

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



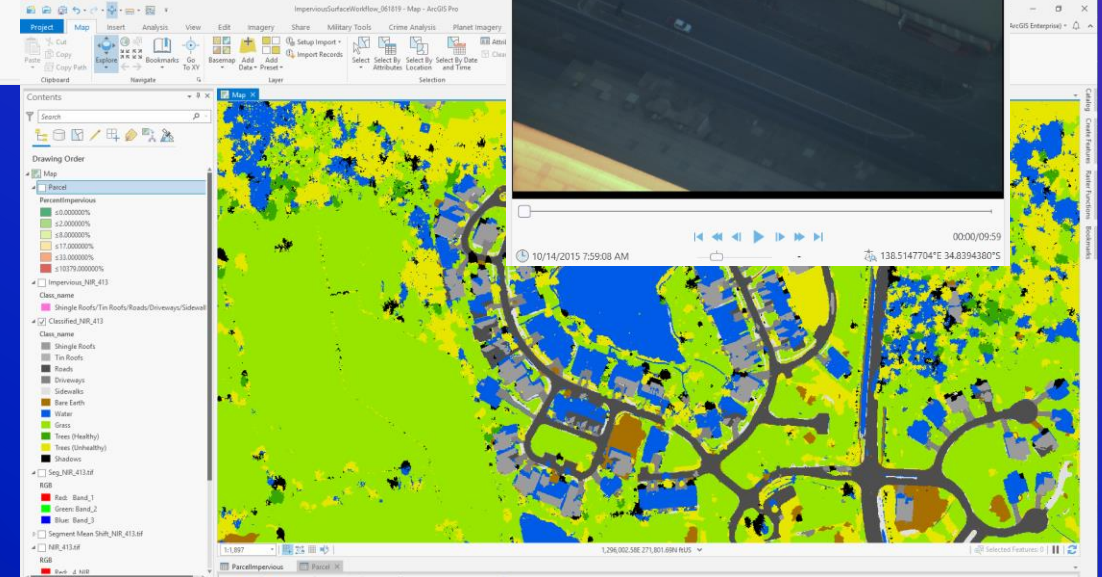
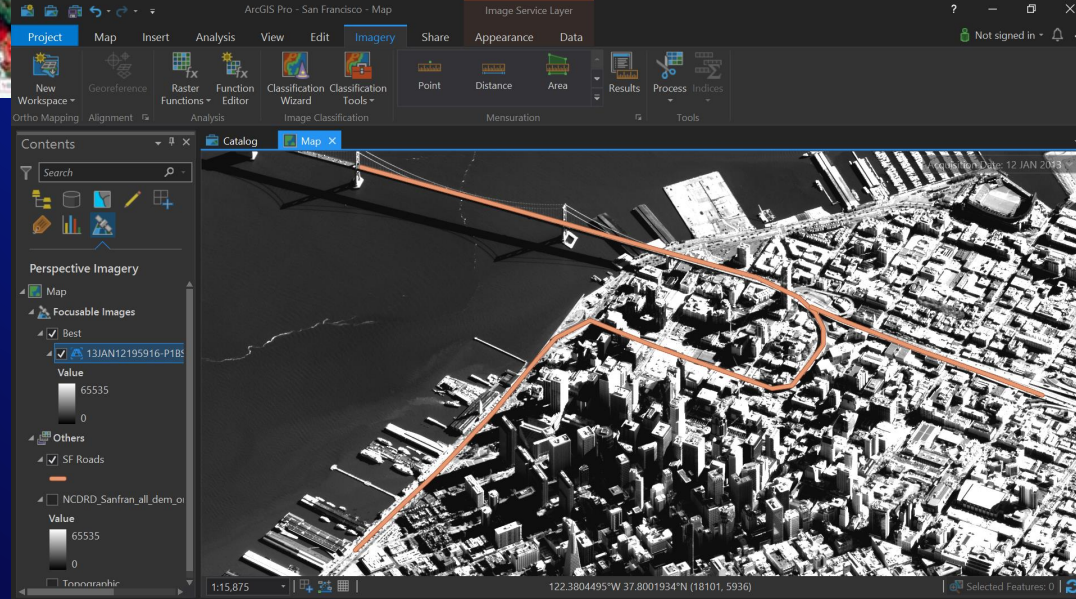
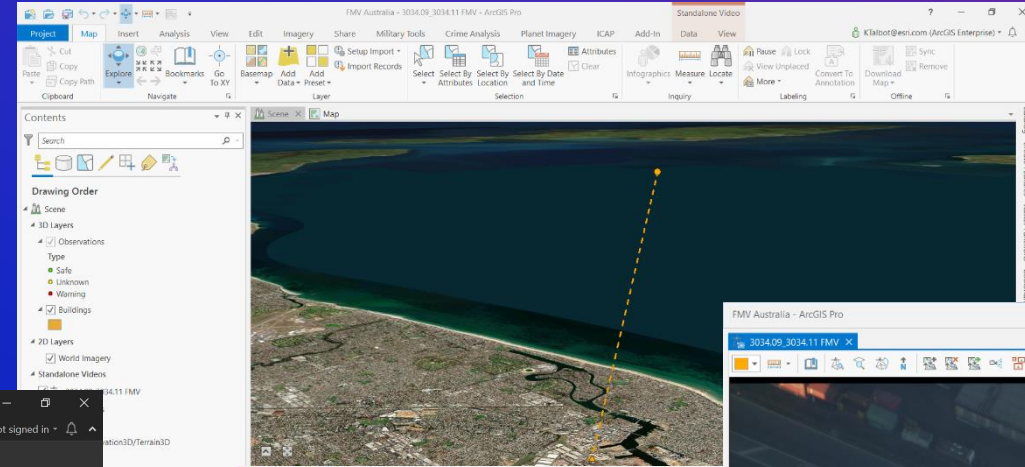
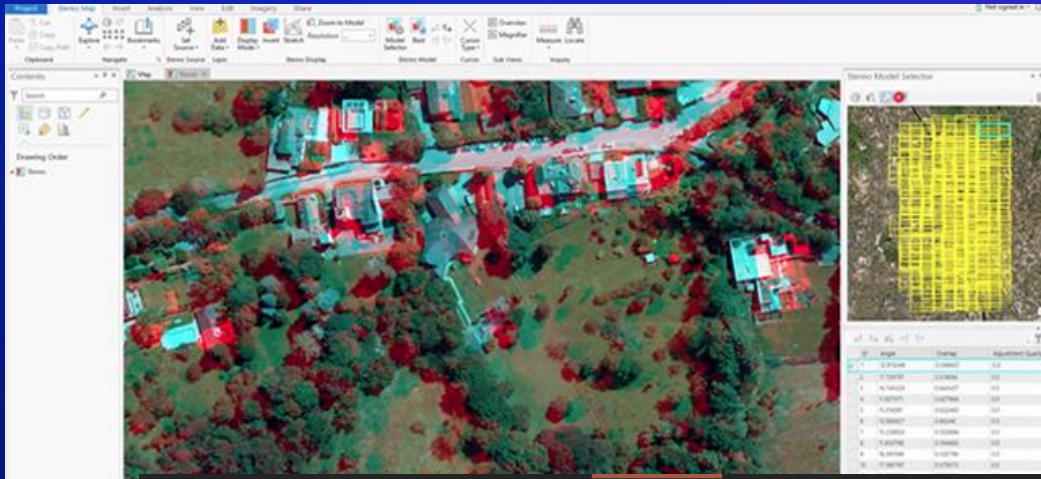


Image Analyst Extension

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**

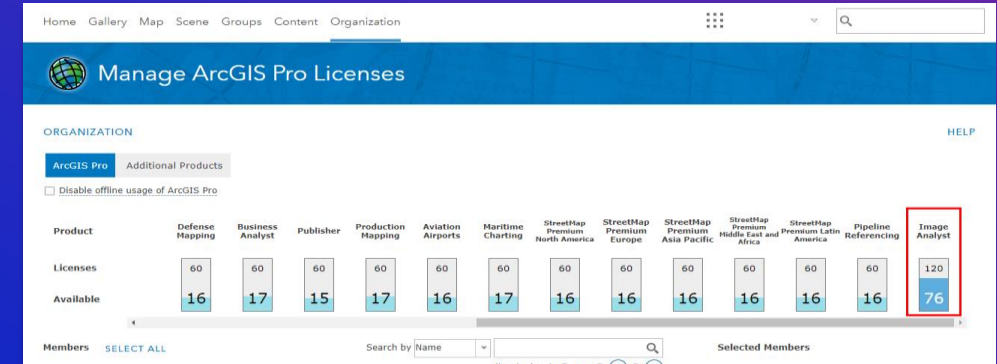
- the same as the Spatial Analyst extension

- **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!



