



ArcGIS Image Analyst: An Introduction

Kyle Talbot, Vinay Viswambharan

Presenters



Kyle Talbot
Solutions Engineer
Imagery and Remote Sensing



Vinay Viswambharan
Principal Product Manager
Imagery Analytics and AI

Imagery and Remote Sensing

ArcGIS is a comprehensive imagery system



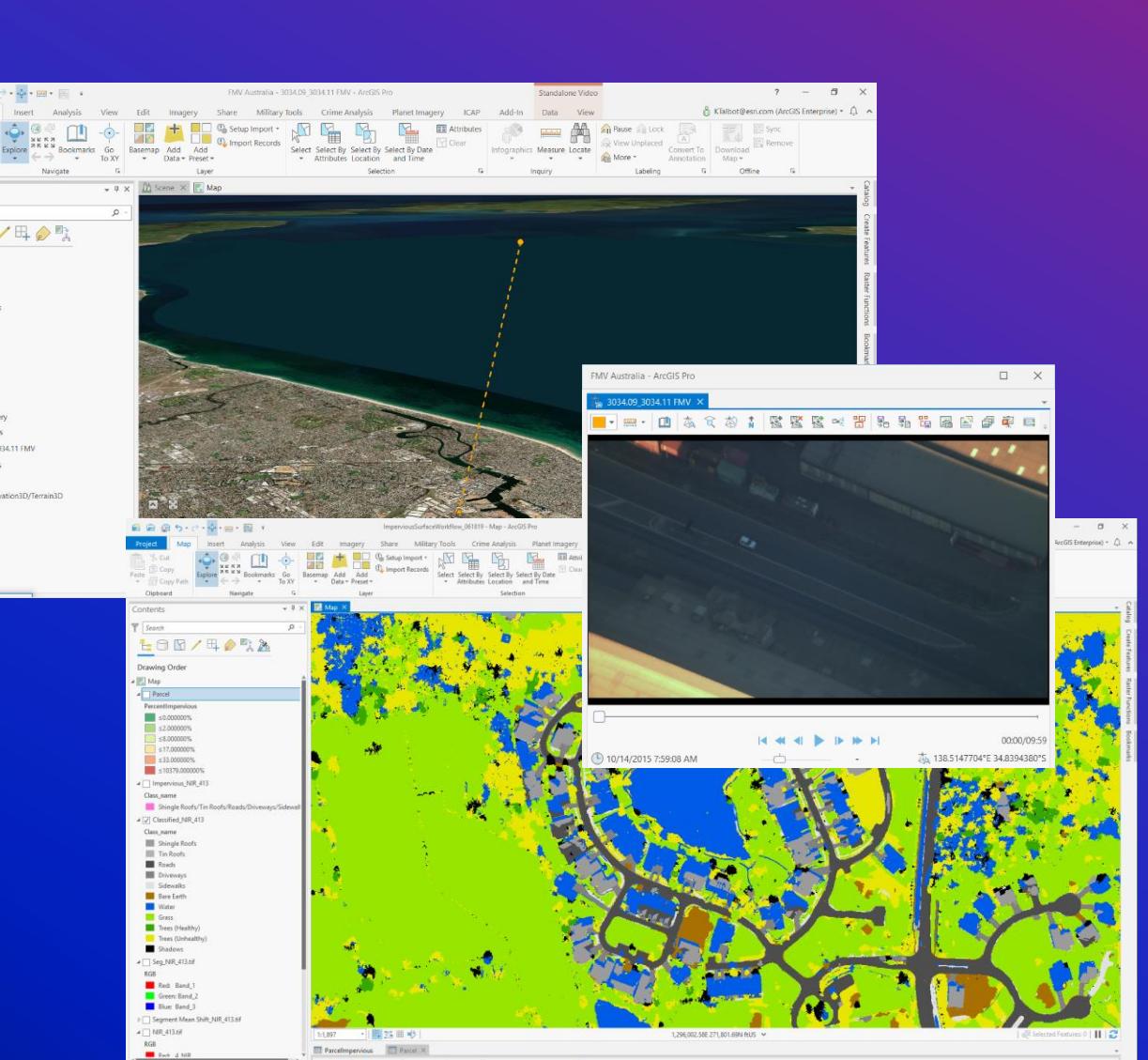
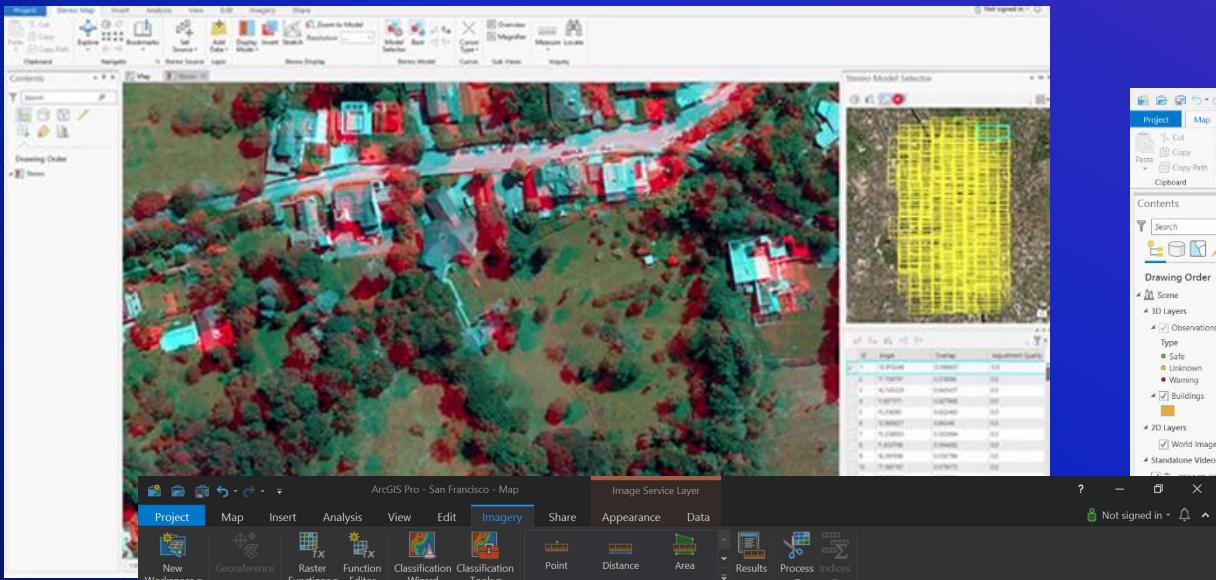


Image Analyst Extension

What is the Image Analyst extension?

- The **Image Analyst extension** is an application extension which extends **ArcGIS Pro** with advanced imagery analysis tools, workflows, and user experiences.
- **Image Analyst is for Image Analysts and Geospatial Analysts who focus on:**
 - visual **enhancement** and exploitation of imagery
 - creation of derived products from imagery
 - taking **measurements** from imagery
 - capturing features from **stereo** imagery
 - **advanced analysis** and **image processing**
 - advanced analysis of **multidimensional raster datasets**
 - exploitation and analysis of **motion imagery** (FMV)
 - extraction information from imagery using AI (**deep learning**) models
 - **editing** of imagery and raster datasets



Image Analyst extension Product Information

- **Availability**

- first release was ArcGIS Pro 2.1
- available for:
 - ArcGIS Pro Basic
 - ArcGIS Pro Standard
 - ArcGIS Pro Advanced

- **Pricing**

- the same as the Spatial Analyst extension

- **Enterprise Agreements**

- Category B (addition to the EA)

- **ArcGIS Image Server**

- All Image Analyst capabilities which are available on the server are included at no additional cost!

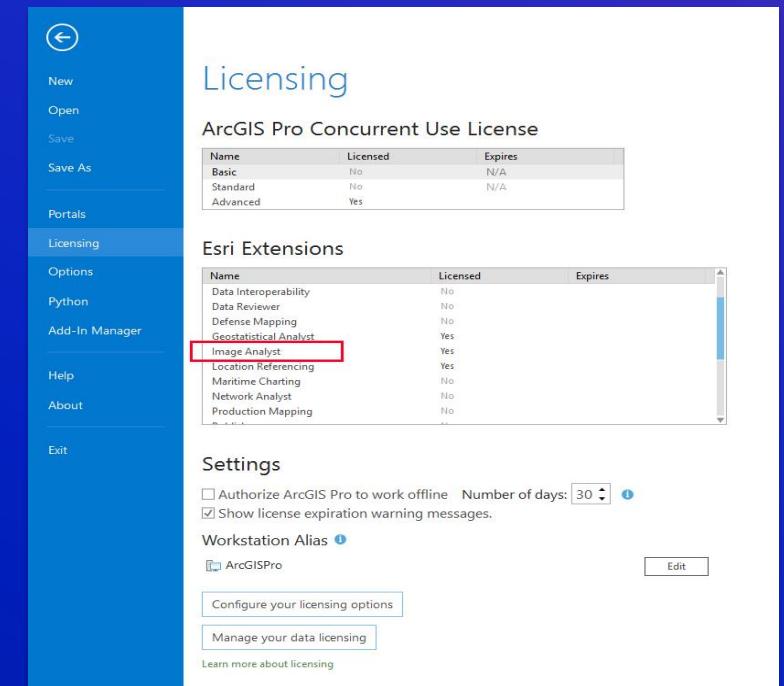
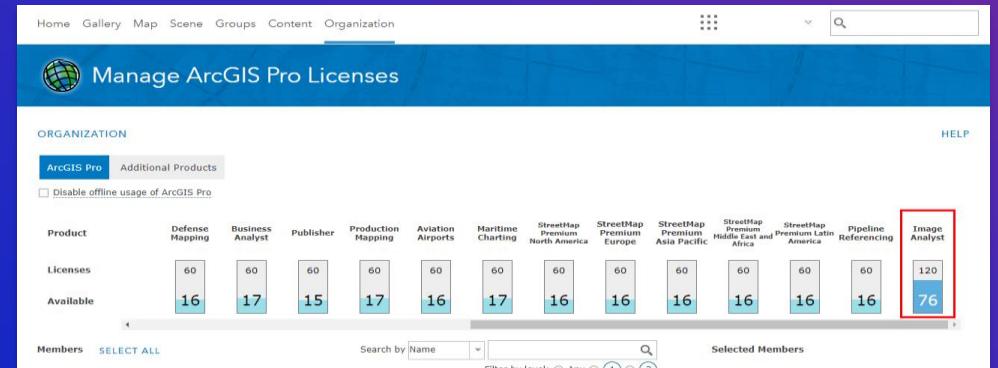


Image Analyst Capabilities

Stereo Mapping

Visualize imagery and capture 3D feature data in a stereo viewing environment.

