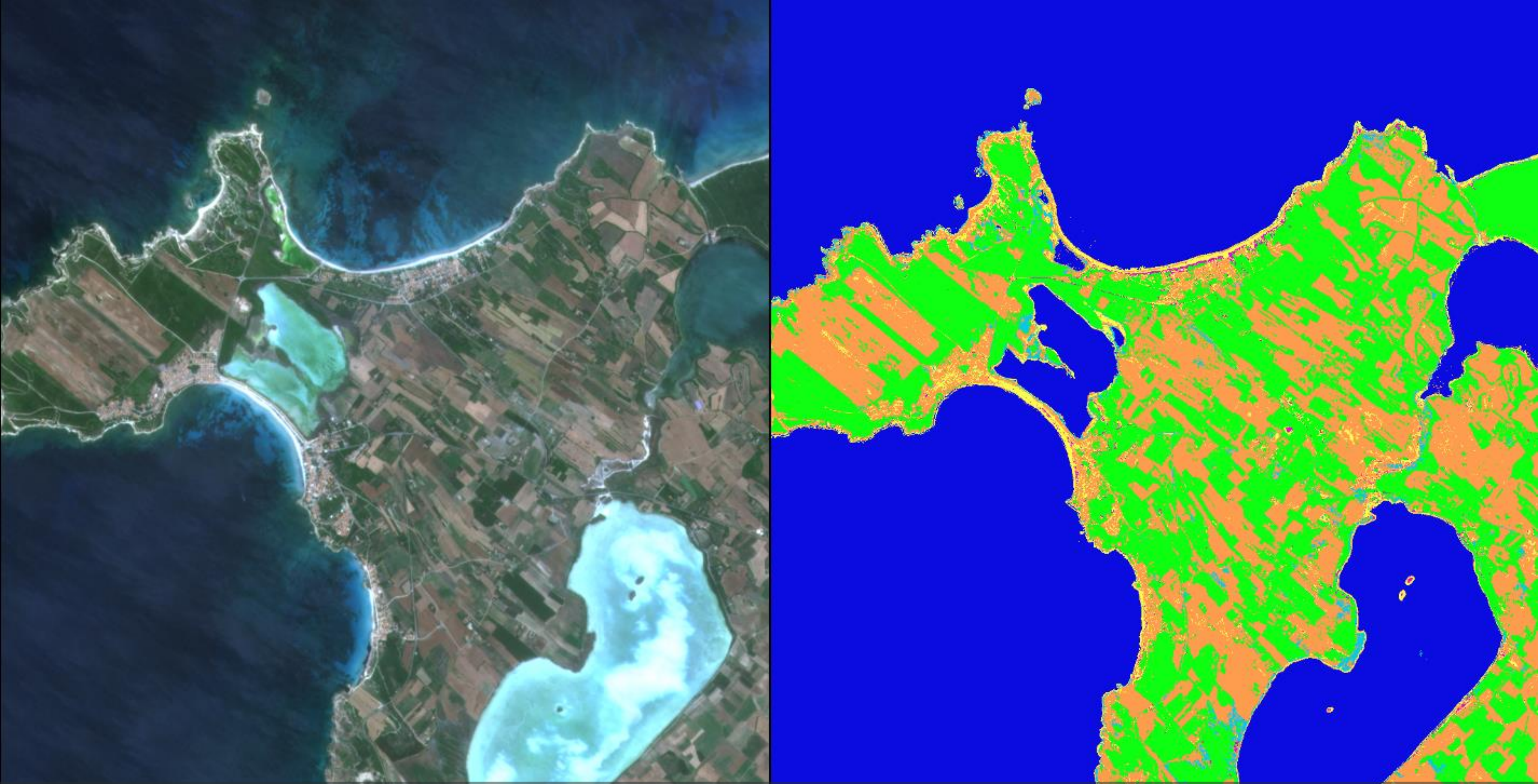
# Machine Learning vs Deep Learning

| MACHINE LEARNING | DEEP LEARNING |
|---|---|
| Easy to use | Harder to use |
| Train and get results in minutes | Train and get results in hours-days |
| Not as much data management | Lots of data management |
| Trains on CPU | Primarily trains on GPU (also CPU) |
| Not affected by resolution | Affected by resolution |
| Mostly spectral | Mostly spatial, some spectral |
| Quality over quantity for training data | Quality and quantity for training data |