Home Gallery Map Scene Groups Content Organization

Manage ArcGIS Pro Licenses

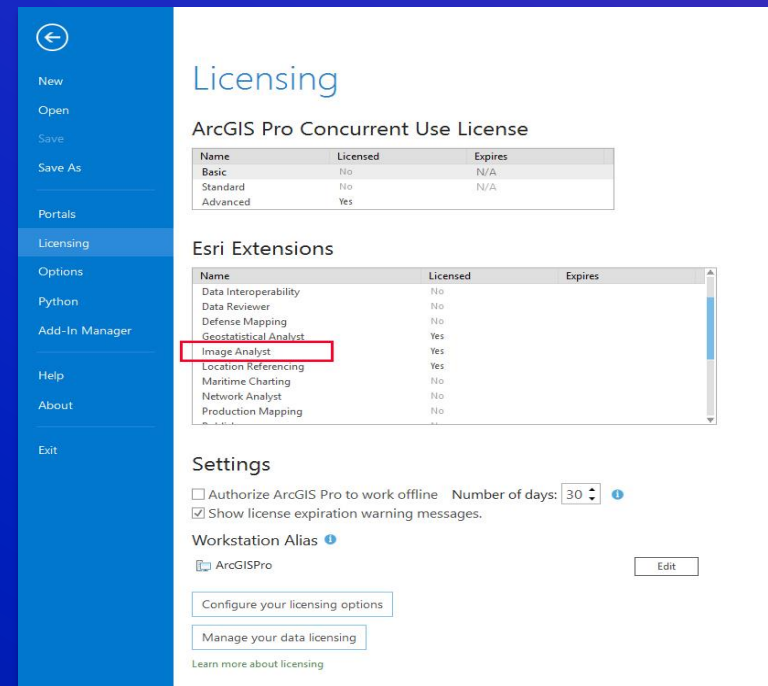
ORGANIZATION

ArcGIS Pro Additional Products

☐ Disable offline usage of ArcGIS Pro

Product	Defense Mapping	Business Analyst	Publisher	Production Mapping	Aviation Airports	Maritime Charting	StreetMap Premium North America	StreetMap Premium Europe	StreetMap Premium Asia Pacific	StreetMap Premium Middle East and Africa	StreetMap Premium Latin America	Pipeline Referencing	Image Analyst
Licenses	60	60	60	60	60	60	60	60	60	60	60	60	120
Available	16	17	15	17	16	17	16	16	16	16	16	16	76

Members SELECT ALL Search by Name Selected Members



←

New
Open
Save
Save As
Portals
Licensing
Options
Python
Add-In Manager
Help
About
Exit

Licensing

ArcGIS Pro Concurrent Use License

Name	Licensed	Expires
Basic	No	N/A
Standard	No	N/A
Advanced	Yes	

Esri Extensions

Name	Licensed	Expires
Data Interoperability	No	
Data Reviewer	No	
Defense Mapping	No	
Geostatistical Analyst	Yes	
Image Analyst	Yes	
Location Referencing	Yes	
Maritime Charting	No	
Network Analyst	No	
Production Mapping	No	

Settings

☐ Authorize ArcGIS Pro to work offline Number of days: 30 ⓘ

☒ Show license expiration warning messages.

Workstation Alias ⓘ

ArcGISPro Edit

Configure your licensing options

Manage your data licensing

Learn more about licensing

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

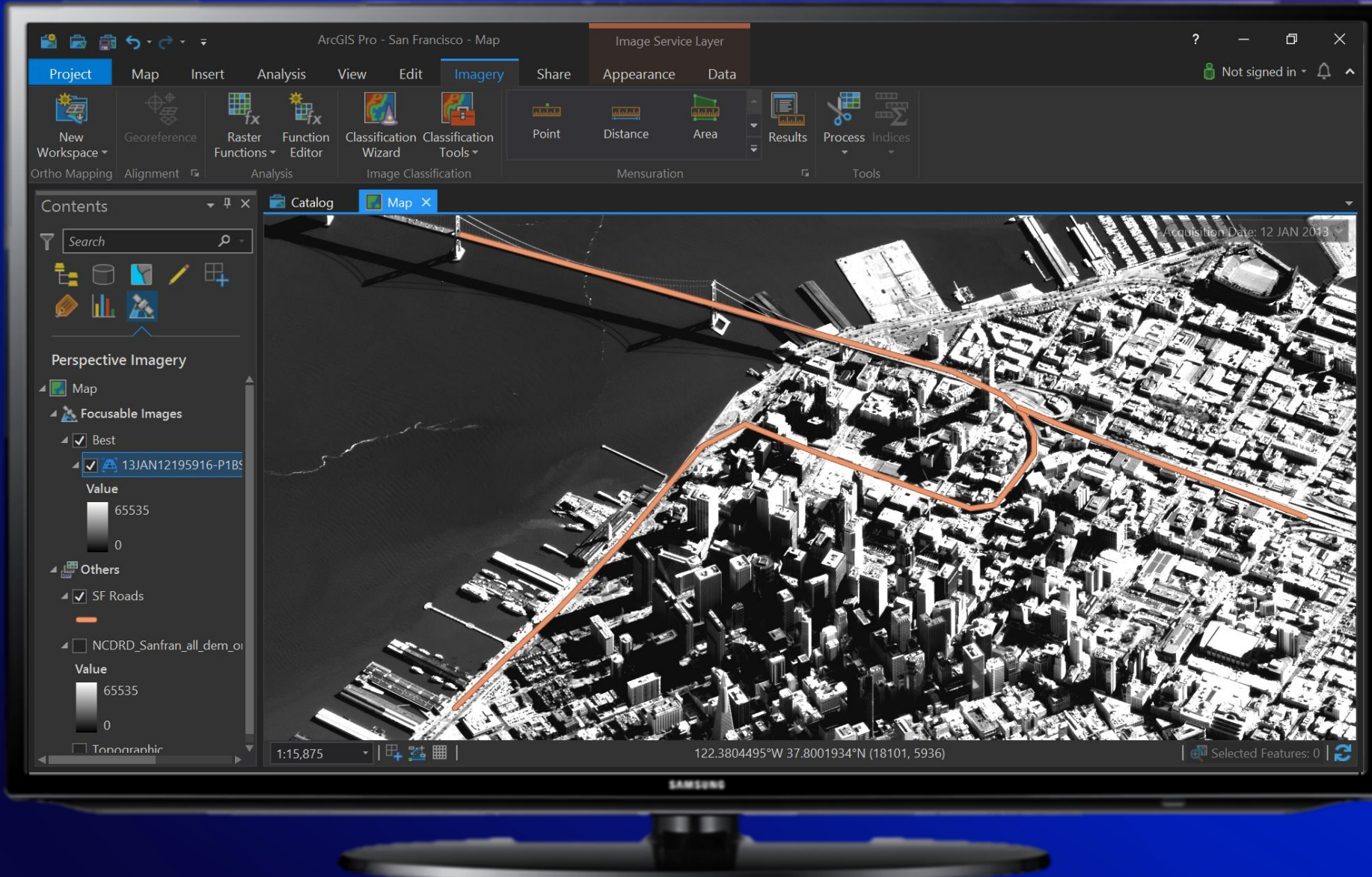
Extract geospatial features from imagery using deep learning techniques such as object detection and image classification / segmentation