Advanced Raster Functions

Perform real-time raster analysis and image processing on an extensive suite of remote sensing data types, and save your results if desired.

Image Classification

Perform object-based and traditional image analysis using image segmentation and classification tools and capabilities.

Advanced Multidimensional Analysis

Perform advanced raster modeling with multidimensional geospatial data using geoprocessing tools, Python, Notebooks, and the ArcPy API

Perspective Imagery

Work with oblique imagery oriented in a natural perspective mode to facilitate effective image interpretation applications.

Pixel Editor

Redact sensitive areas from images, clean up raster analysis results, and edit DEMs.

Motion Imagery

Work with geospatially enabled video data together with your GIS data to assist in timely, well-informed decision support.

Deep Learning

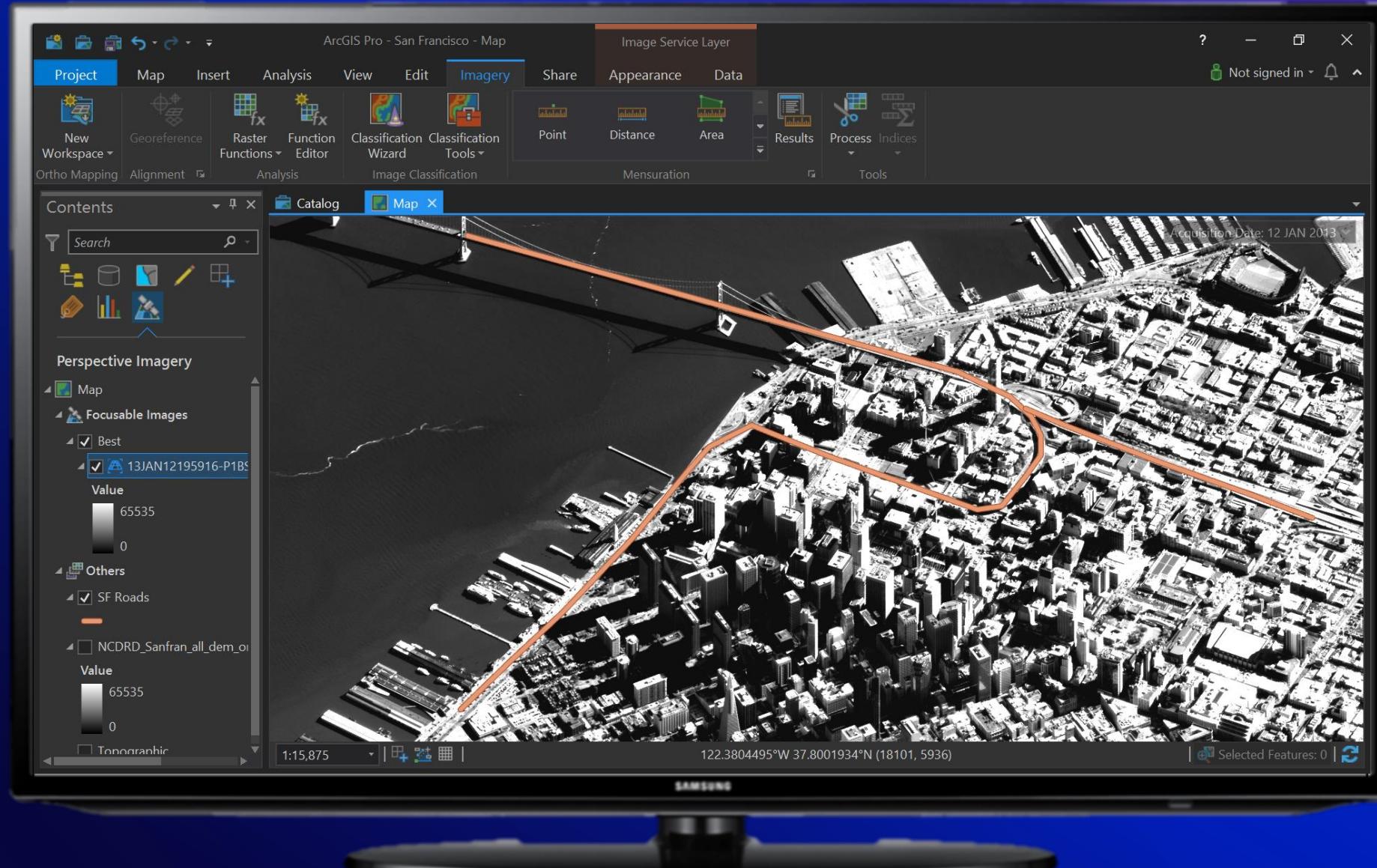
Extract geospatial features from imagery using deep learning techniques such as object detection and image classification / segmentation

Change Detection

- Perform pixel-to-pixel or time-series based change detection using intuitive wizards, geoprocessing tools, and the ArcPy API

Image Coordinate Space

ArcGIS Pro with Image Analyst Extension

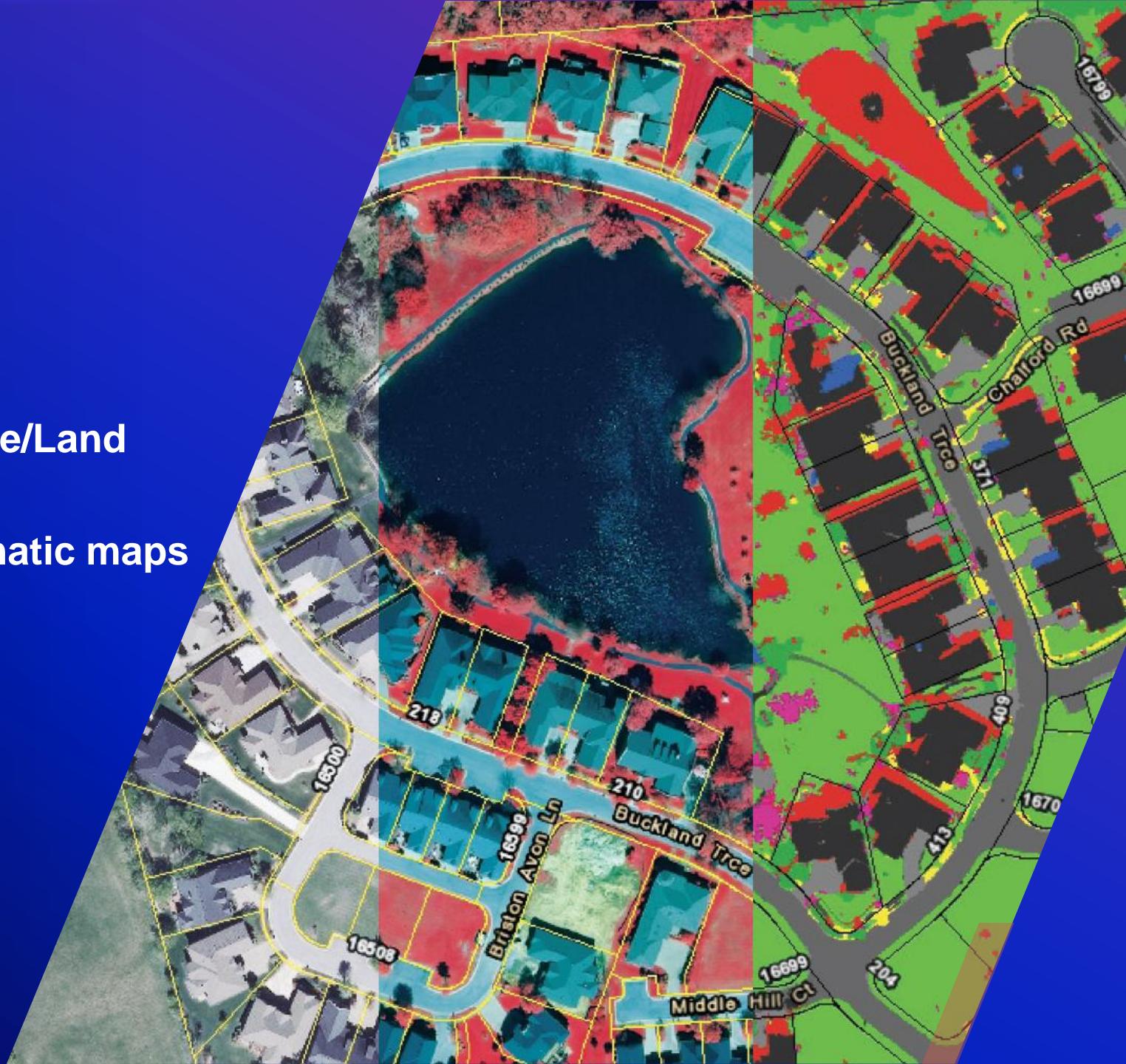


Capabilities:

- Work with highly-oblique imagery
- Switch between map-centric and image-centric views
- Capture feature information correctly

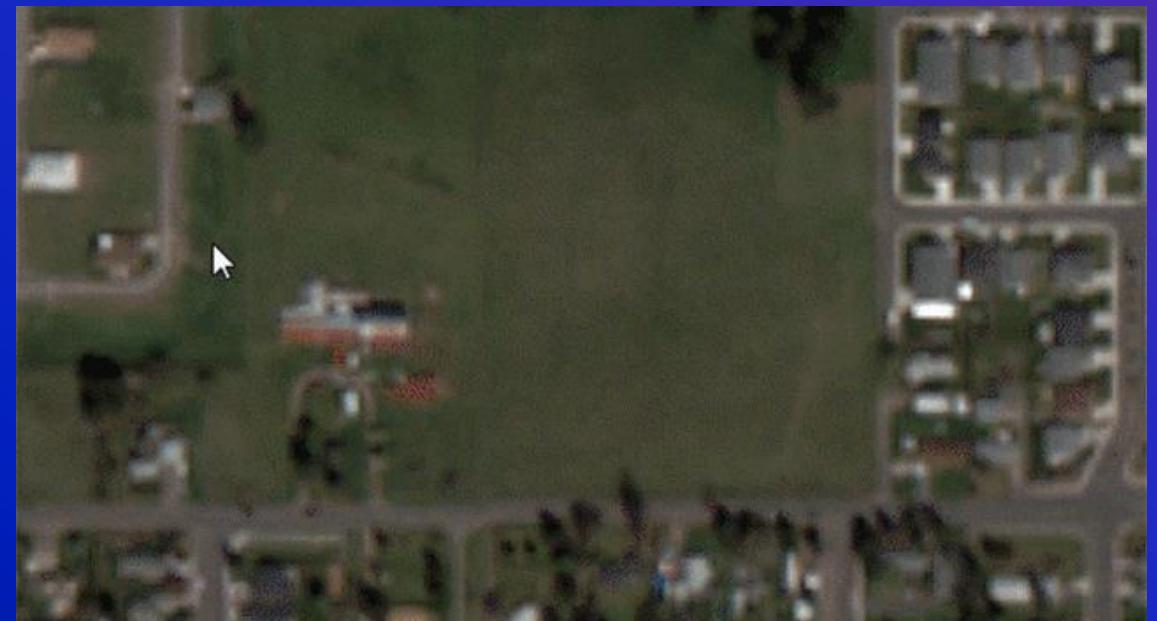
Image Classification Tools

- Supervised or Unsupervised
- Object or Pixel Based
- Assigning Classes in a Land Use/Land Cover System
- Outputs are used to create thematic maps
- Supporting Layer in a GIS