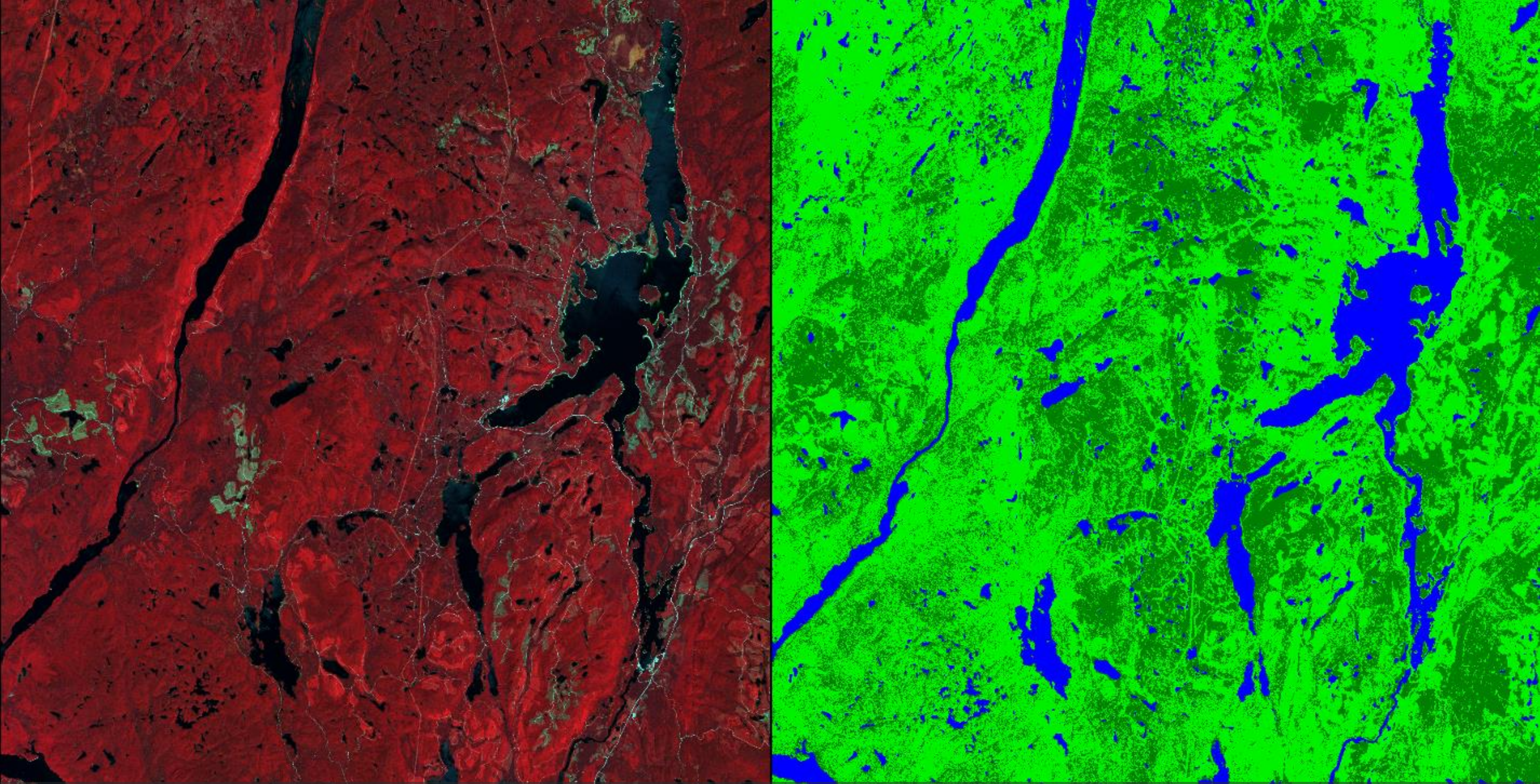
# Traditional Approaches vs Machine Learning

| TRADITIONAL | MACHINE LEARNING |
|---|---|
| • Examples:<br>– Spectral Angle Mapper<br>– Mahalanobis Distance | • Examples:<br>– Random Forest<br>– SVM |
| • Algorithm specific parameters dictate performance | • No thresholds |
| • Often rule images or ancillary files providing additional metrics | • Only classification image |
| • Only specify the things you want to find | • Provide examples of features not to detect |
| • Does not classify every pixel | • Classifies every pixels |
| • No lengthy or CPU/GPU intensive training | • No lengthy or CPU/GPU intensive training |
| • Statistics from regions of interest as samples | • Pixels from regions of interest as samples |