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

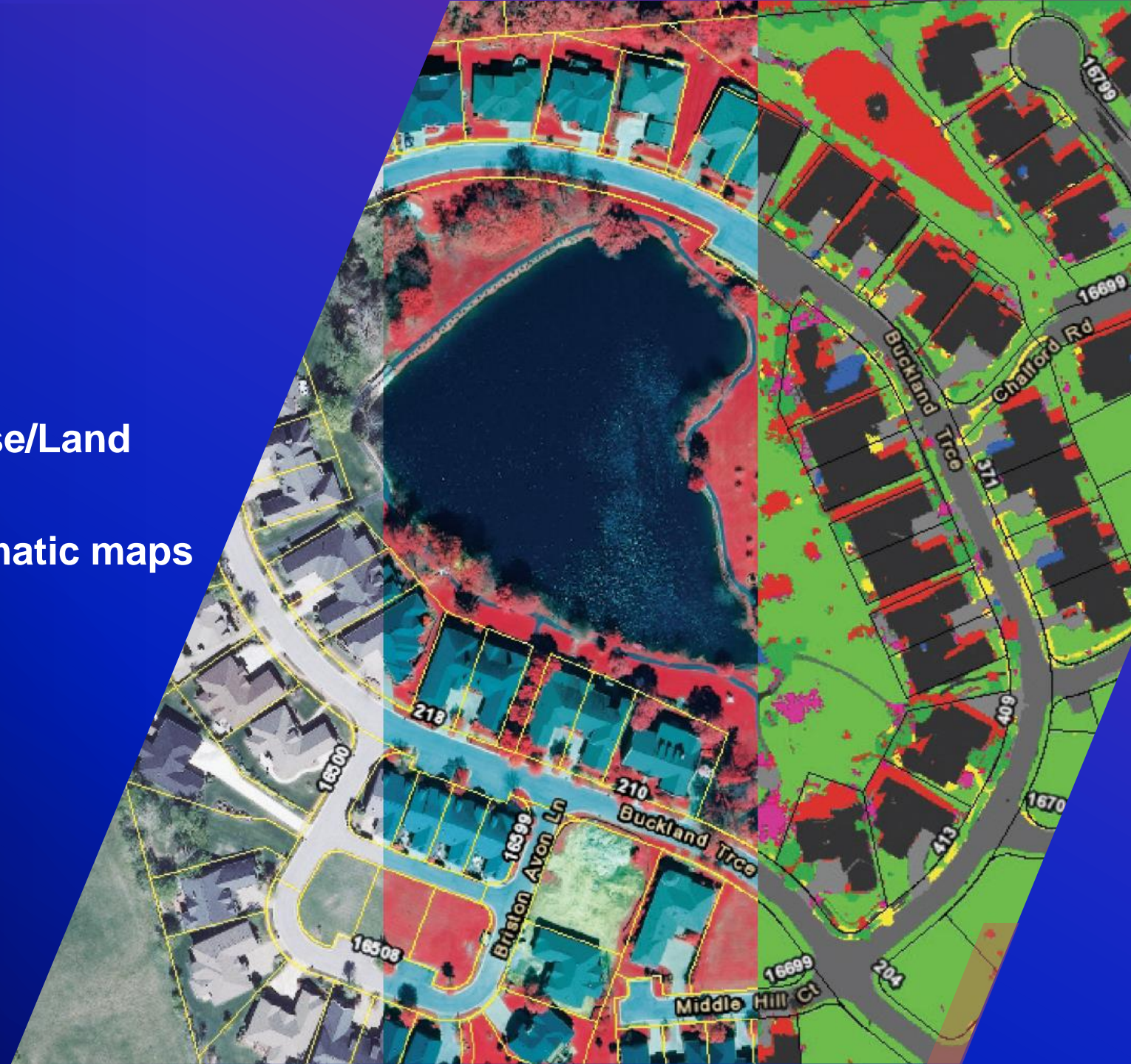


Capabilities:

- 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

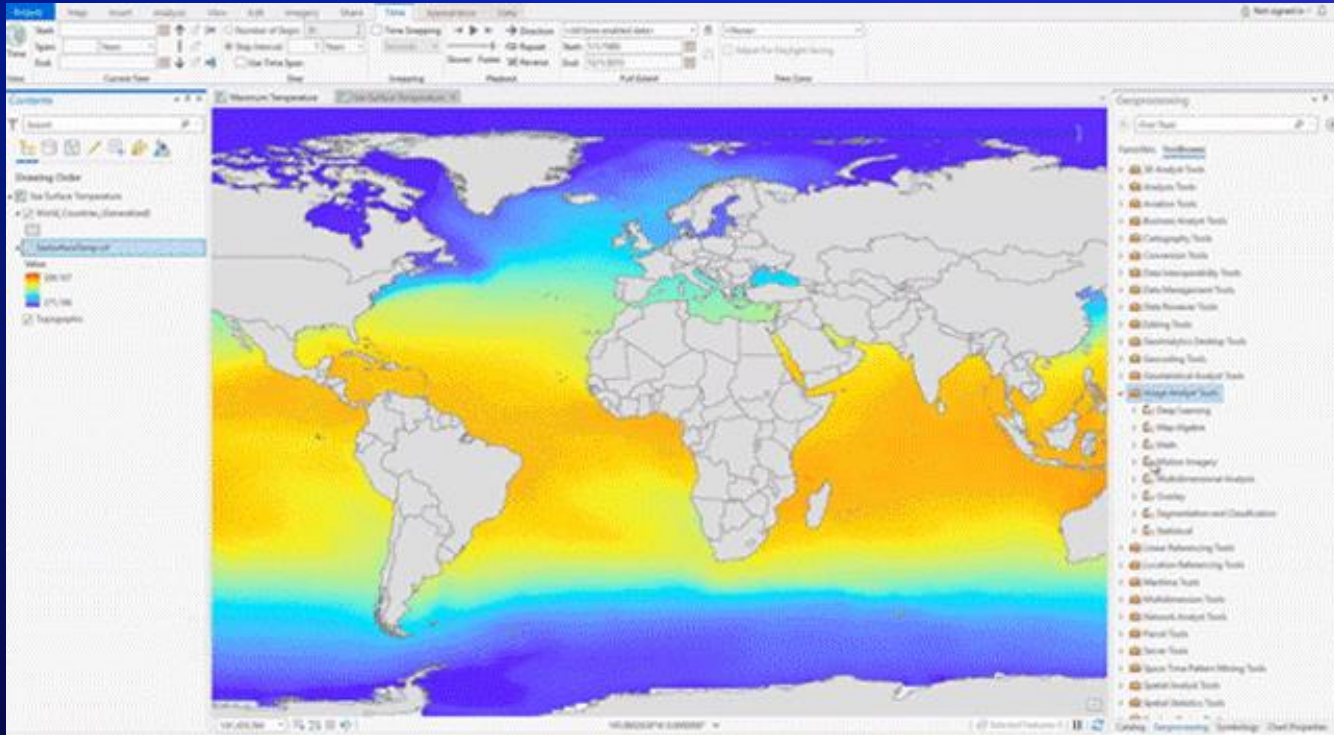


Pixel Editor

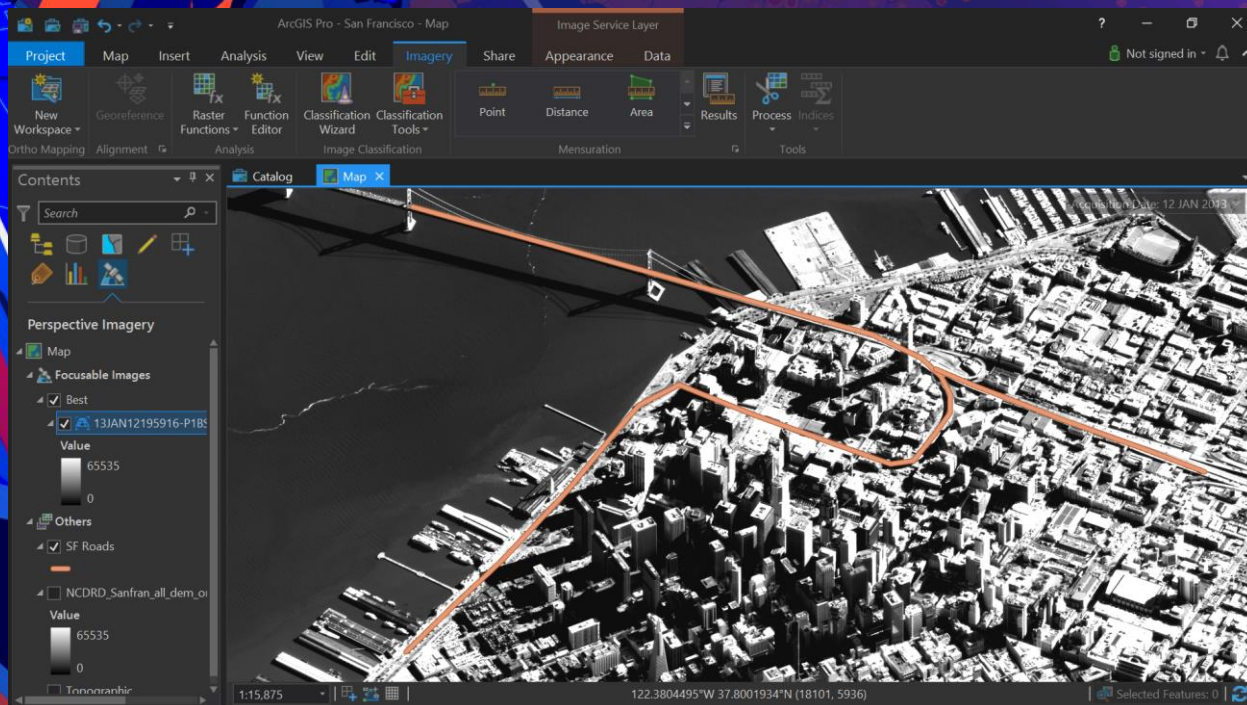
- **Manipulate pixel values for raster and imagery data**
 - Elevation data
 - Reclassify
 - Redaction
 - Replace clouds