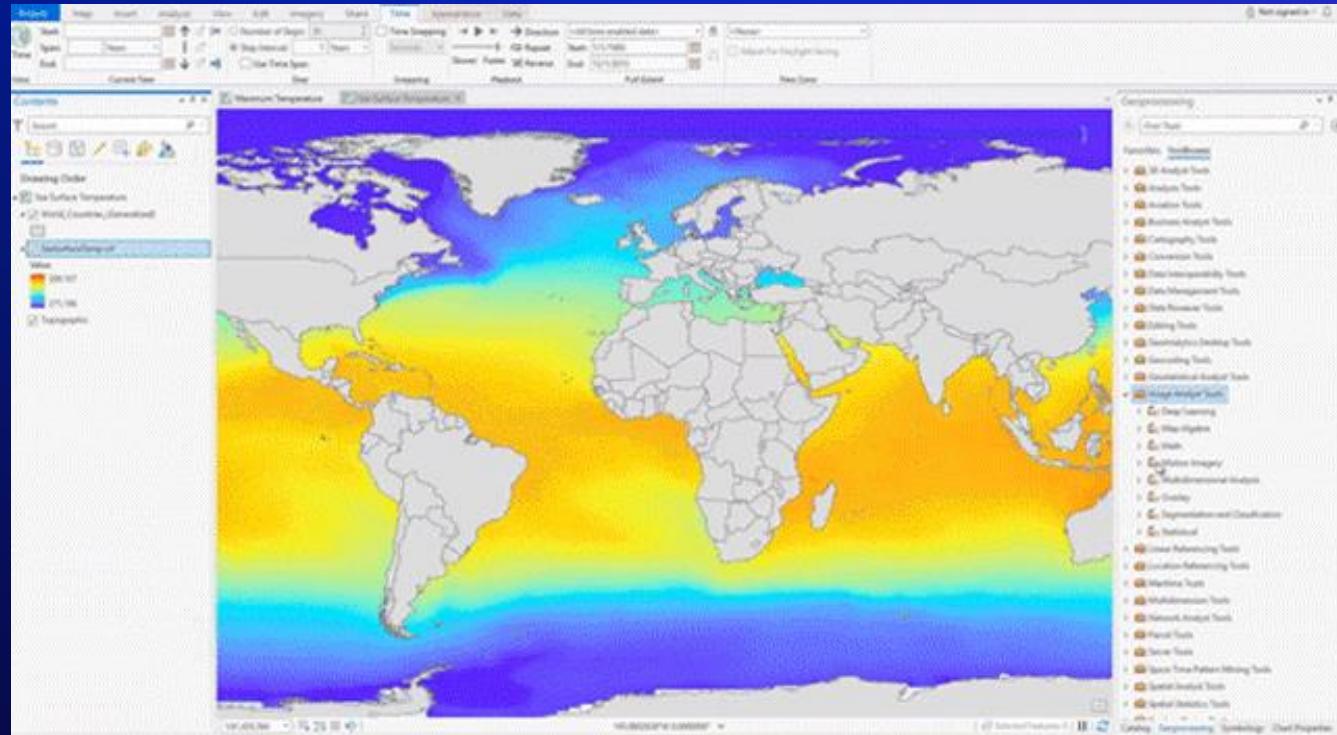


Pixel Editor

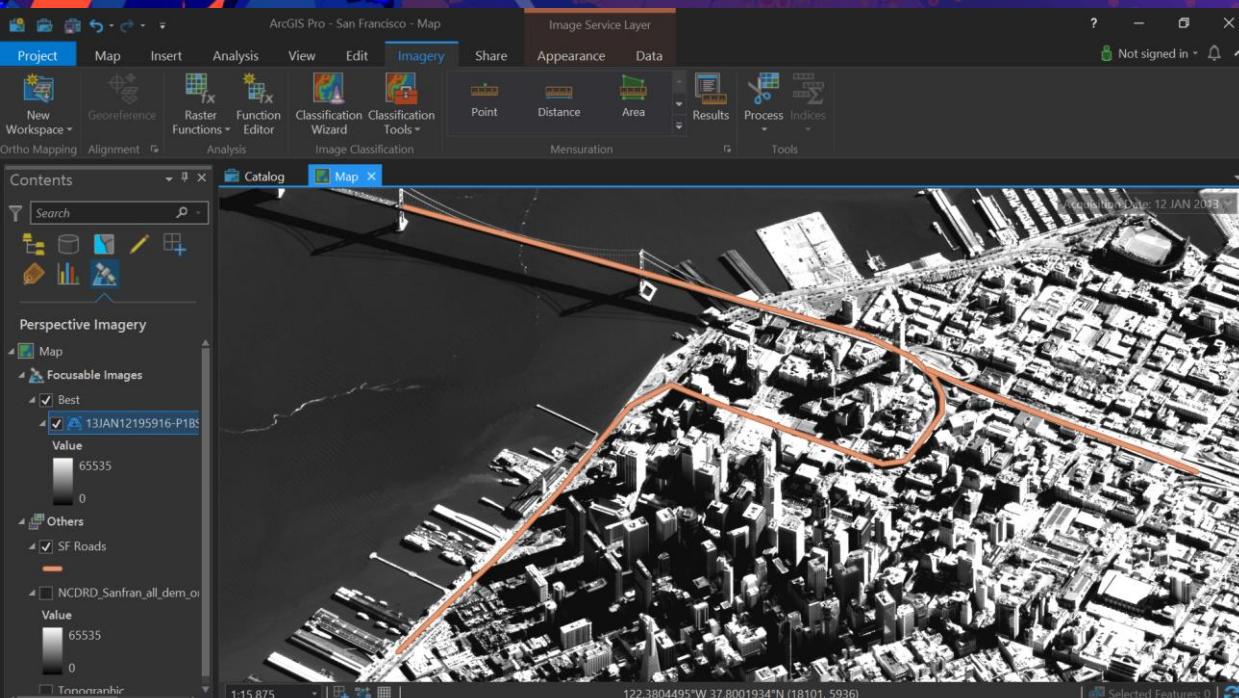
- Manipulate pixel values for raster and imagery data
 - Elevation data
 - Reclassify
 - Redaction
 - Replace clouds



Multidimensional Rasters



- **Create and manage datasets**
 - Dedicated multidimensional tab
 - Data management group
- **Explore**
 - Extent group
 - Current Display Slice
- **Perform complex analysis**
 - Temporal profile charts
 - Geoprocessing tools
 - Aggregate
 - Anomaly
 - Trend