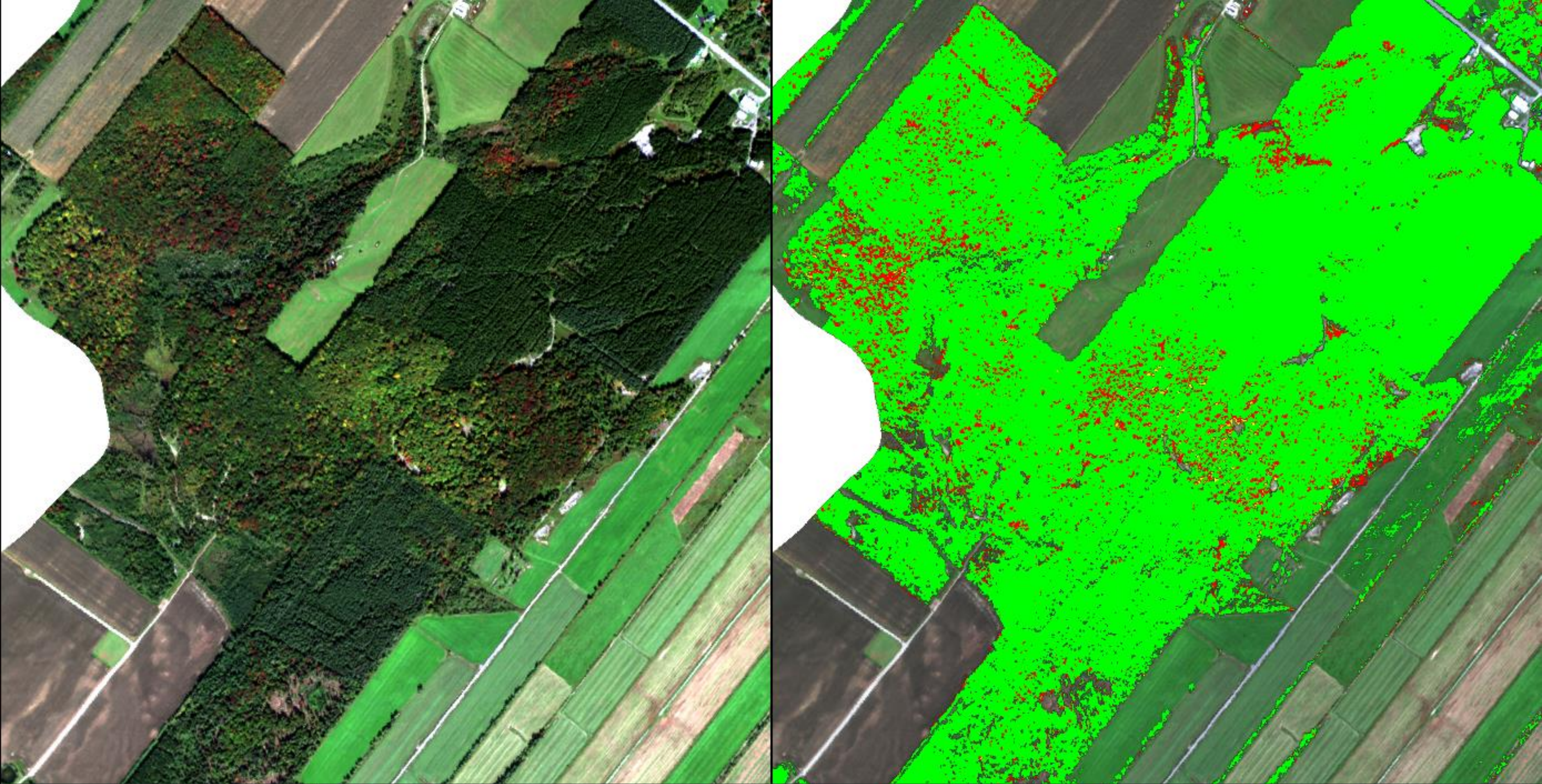
# Machine Learning Use Cases and Examples