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

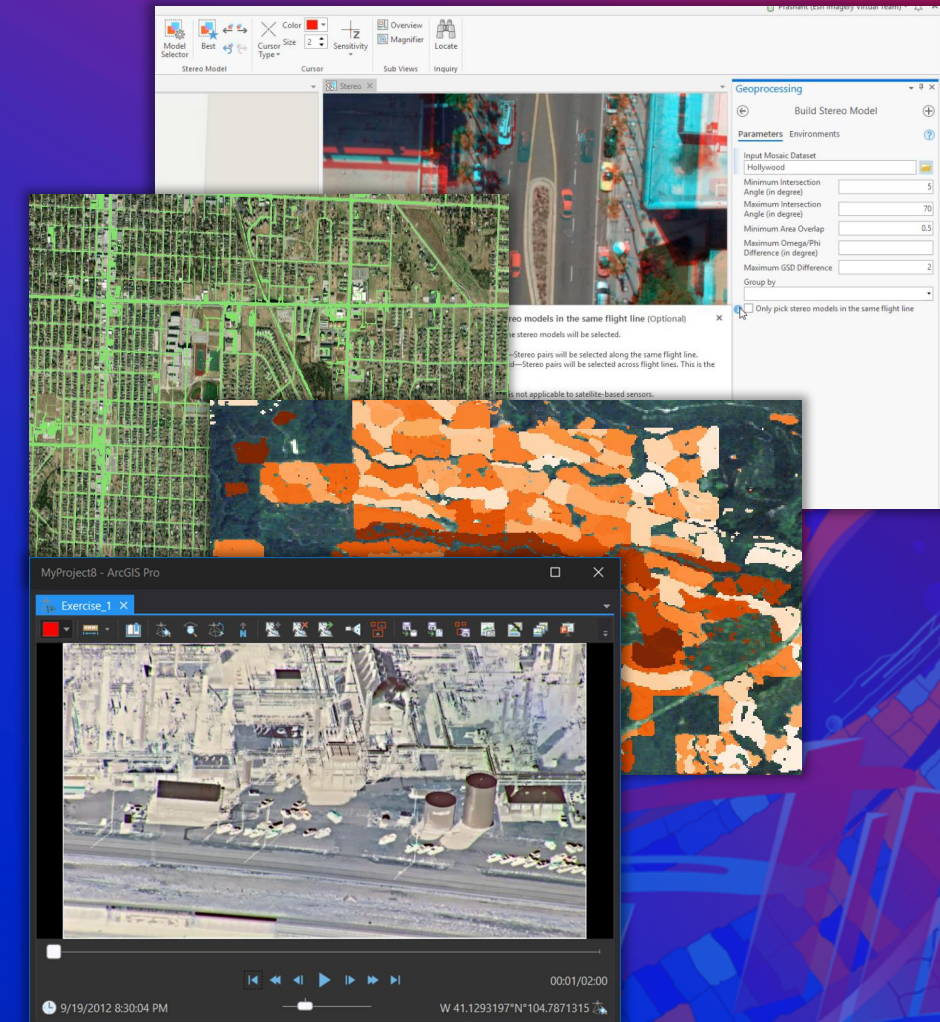
- New wizard and algorithms

Stereo mapping

- User experience improvements when working with stereo collections

Motion imagery

- Time slider improvements
- Application settings
- UX improvements



Deep Learning

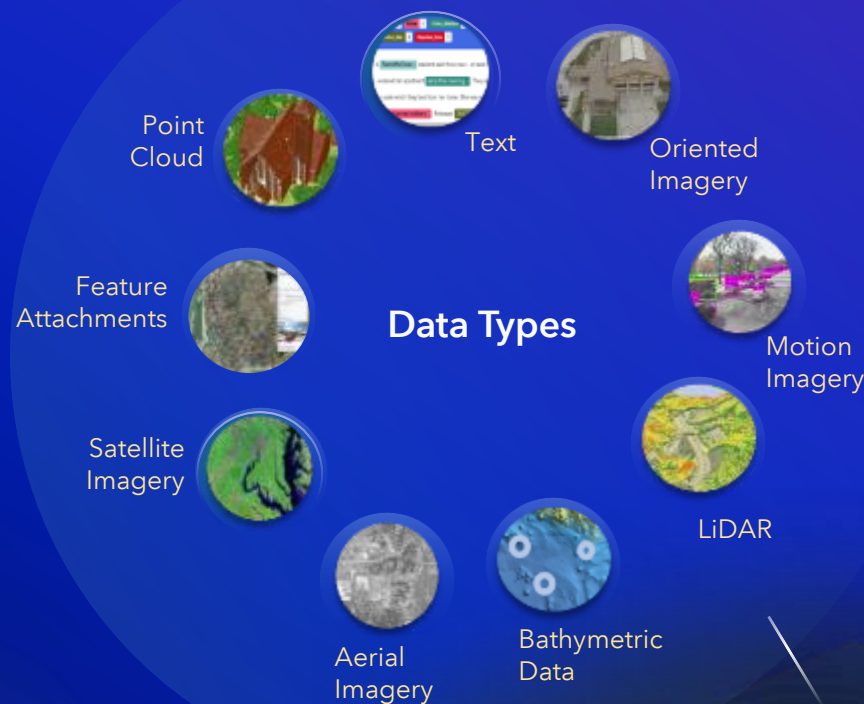
New model and tools

New

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



Integration



Data Types



Tasks

Deep Learning

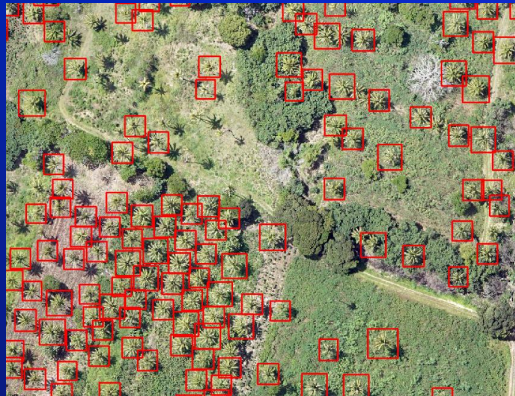
New model types

New Object Detection Models

- YOLOV3
- FasteRCNN

Applications

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

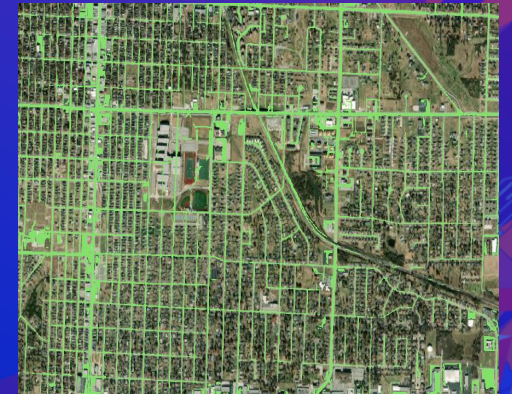


New Pixel Classification Models

- 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

Accuracy Assessment Report

Disclaimer

The confidence range of the detected features is [51.921958, 83.617765]. If features with small confidences were filtered out (e.g., by a threshold in the model inference), caution is advised when comparing the results below to other results based on the full confidence range [0, 100].


Average Precision averaged across these 10 IoU thresholds [0.5, 0.65, ..., 0.95] and all classes

mAP @ IoU [0.5 : 0.95] @ All_Classes = 0.019777.