Demos

Kyle Talbot
Vinay Viswambrahan

What's New

Deep learning

- New models and tools

Change detection

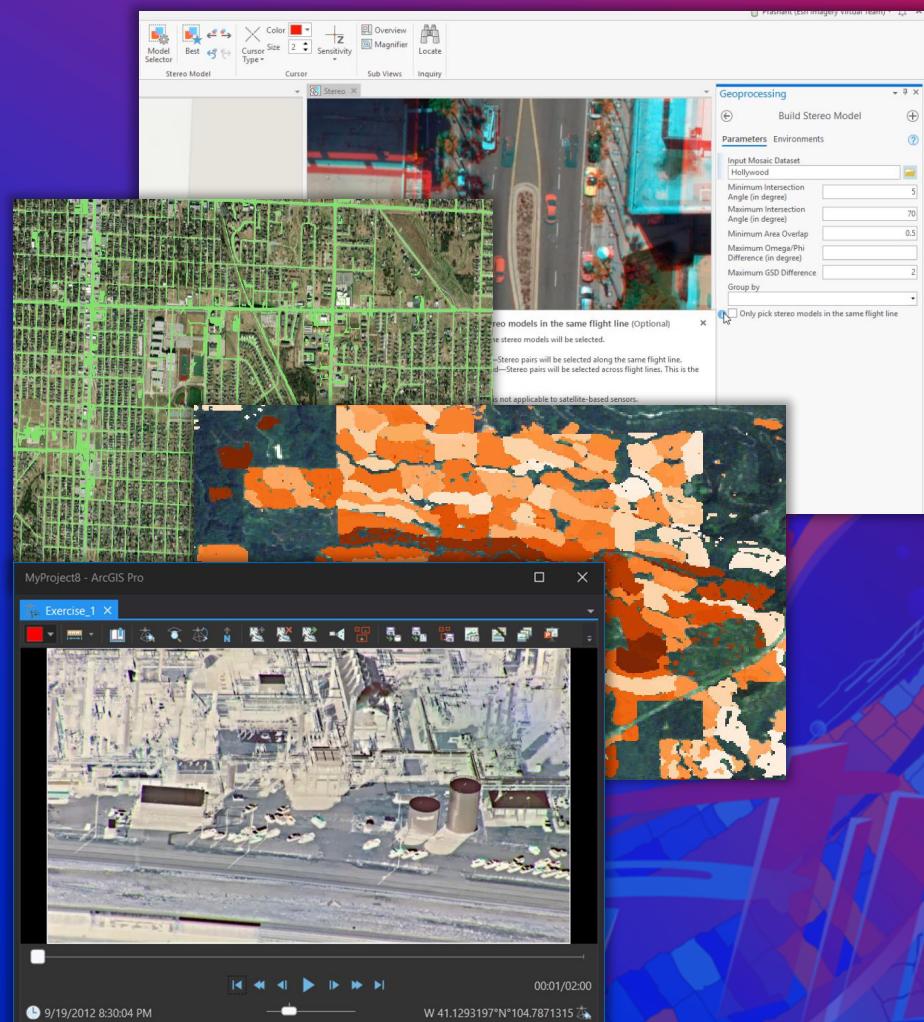
- New wizard and algorithms

Stereo mapping

- User experience improvements when working with stereo collections

Motion imagery

- Time slider improvements
- Application settings
- UX improvements

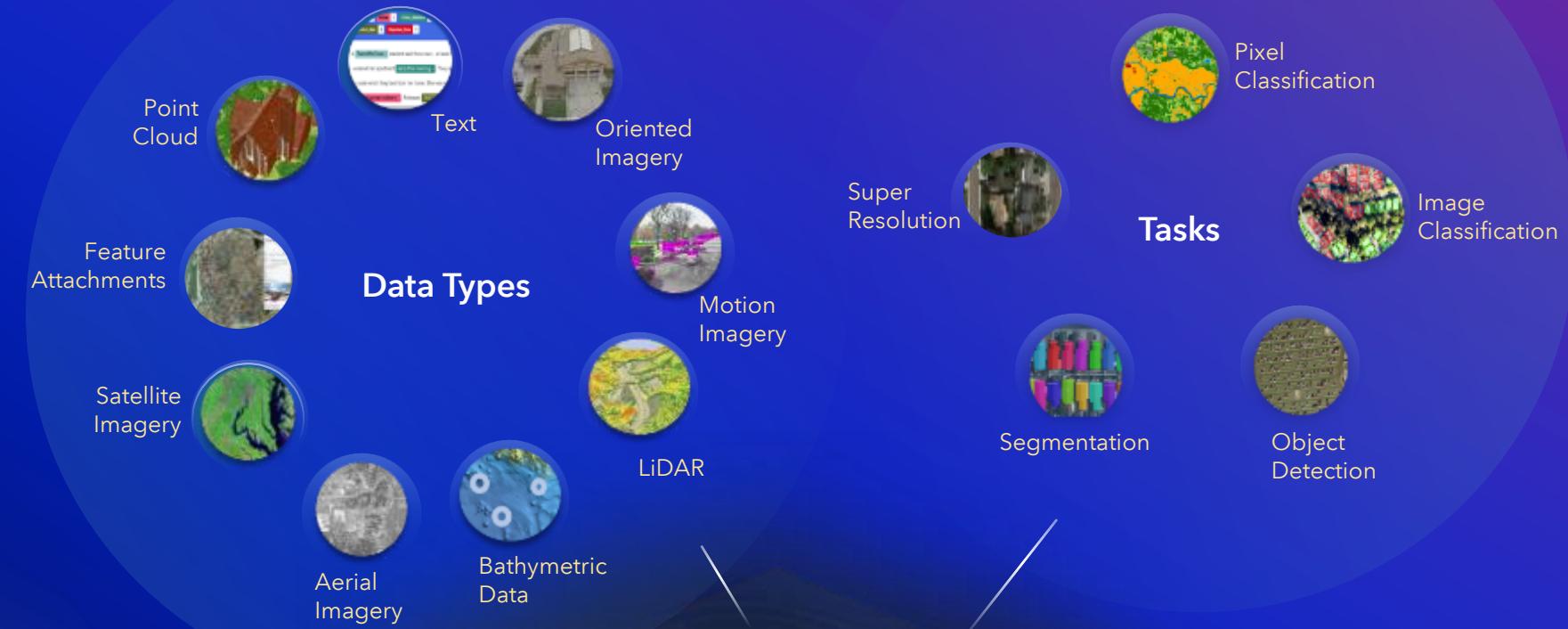


Deep Learning

New model and tools

New

- FasterRCNN
- YOLOv3
- Multispectral support
- Sparse training
- Multilabel classification



Integration

Deep Learning

New model types

New Object Detection Models

- YOLOV3
- FasteRCNN

Applications

- Detect cars, trees, planes
- Shipwrecks
- Fire hydrants
- Detect encroaching features



New Pixel Classification Models

- DeepLab

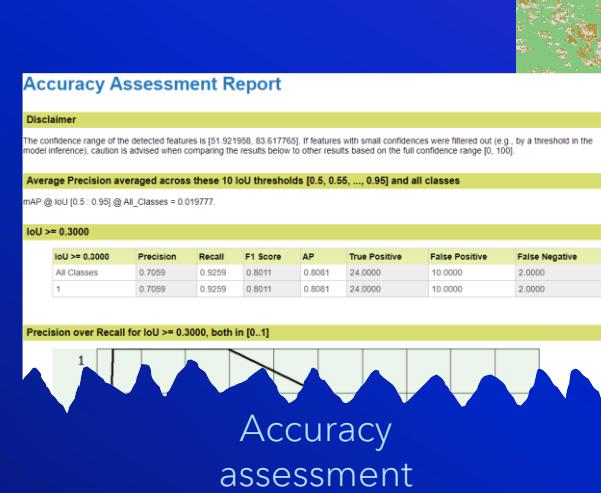
Applications

- Extract road networks
- Routing

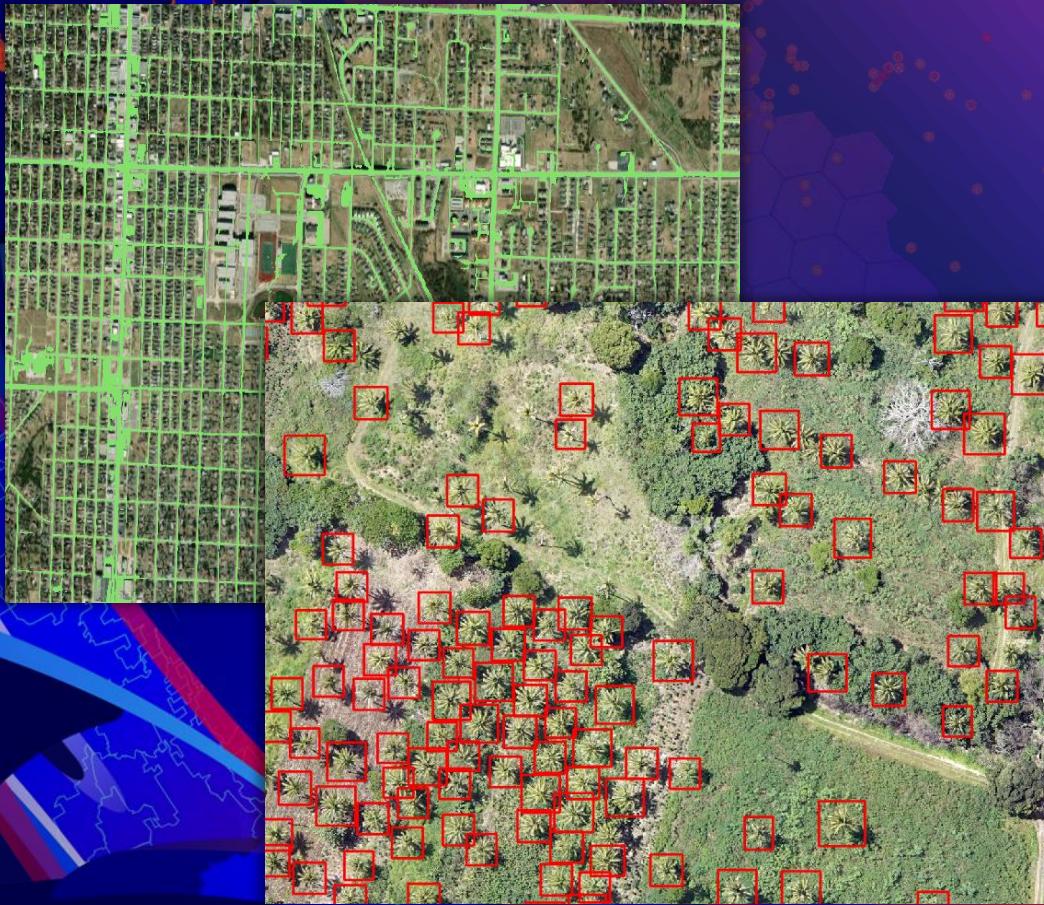


Deep Learning Multispectral Imagery

- Perform training on multispectral imagery
- Enhanced training for better results especially with sparse training data
- Enhance “Feature Classification” model type to classify a feature into Multiple labels
- Better accuracy assessment results
 - Recall
 - Precision
 - F1score
 - Average Precision