Landcover classification using PlanetScope data

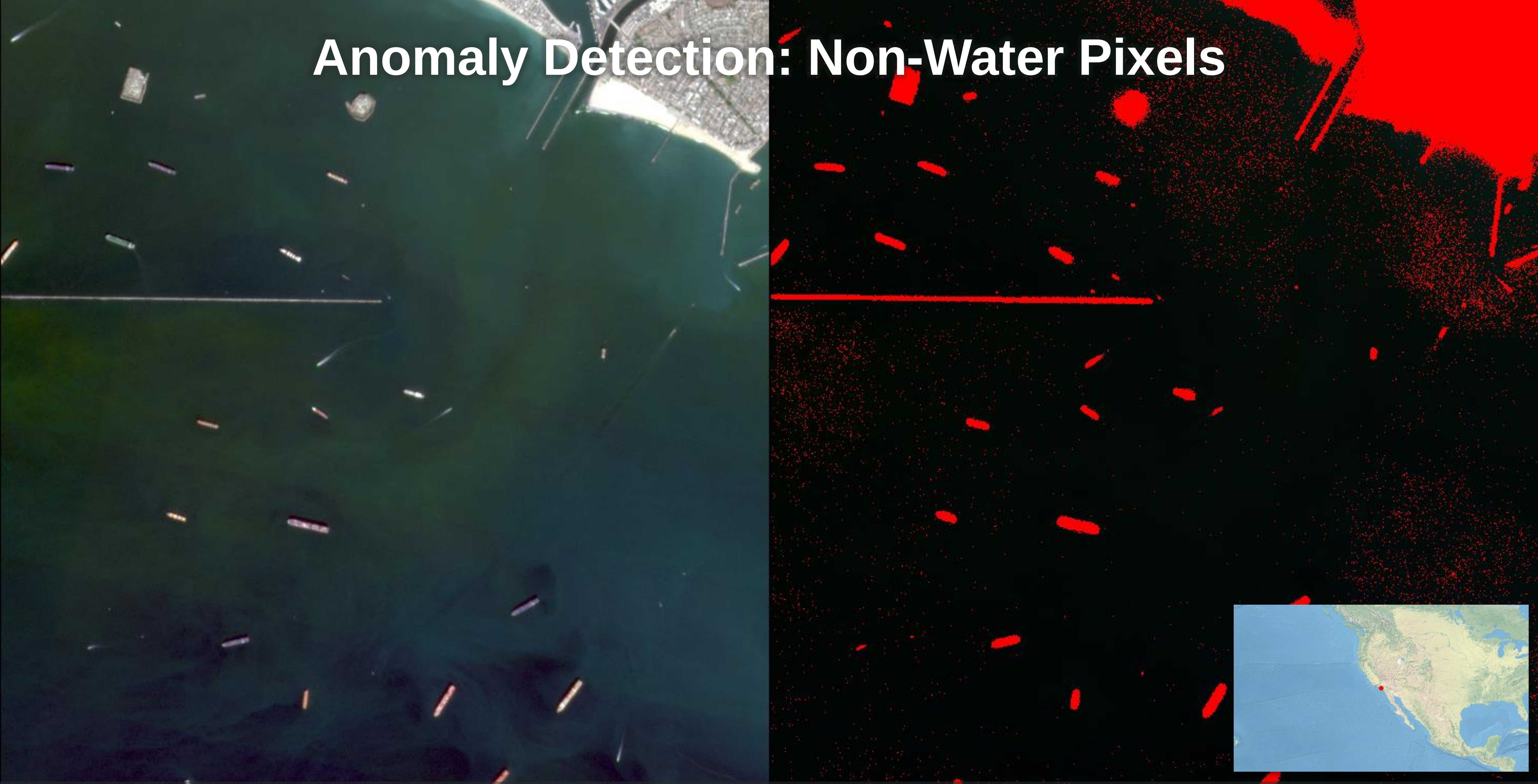


**Landcover classification using unsupervised machine