IoU >= 0.3000

IoU >= 0.3000	Precision	Recall	F1 Score	AP	True Positive	False Positive	False Negative
All Classes	0.7059	0.9259	0.8011	0.8081	24.0000	10.0000	2.0000
1	0.7059	0.9259	0.8011	0.8081	24.0000	10.0000	2.0000

Precision over Recall for IoU >= 0.3000, both in [0..1]



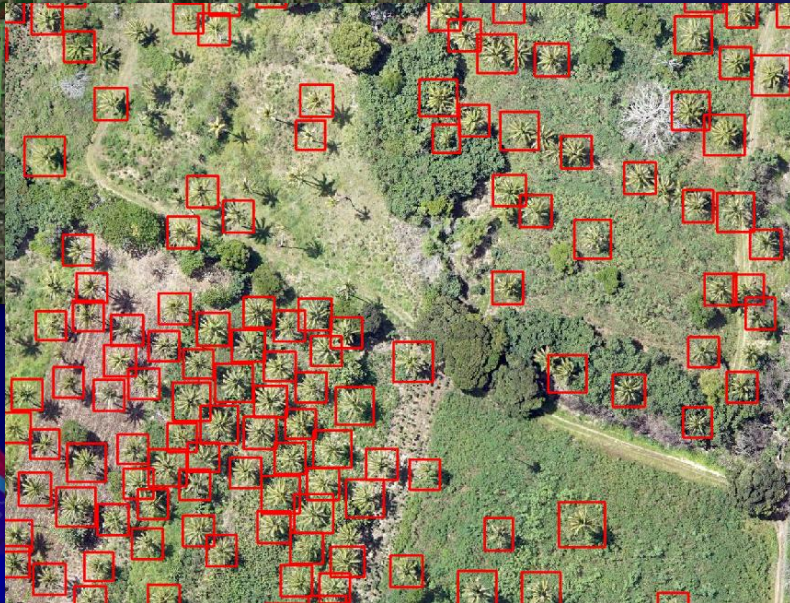
Accuracy
assessment



Multilabel
classification



Multilabel
classification

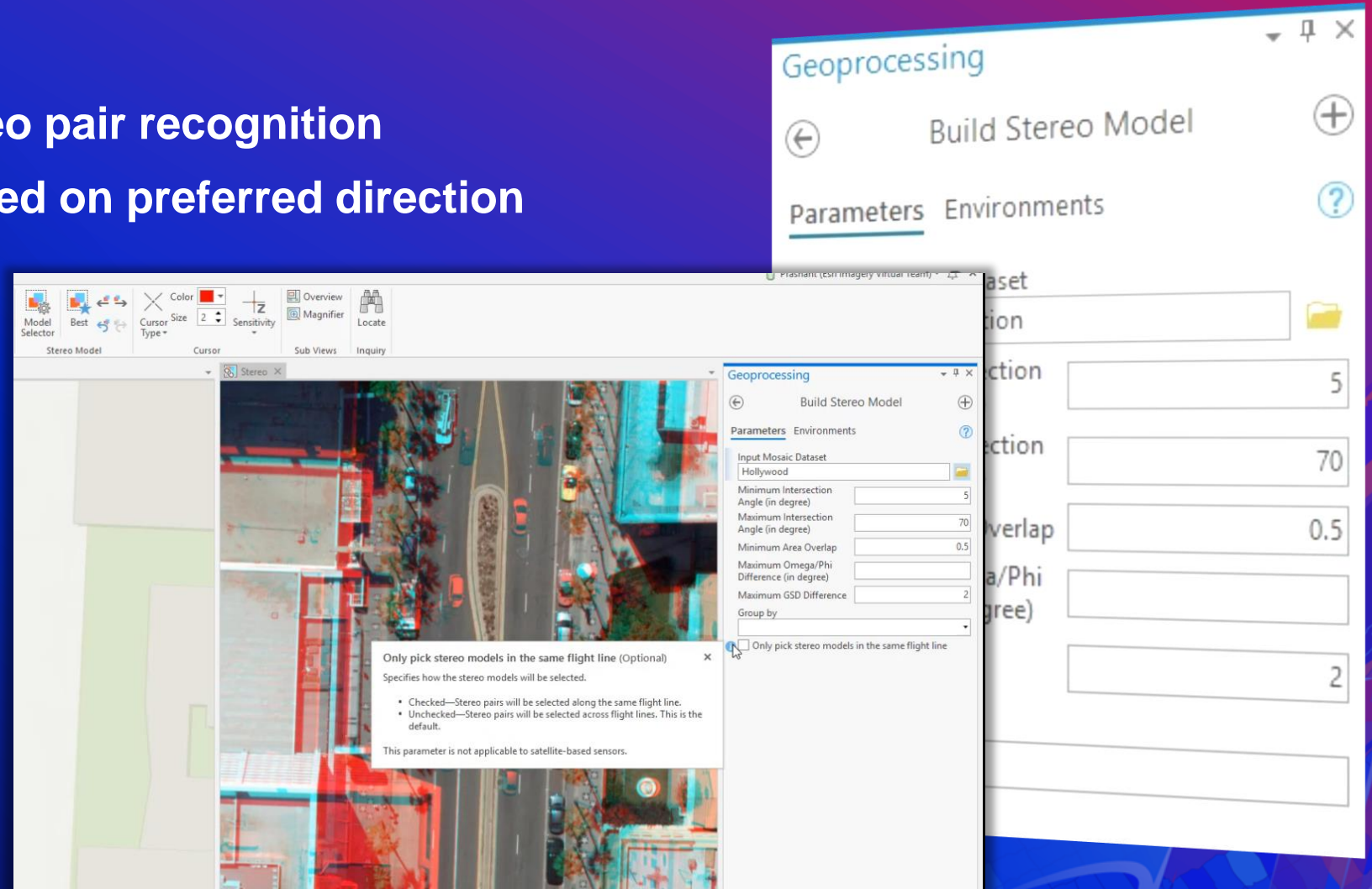


Demo - Deep Learning in ArcGIS Pro

Vinay Viswambrahan

Stereo Mapping

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



The screenshot displays the ArcGIS Pro software interface. The main map area shows a 3D perspective view of a city street with buildings and roads. A 'Map Properties: Stereo' dialog box is open in the center. The 'Stereo' tab is selected in the left sidebar. The 'Stereo Model Collection' section shows the following information:

- Location: C:\Users\pran4761\Documents\ArcGIS\Project\Proj732\StereoDemo\Proj732StereoDemo.gdb
- Dataset: Orlando
- Workspace: File Geodatabase
- Left Image ID: 18_0000138-000290-091314134132-CAM1
- Right Image ID: 17_0000137-000289-091314134130-CAM1

A dropdown menu is open for 'Orient stereo models so that North is', with 'Down or Left' selected. The 'Contents' pane on the left shows the 'Hollywood_Roads' layer. The 'Stereo Model Selector' pane on the right shows a list of stereo models with columns for ID, Angle, Overlap, and Adjustment.

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 LandTendr

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

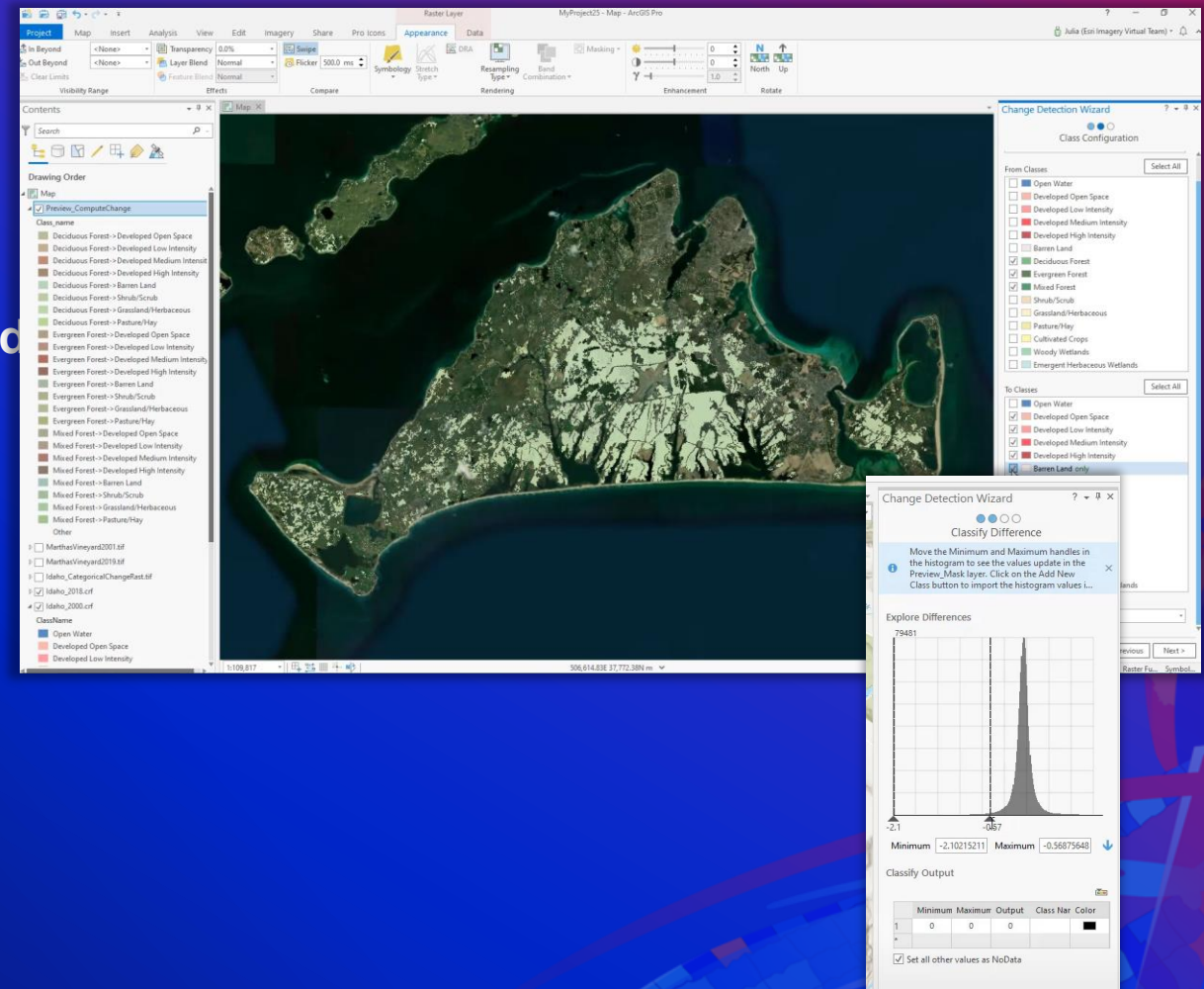
- Change detection wizard
- LandTendr change detection