Multilabel
classification

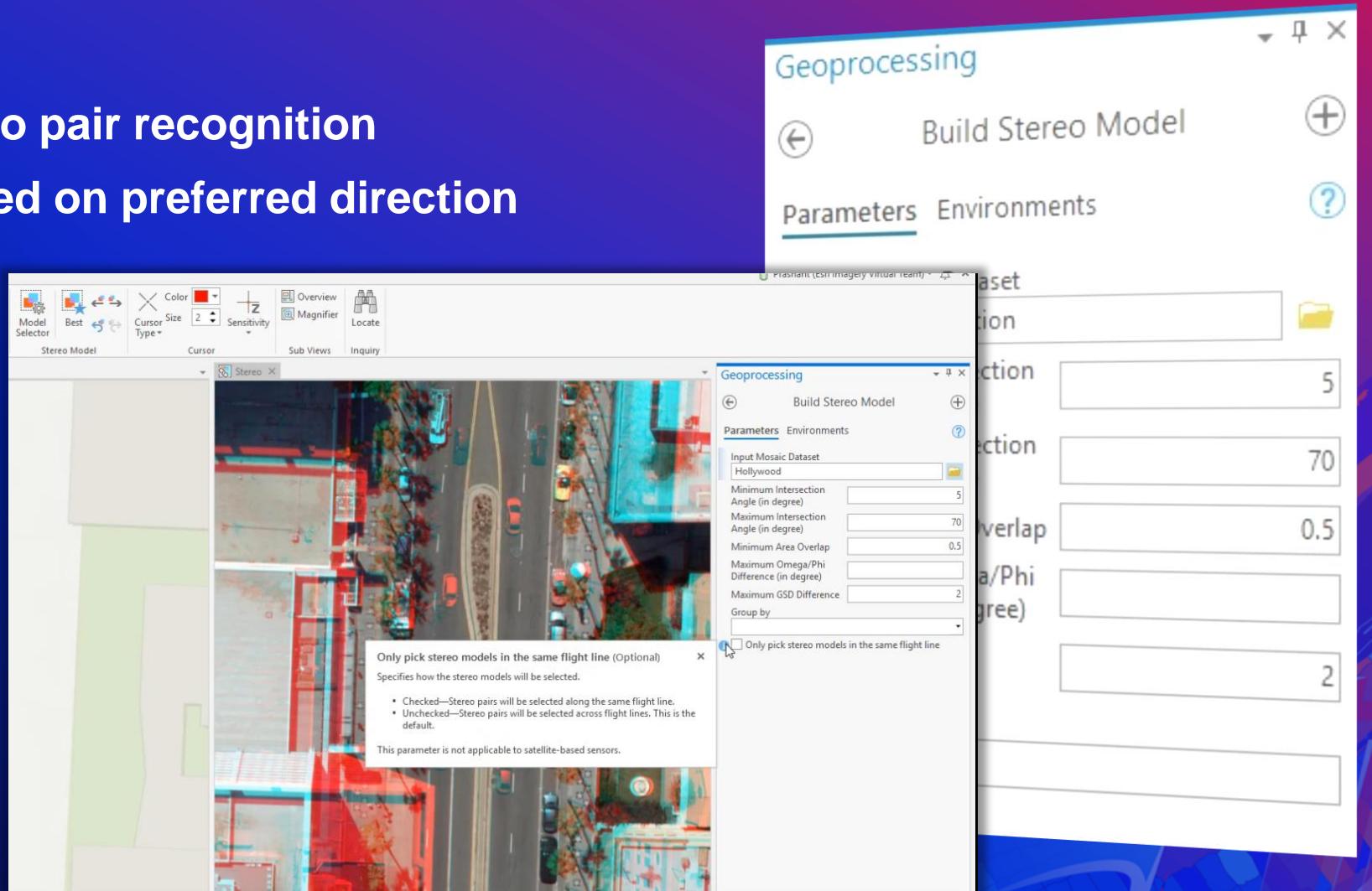


Demo - Deep Learning in ArcGIS Pro

Vinay Viswambahan

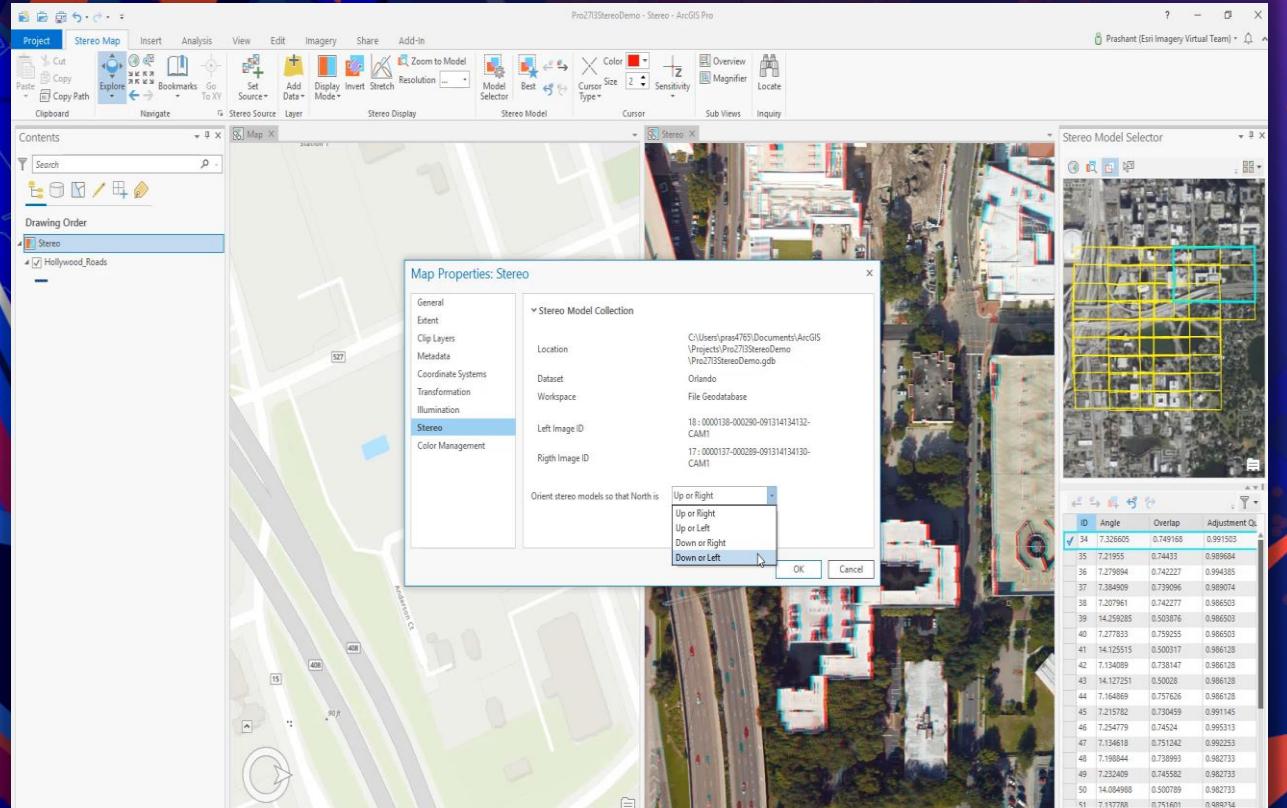
Stereo Mapping

- More control over stereo pair recognition
- Orient stereo pairs based on preferred direction



Demo - Stereo Mapping

Kyle Talbot



Change Detection

Compare images or time-series data

Image to Image Change Detection

2 images
Identify what are the changes
Simple static change detection
Percent Change
Change in Categorical data

Time Series Analysis

Time series data
CCDC and LandTindr
Identify when is the change and then explor

Detecting Change

Most recent
Number of changes
Earliest
Largest change
Duration of change

Classify

Classify time Slices
Can be Input to simple change

New

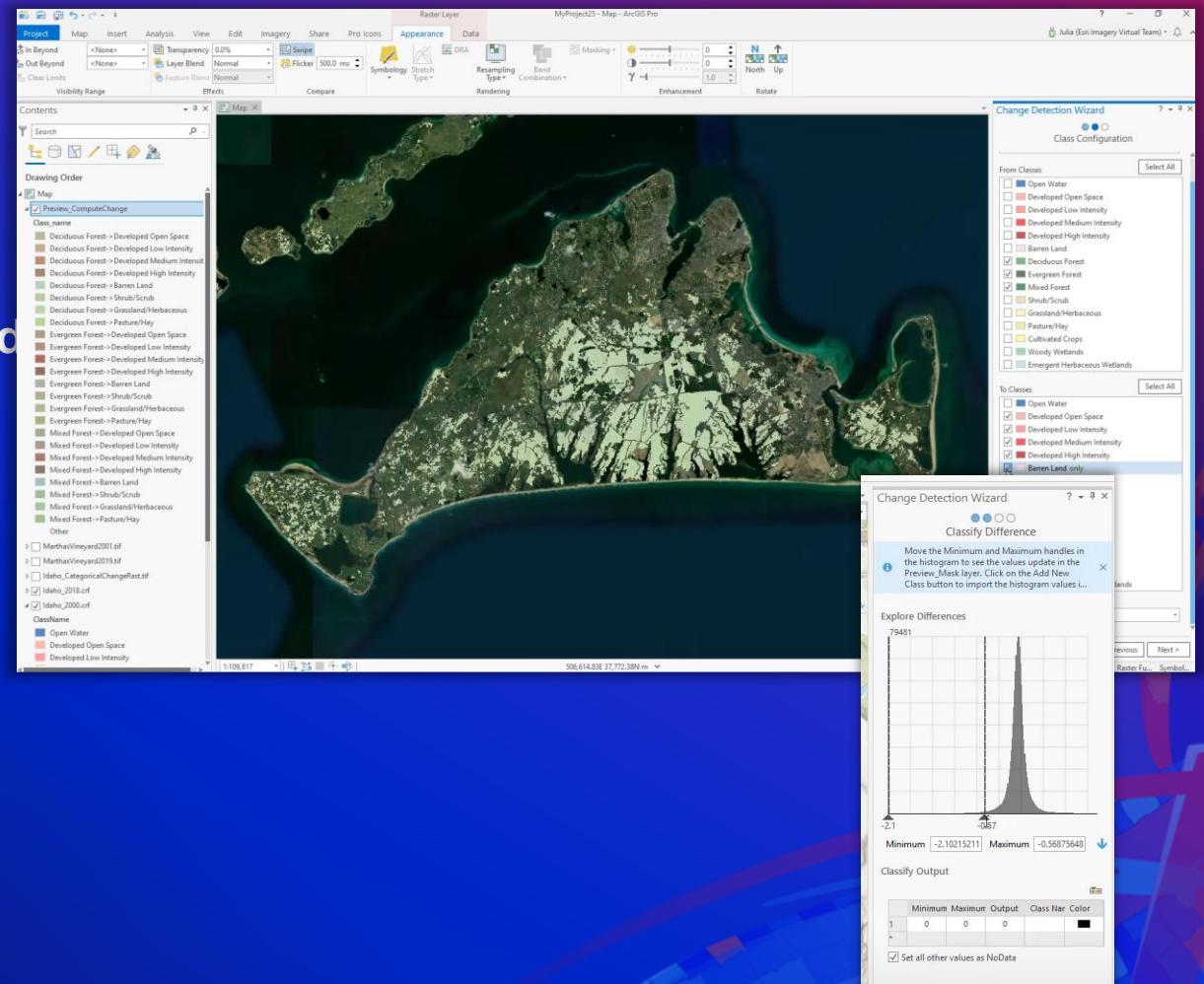
- Change detection wizard
- LandTindr change detection