learning algorithms**

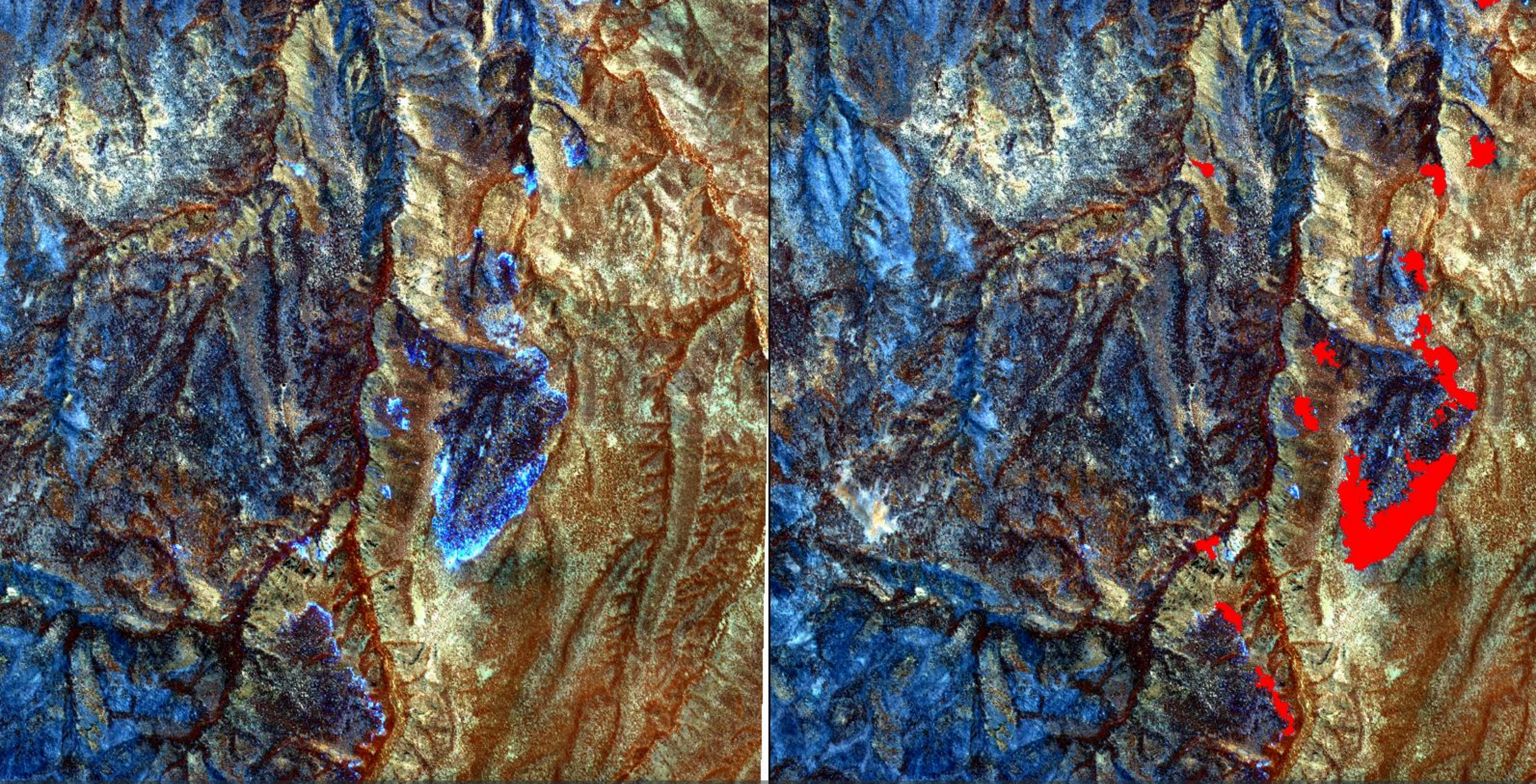


Use hyperspectral data to identify trees at different stages of shedding their leaves