Change Detection

Change detection wizard

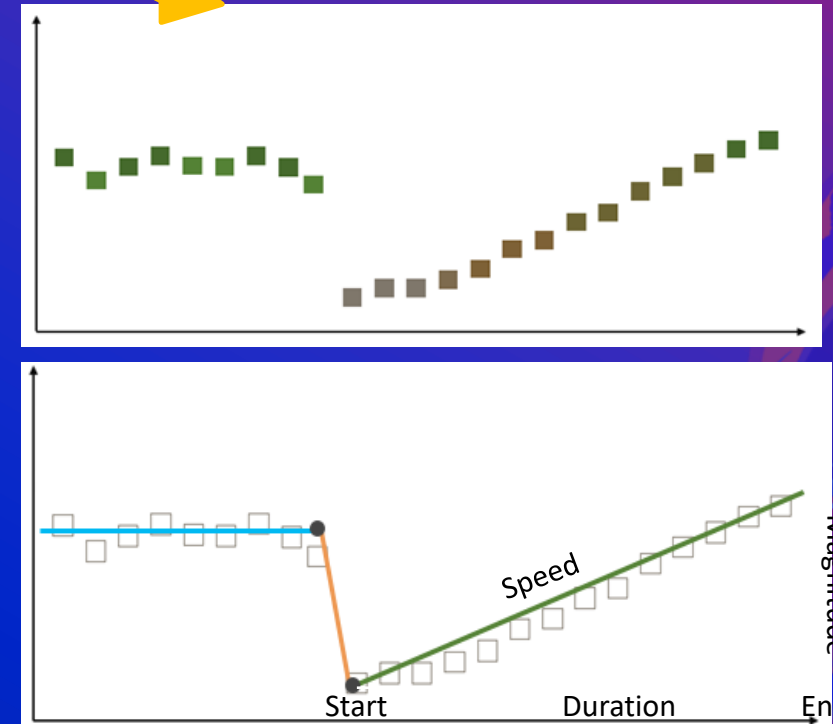
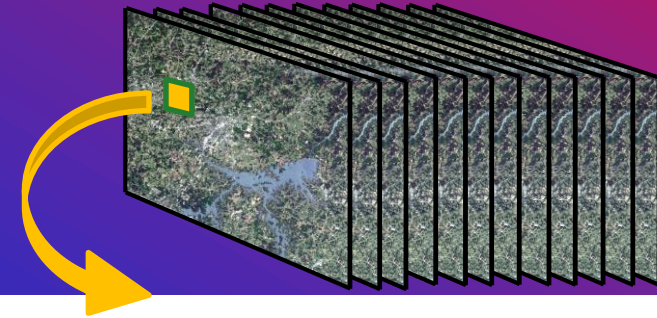
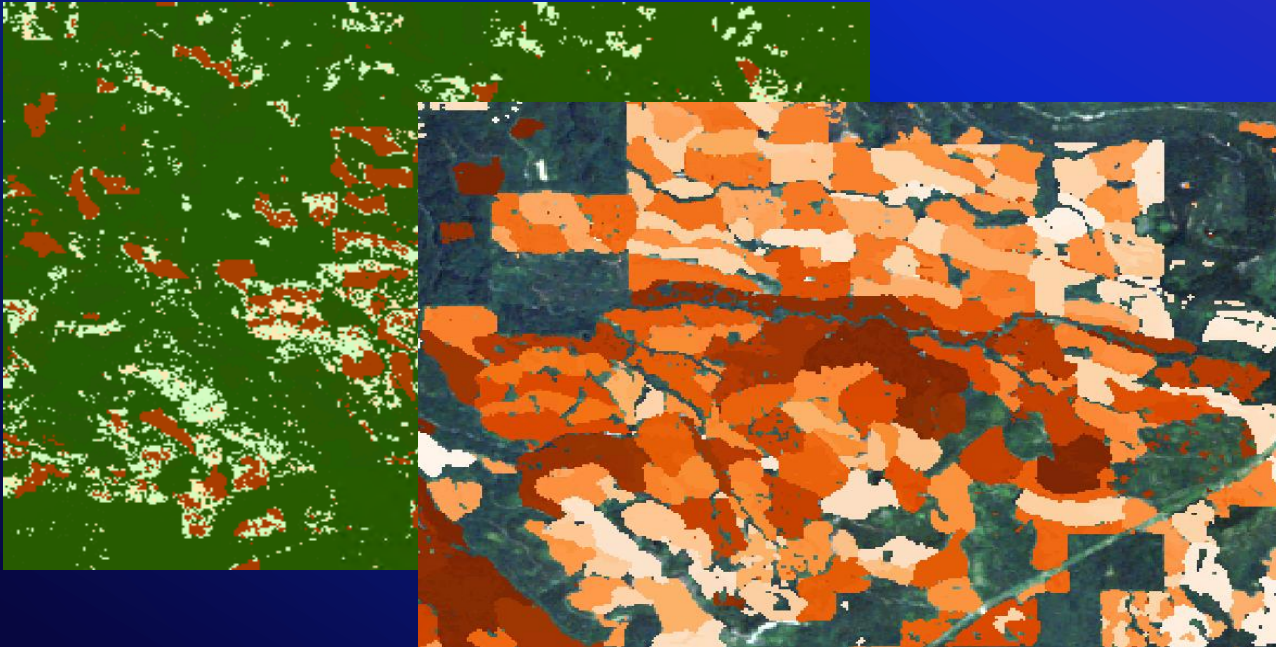
- **Guided workflows for:**
 - Categorical change detection
 - Time-series change detection
 - Pixel-value change detection
- Includes predefined differences based on index
- Postprocessing cleanup tools
 - Remove noise (neighborhood smoothing)
- Ability to write out
 - Function rasters
 - Raster function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