Change Detection

Change detection wizard

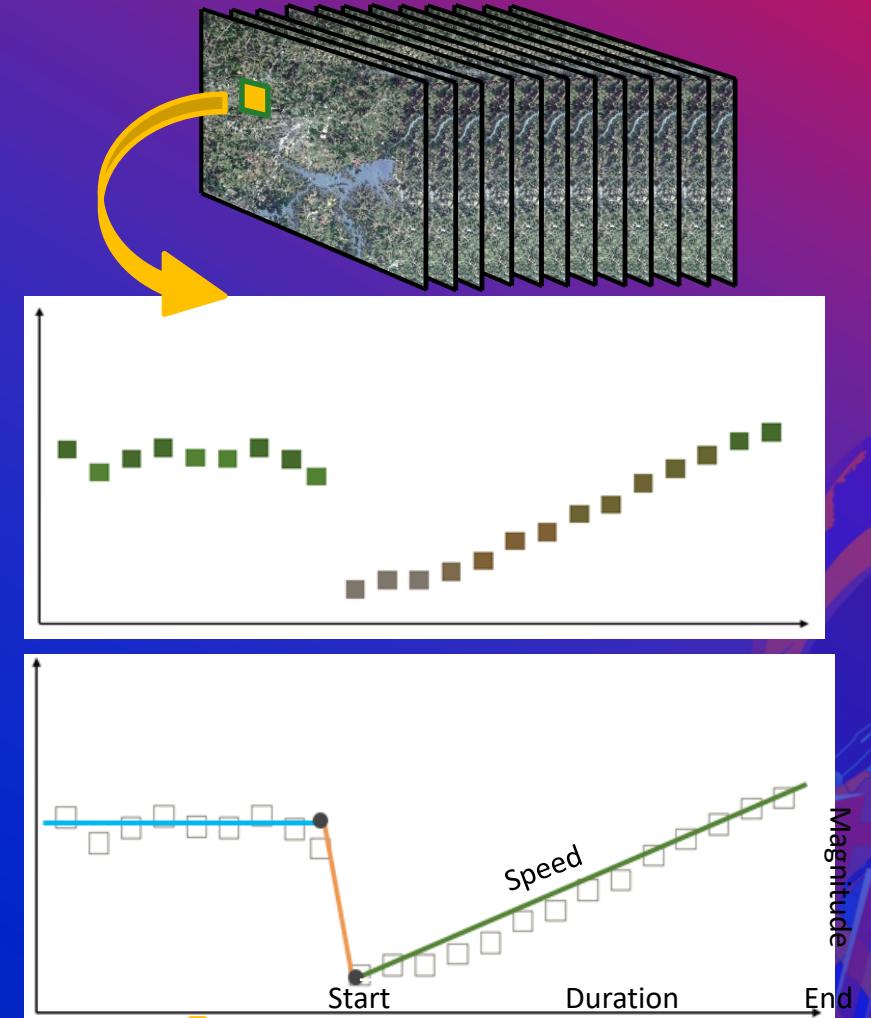
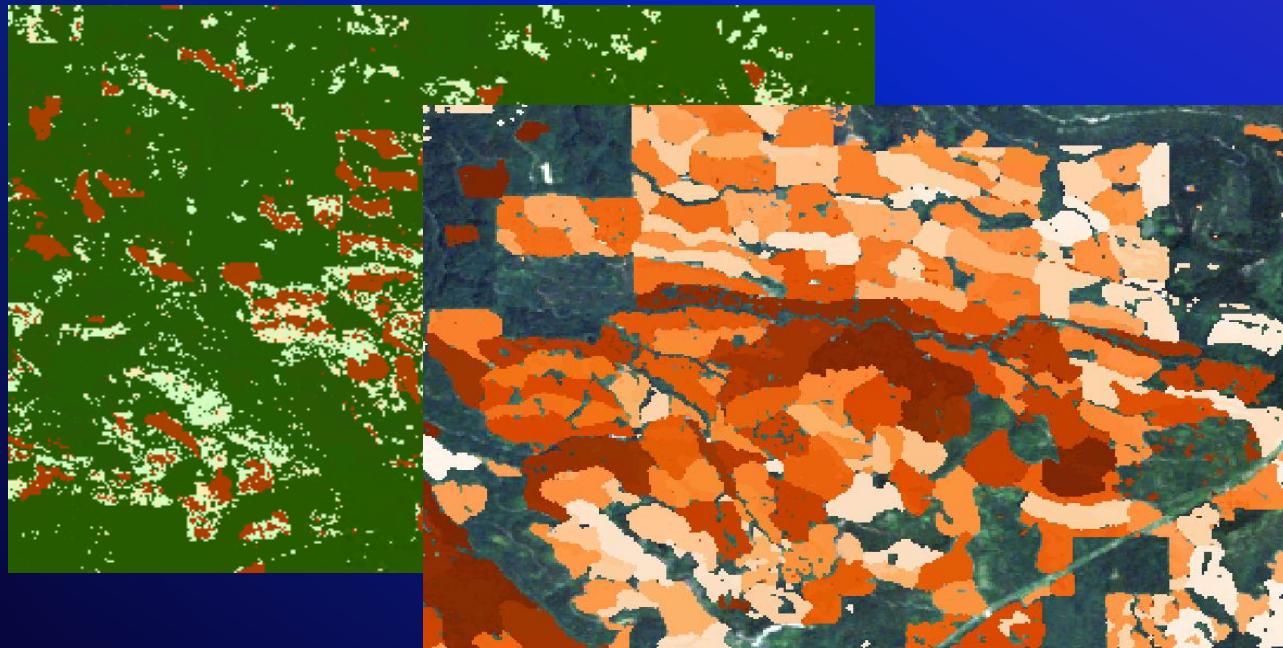
- **Guided workflows for:**
 - Categorical change detection
 - Time-series change detection
 - Pixel-value change detection
- **Includes predefined differences based on industry**
- **Postprocessing cleanup tools**
 - Remove noise (neighborhood smoothing)
- **Ability to write out**
 - Function rasters
 - Raster function template
 - Persisted dataset
 - Feature class



Change Detection

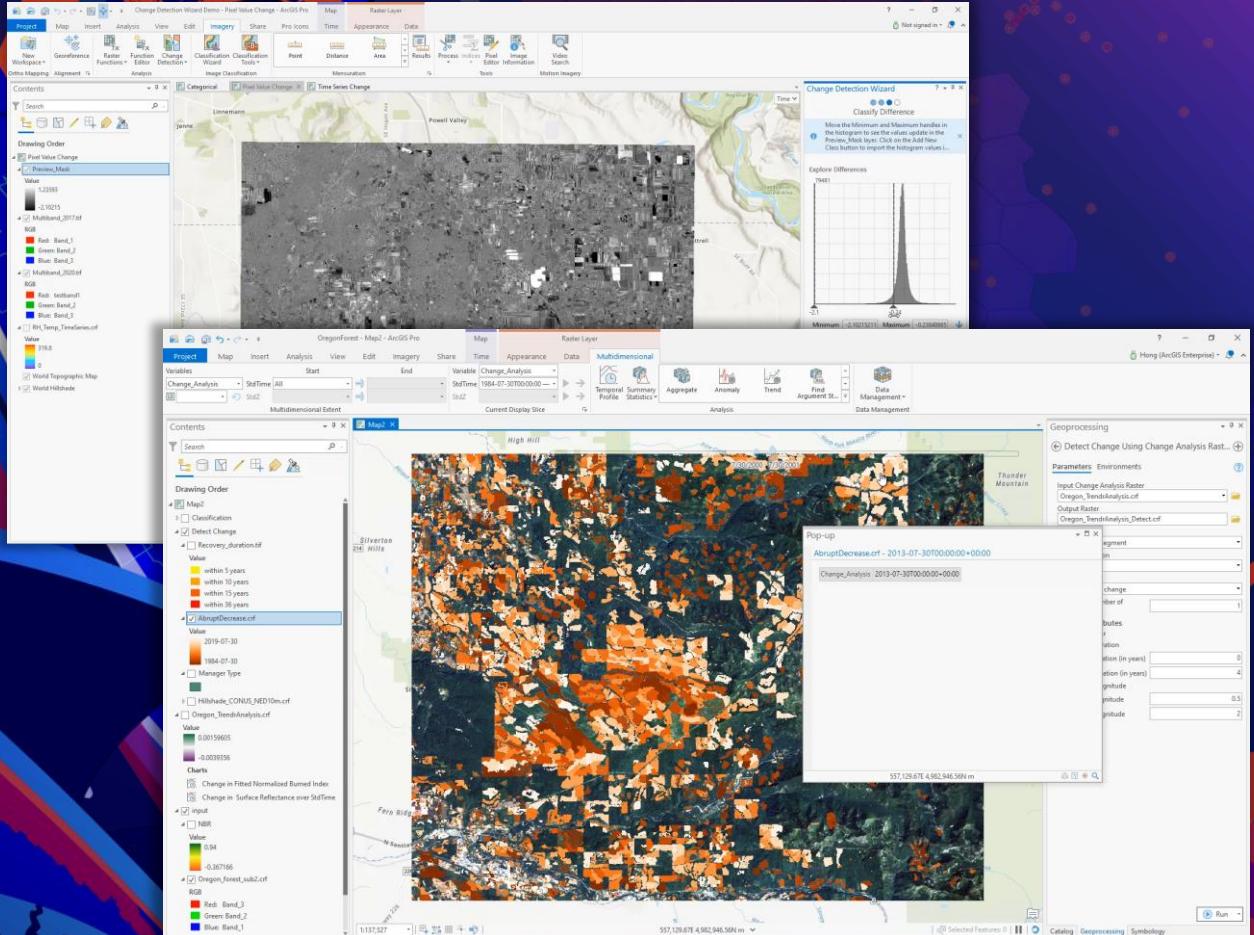
Landsat based detection of trends in disturbance and recovery

- **New *Analyze Changes Using LandTrendr* tool**
- **Enhanced *Detect Change Using Change Analysis Raster* tool**