# Anomaly Detection: Non-Water Pixels

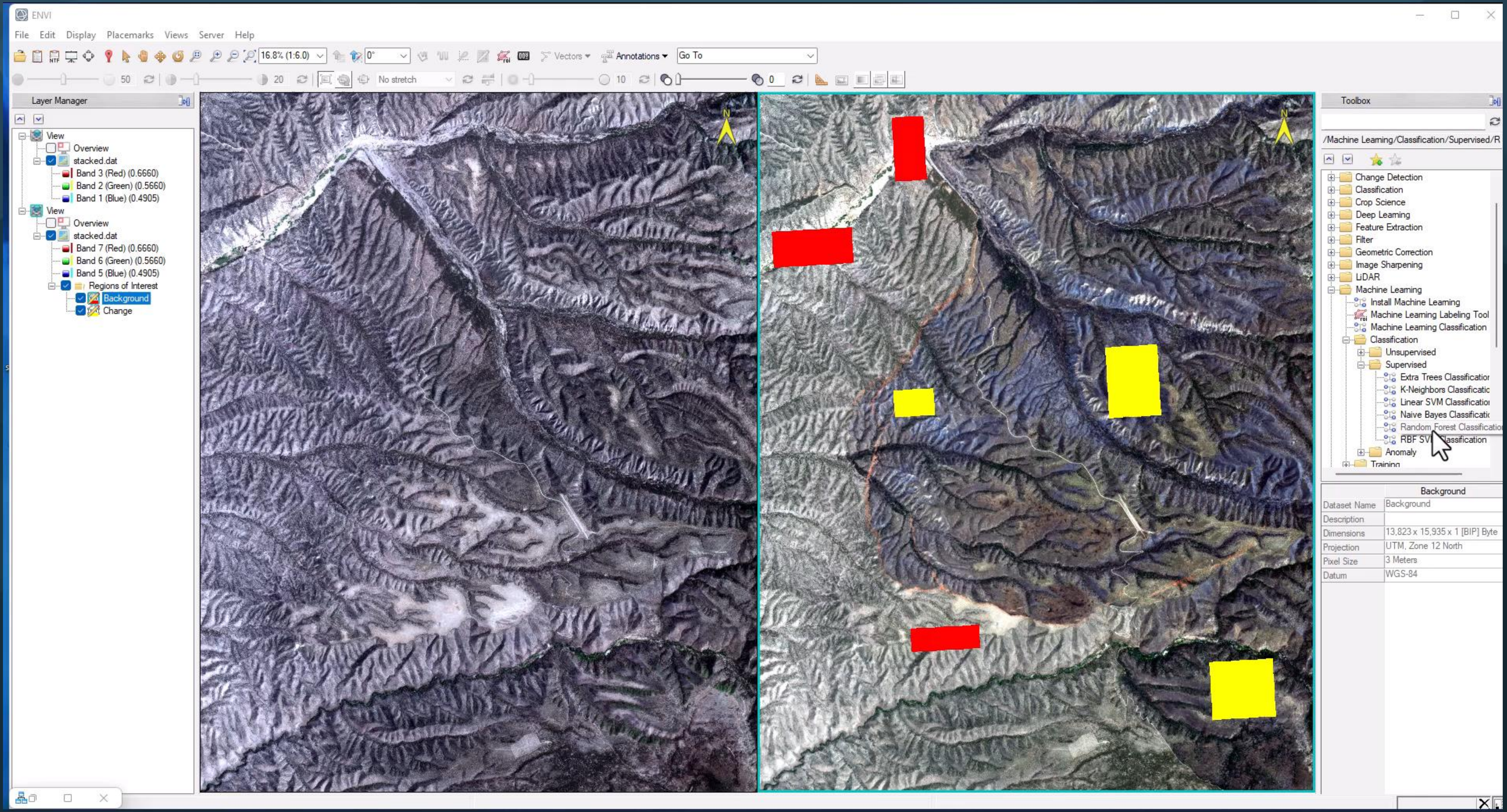


33.712012°N, 118.136288°W



Using anomaly detection to identify fires in WorldView 3 SWIR data

# Supervised Change Detection: PlanetScope



# Questions and Discussion


**JP Metcalf**

*Solutions Engineer*

jp.metcalf@l3harris.com

**Zachary Norman**

*Product Manager*

zachary.norman@l3harris.com

**L3Harris Geospatial**

**www.L3HarrisGeospatial.com**
geospatialinfo@l3harris.com
303-786-9900