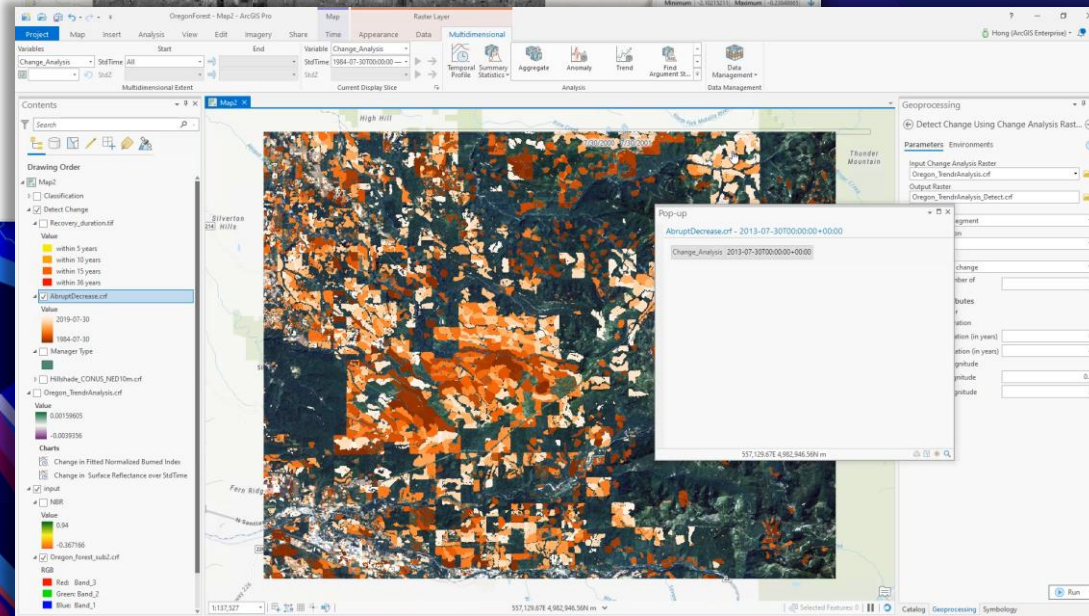
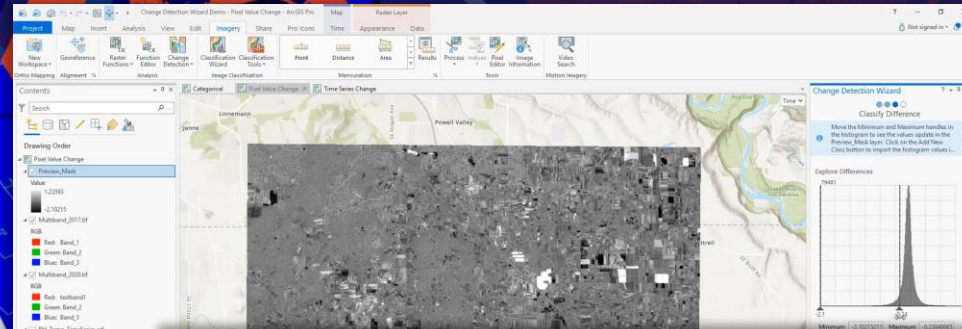