Demo – Change Detection

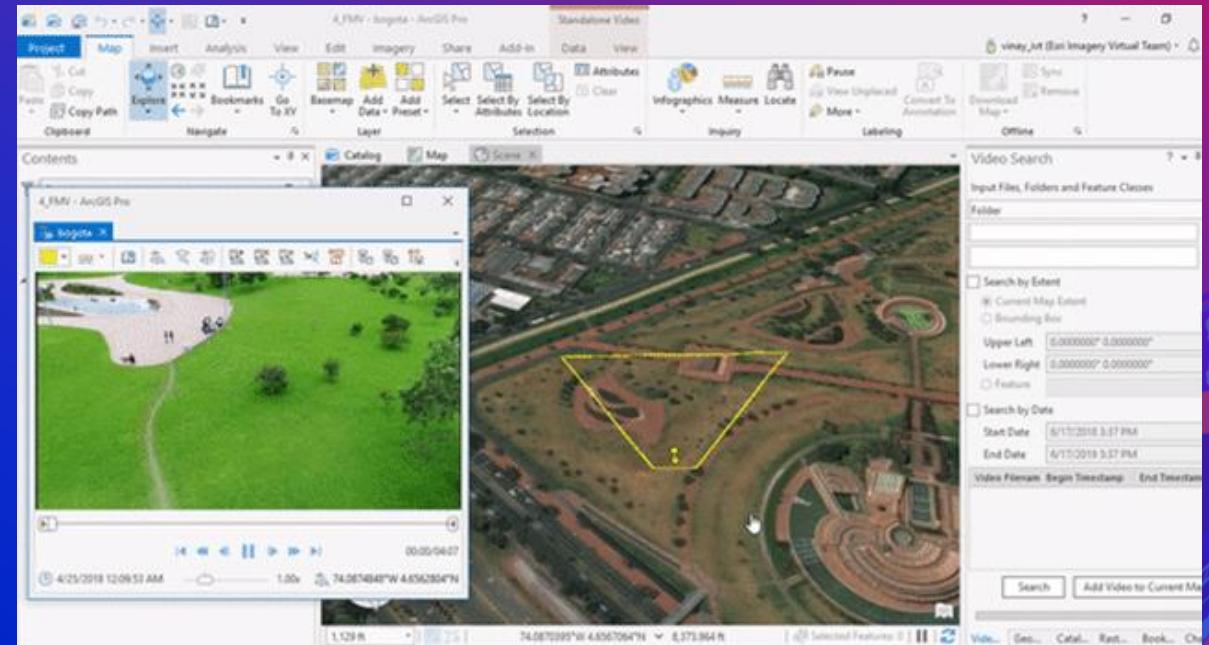
Vinay Viswambrahan



Motion Imagery

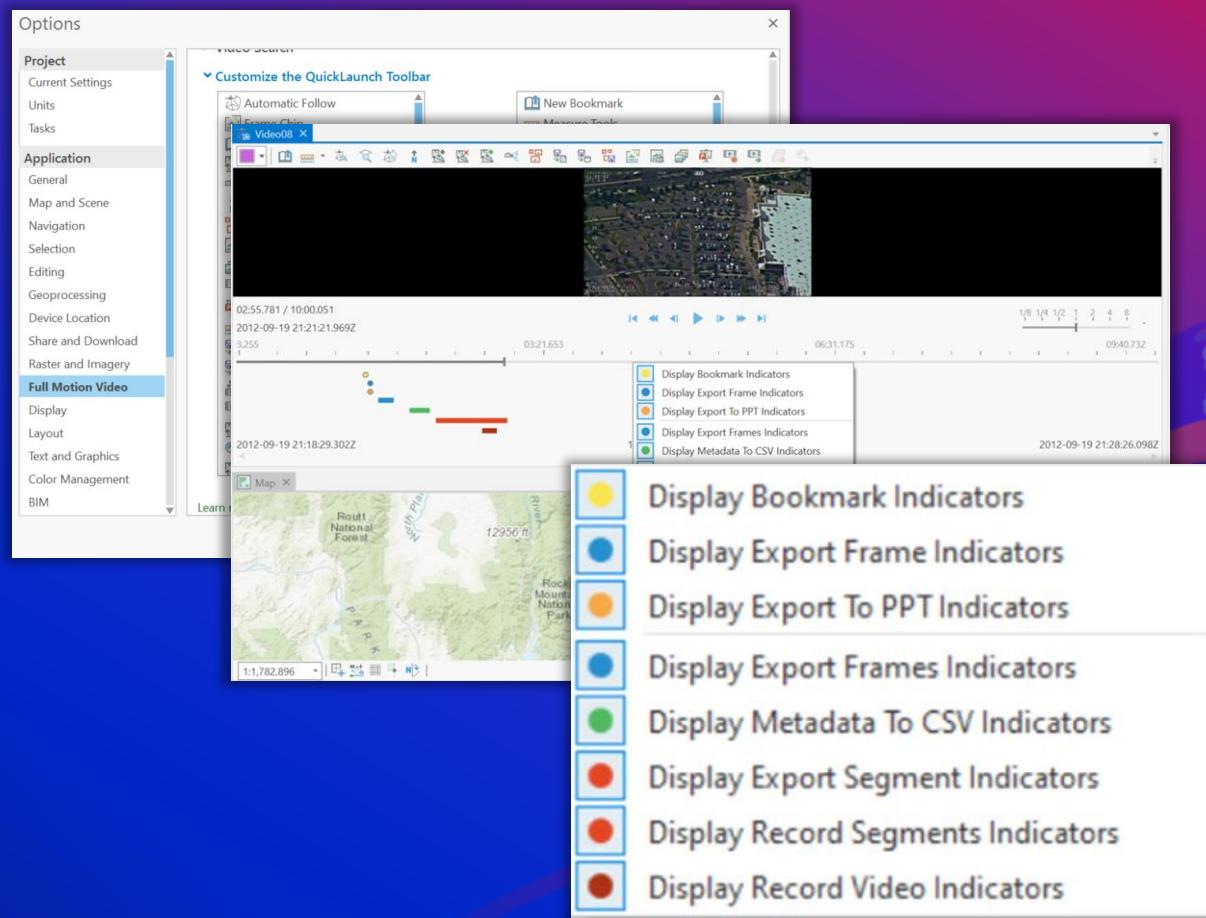
Make informed, timely decisions

- **Integrate**
 - Seamlessly view and capture features from your videos and maps
- **Reporting**
 - Efficient reporting capability for decision makers
- **Compatible**
 - Directly use video with MISB-formatted metadata, or use the Multiplexer to convert video to MISB compatible format



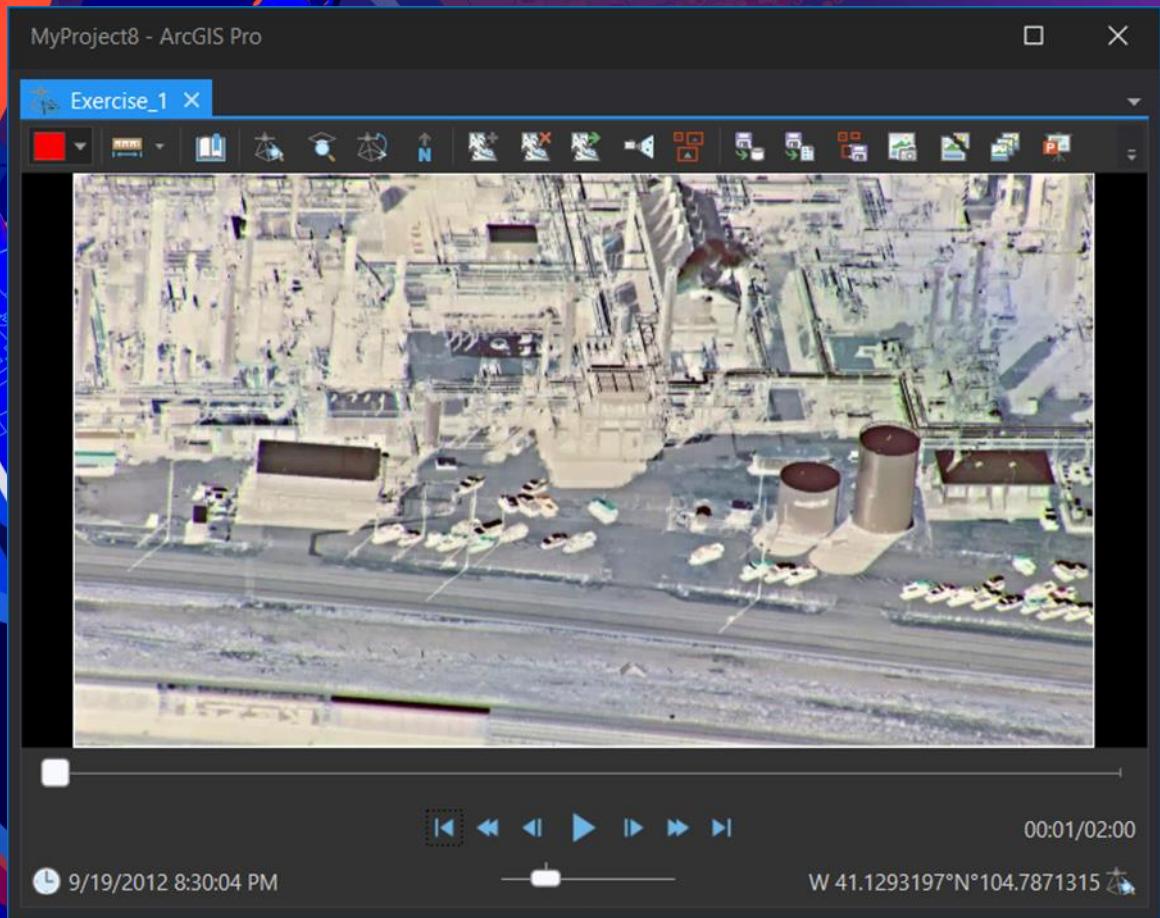
Motion Imagery Enhancements 2.7

- Configurable Controls for Video Player Window
- Interactive Time Slider showing where bookmarks, snaps, recordings, and features were created
- Jump to Frame to easily go to an exact frame in a video
- Frame Dropping Support
 - Better support for > 30 fps



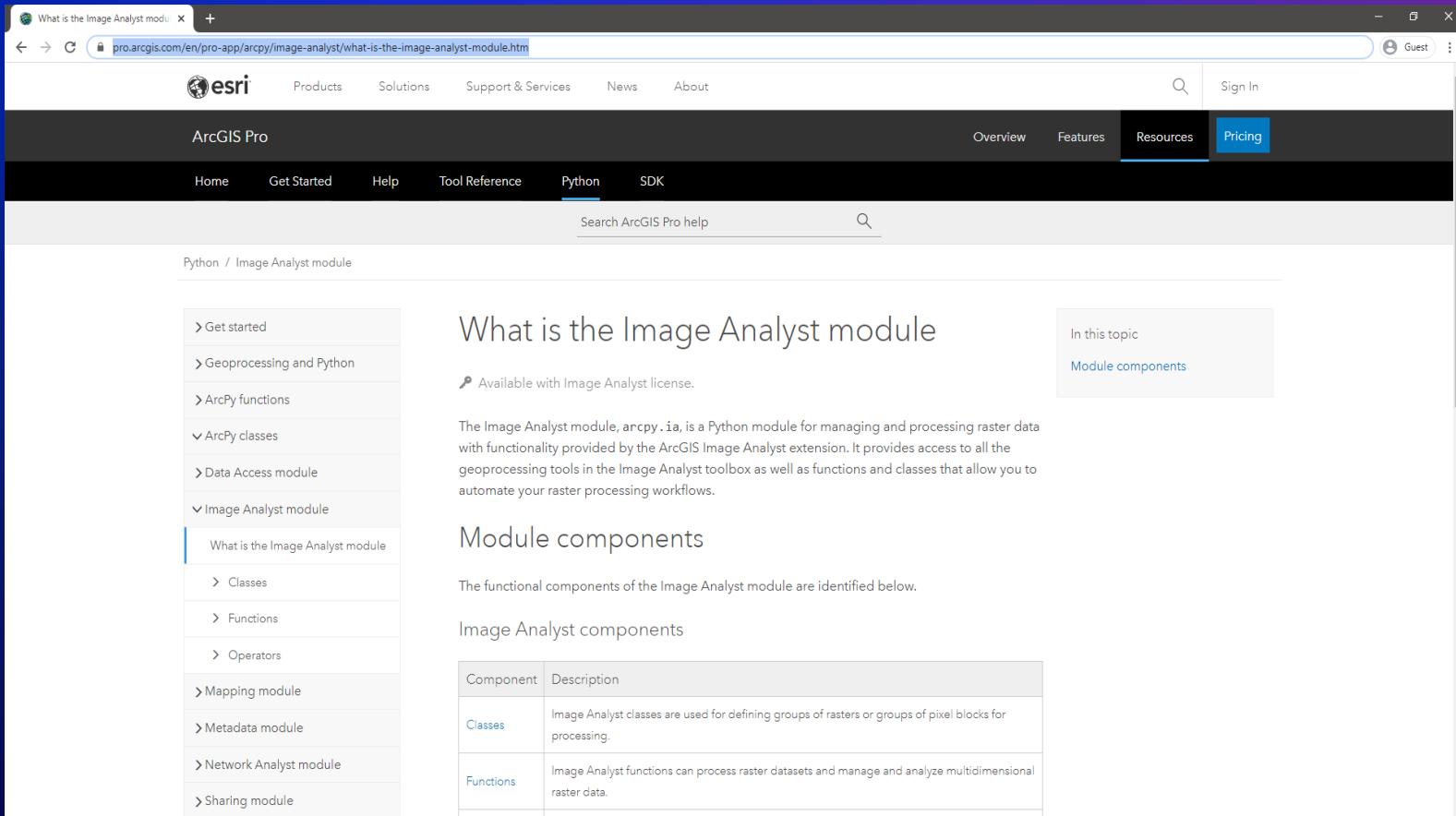
Demo – Motion Imagery

Kyle Talbot



ArcPy APIs in ArcGIS Pro 2.6

- **66 new ArcPy APIs were added to the `arcpy.ia` module**
 - 6 are specifically licensed to Image Analyst
 - 60 are licensed across basic, standard, and advanced
 - <https://pro.arcgis.com/en/pro-app/arcpy/image-analyst/what-is-the-image-analyst-module.htm>

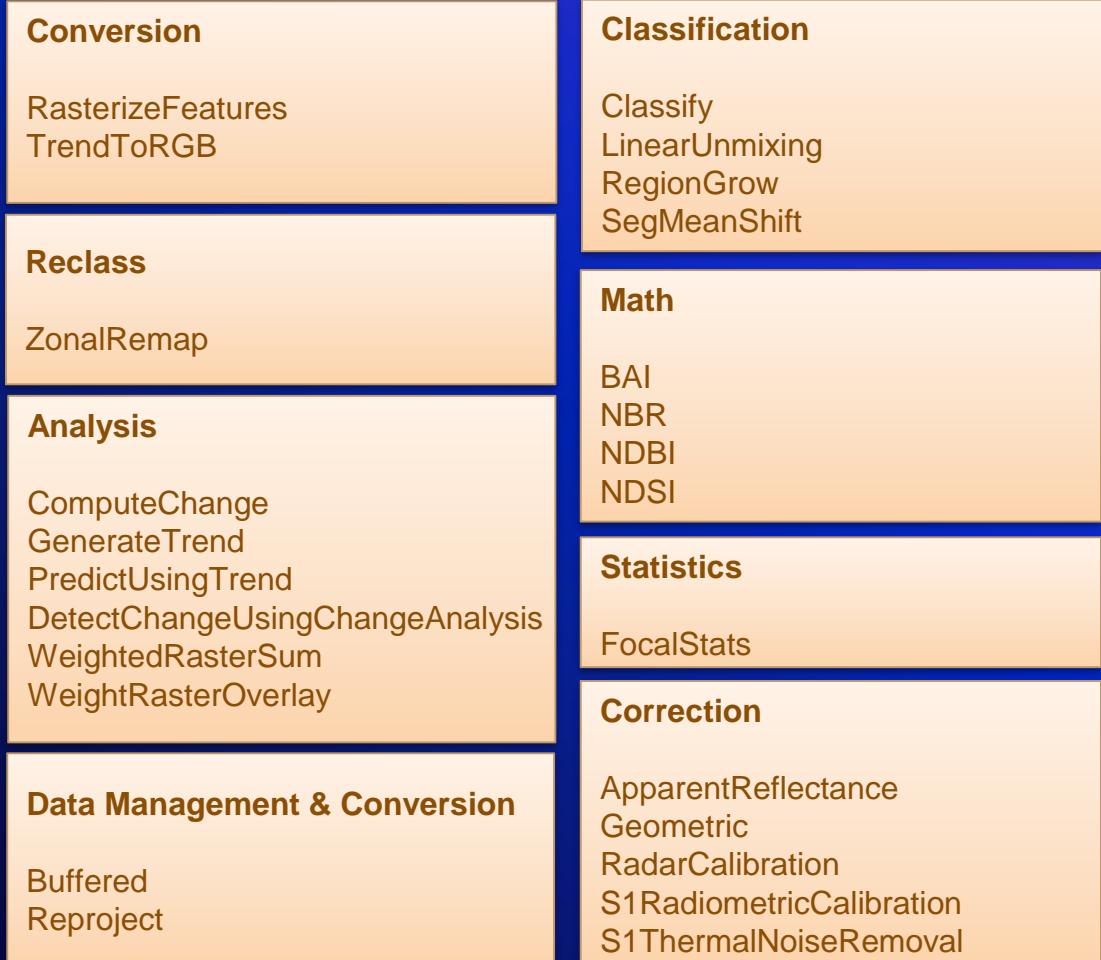


The screenshot shows a web browser displaying the ArcGIS Pro help documentation for the Image Analyst module. The URL in the address bar is pro.arcgis.com/en/pro-app/arcpy/image-analyst/what-is-the-image-analyst-module.htm. The page is titled "What is the Image Analyst module". A sidebar on the left shows a navigation tree under "Python / Image Analyst module", with "Image Analyst module" expanded to show "What is the Image Analyst module", "Classes", "Functions", and "Operators". The main content area describes the Image Analyst module as a Python module for managing and processing raster data, available with the Image Analyst license. It provides access to all the geoprocessing tools in the Image Analyst toolbox and functions and classes for automating raster processing workflows. Below this, a section titled "Module components" lists "Image Analyst components" with a table mapping components to descriptions. The table has two rows:

Component	Description
Classes	Image Analyst classes are used for defining groups of rasters or groups of pixel blocks for processing.
Functions	Image Analyst functions can process raster datasets and manage and analyze multidimensional raster data.

ArcPy Functions for Raster Analysis in ArcGIS Pro 2.7

New raster functions



3. Linear Spectral Unmixing Analysis

```
In [10]: def LinearUnmixingFunc(item):  
    raster = item["Raster"]  
    # get each raster band  
    raster_bands = [raster[i].read() for raster in raster.getRasterBands()]  
    bandCount = len(raster_bands)  
    # calculate log(each_raster_band)  
    for i in range(bandCount):  
        raster_bands.append(arcpy.ia.ln(raster_bands[i]))  
  
    # calculate each_raster_band*log(each_raster_band)  
    for i in range(bandCount):  
        raster_bands.append(raster_bands[i]*raster_bands[i+bandCount])  
  
    # calculate each_raster_band* each_raster_bandj  
    for i in range(bandCount):  
        for j in range(i+1, bandCount):  
            raster_bands.append(raster_bands[i]*raster_bands[j])  
  
    # calculate log(each_raster_bandj)* log(each_raster_bandj)  
    for i in range(bandCount, 2*bandCount):  
        for j in range(i+1, 2*bandCount):  
            raster_bands.append(raster_bands[i]*raster_bands[j])  
  
    # calculate (each_raster_bandj - each_raster_bandi)/(each_raster_bandi + each_raster_bandj)  
    for i in range(bandCount):  
        for j in range(i+1, bandCount):  
            raster_bands.append(1.0*(raster_bands[j]-raster_bands[i])/(raster_bands[i]+raster_bands[j]))  
  
    # raster bands  
    composite_raster = arcpy.ia.CompositeBand(raster_bands)  
    # edc file  
    spectral_profile = r"\\chang\\dataCube\\fraction_cover_spectral_profile.json"  
    unmixing_raster = arcpy.ia.linearUnmixing(composite_raster, spectral_profile, sum_to_one=True, non_negative=True)  
    return {"Raster": unmixing_raster, "Name": "unmixingoutput", "StartTime": item["StdTime"]}
```

3. Render Trend Output

```
In [7]: # render the trend as RGB  
from arcgis.raster.functions import trend_to_rgb  
trend_in_rgb = trend_to_rgb(nightlight_linear_trend, model_type="LINEAR")  
render_trend_in_rgb = stretch(trend_in_rgb, "MinMax", [0,0.0005,0,0.0002], [0,20,0.3332,2.934], [0,0.0005,0,0.0001])  
render_trend_in_rgb
```

Out [7]:

Demo – arcpy.ia module

Explore NDVI data through time

jupyter Explore a Time Series of NDVI Last Checkpoint: 11/12/2019 (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Logout

Not Trusted Python 3

visualize Raster item

In [6]: # get the raster within the first item in the collection
first_item_raster = rc2[0]['Raster']
rendered_raster = arcpy.ia.Render(first_item_raster, rendering_rule = {"bands": [4, 3, 2], "min":5000, "max":15000})
rendered_raster.exportImage(width = 400)

Out[6]:



3. Remove Cloud and Shadow

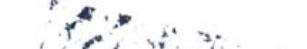
In [7]: # only keep those clean pixels using the quality band
def clean(item):
 raster = item['Raster']
 quality_raster = raster.getRasterBands(12)
 mask = arcpy.ia.Apply(raster == 2720) | (quality_raster == 2724) | (quality_raster == 2728) | (quality_raster == 27)
 masked_clean_band_list = []
 for bandRaster in raster.getRasterBands():
 arcpy.ia.Com is the gp tool, will persist data
 masked_clean_band = arcpy.ia.CompositeBand(mask, bandRaster, 0), {"Local": {"Operation": 76}} # "Value <> 66 AND Value <> 130"
 masked_clean_band_list.append(masked_clean_band)
 masked_clean_raster = arcpy.ia.CompositeBand(masked_clean_band_list)
 return {"raster": masked_clean_raster, "Name": item["Name"], "AcquisitionDate": item["AcquisitionDate"]}

In [8]: cleaned_rc = rc2.map(clean)

Visualize

In [9]: # get the raster within the 2nd item in the collection
first_item_raster = cleaned_rc[0]['Raster']
rendered_raster = arcpy.ia.Render(first_item_raster, rendering_rule = {"bands": [4, 3, 2], "min":6000, "max":15000})
rendered_raster.exportImage(width = 400)

Out[9]:





esri®

THE
SCIENCE
OF
WHERE®



Presentation Title

Presenter Names

2021 ESRI FEDERAL GIS CONFERENCE



esri®

THE
SCIENCE
OF
WHERE®



Presentation Title

Presenter Names

2021 ESRI FEDERAL GIS CONFERENCE

Headline

Subhead

- Bullets

Headline

Subhead

- Bullets

Section Header

Subhead



Headline

Subhead



esri®

THE
SCIENCE
OF
WHERE®