Change Detection

Landsat based detection of trends in disturbance and recovery

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





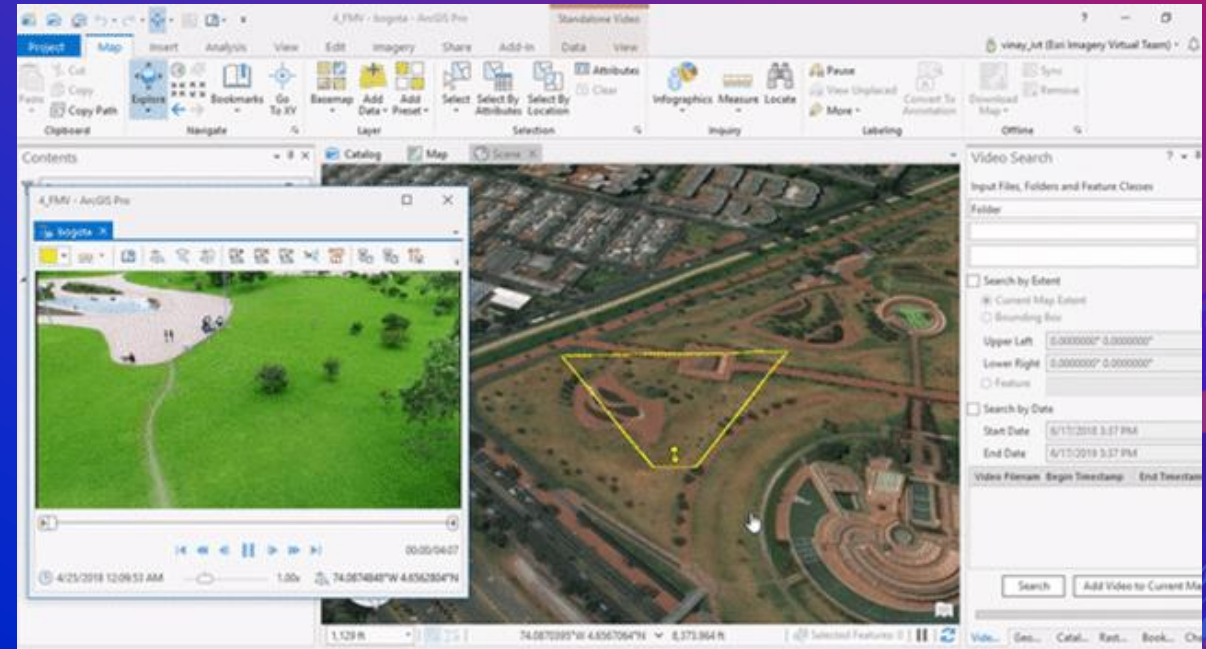
Demo – Change Detection

Vinay Viswambrahan

Motion Imagery

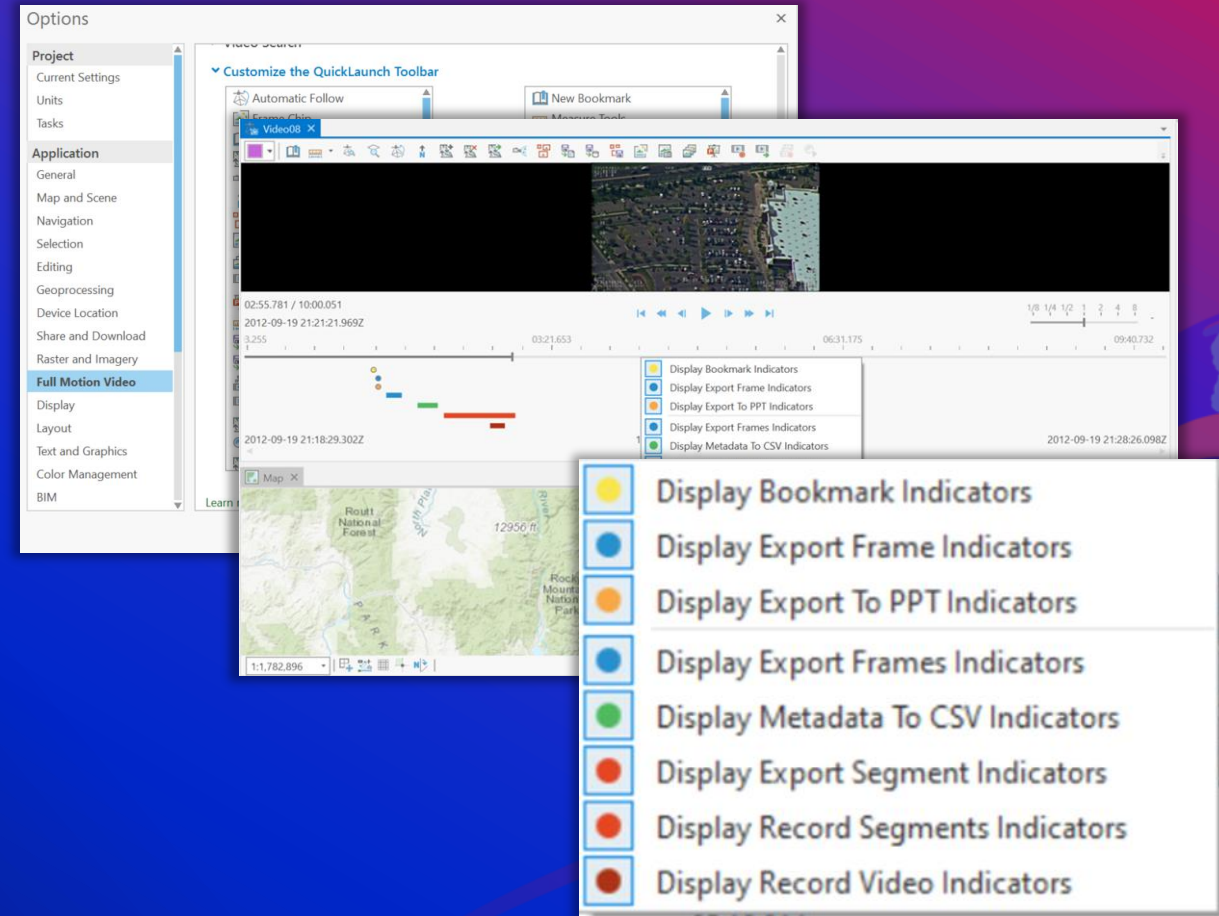
Make informed, timely decisions

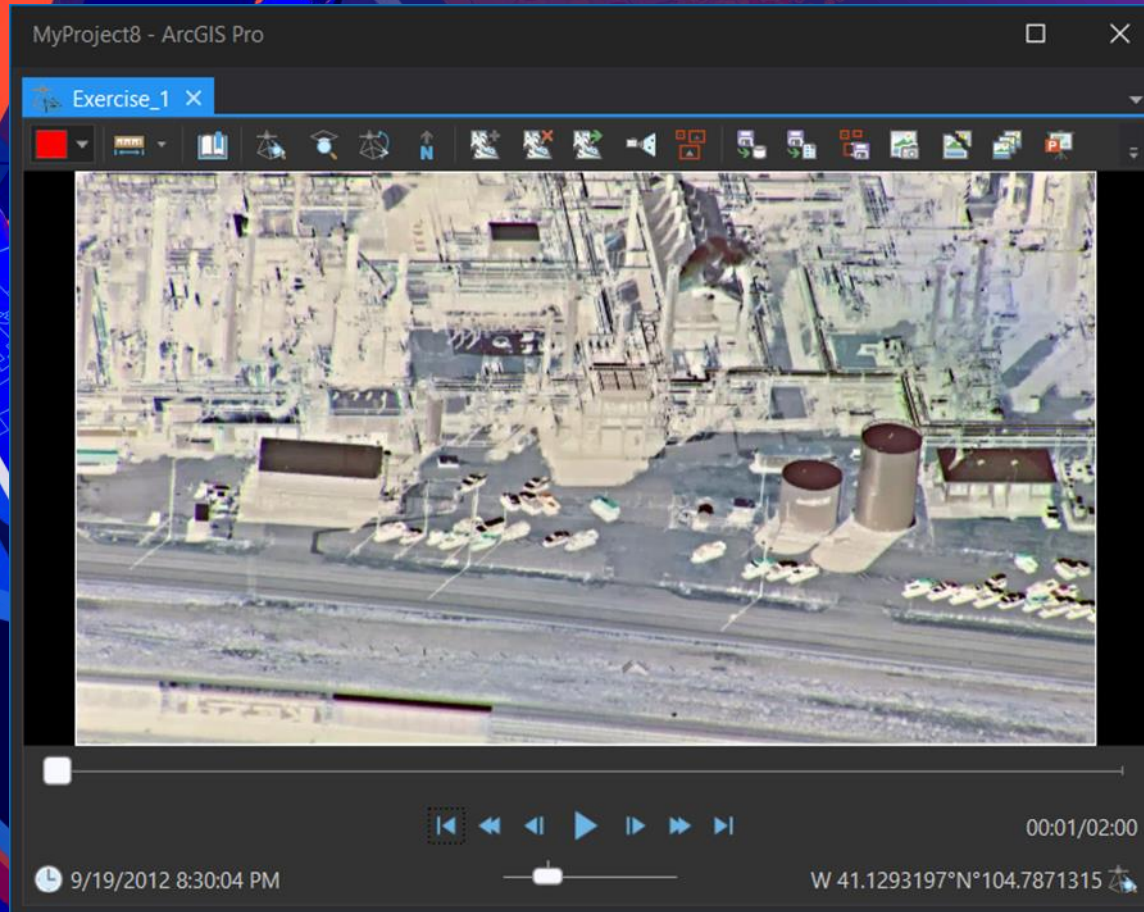
- **Integrate**
 - Seamlessly view and capture features from your videos and maps
- **Reporting**
 - Efficient reporting capability for decision makers
- **Compatible**
 - Directly use video with MISB-formatted metadata, or use the Multiplexer to convert video to MISB compatible format



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



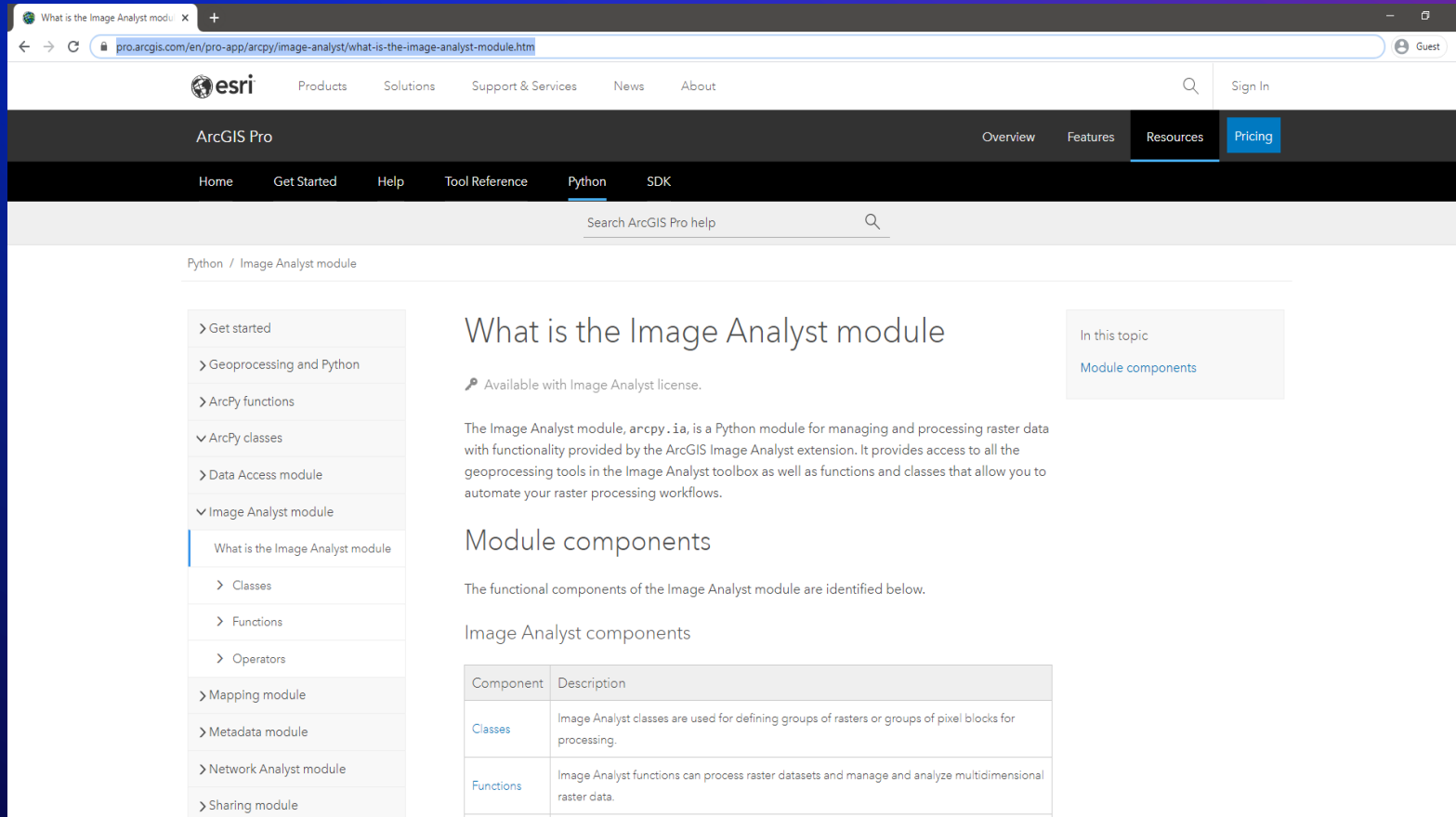


Demo – Motion Imagery

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 the ArcGIS Pro help website. The browser address bar displays the URL: pro.arcgis.com/en/pro-app/arcpy/image-analyst/what-is-the-image-analyst-module.htm. The page features a navigation bar with links for Products, Solutions, Support & Services, News, and About. Below this, there's a section for ArcGIS Pro with tabs for Overview, Features, Resources, and Pricing. The main content area is titled "Python / Image Analyst module" and includes a sidebar with a tree view of the module's structure. The main heading is "What is the Image Analyst module", followed by a note that it's available with the Image Analyst license. The text describes the module as a Python tool for managing and processing raster data. Below this, the "Module components" section states that functional components are identified below. A table titled "Image Analyst components" lists the components and their descriptions.

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

Conversion

RasterizeFeatures
TrendToRGB

Reclass

ZonalRemap

Analysis

ComputeChange
GenerateTrend
PredictUsingTrend
DetectChangeUsingChangeAnalysis
WeightedRasterSum
WeightRasterOverlay

Data Management & Conversion

Buffered
Reproject

Classification

Classify
LinearUnmixing
RegionGrow
SegMeanShift

Math

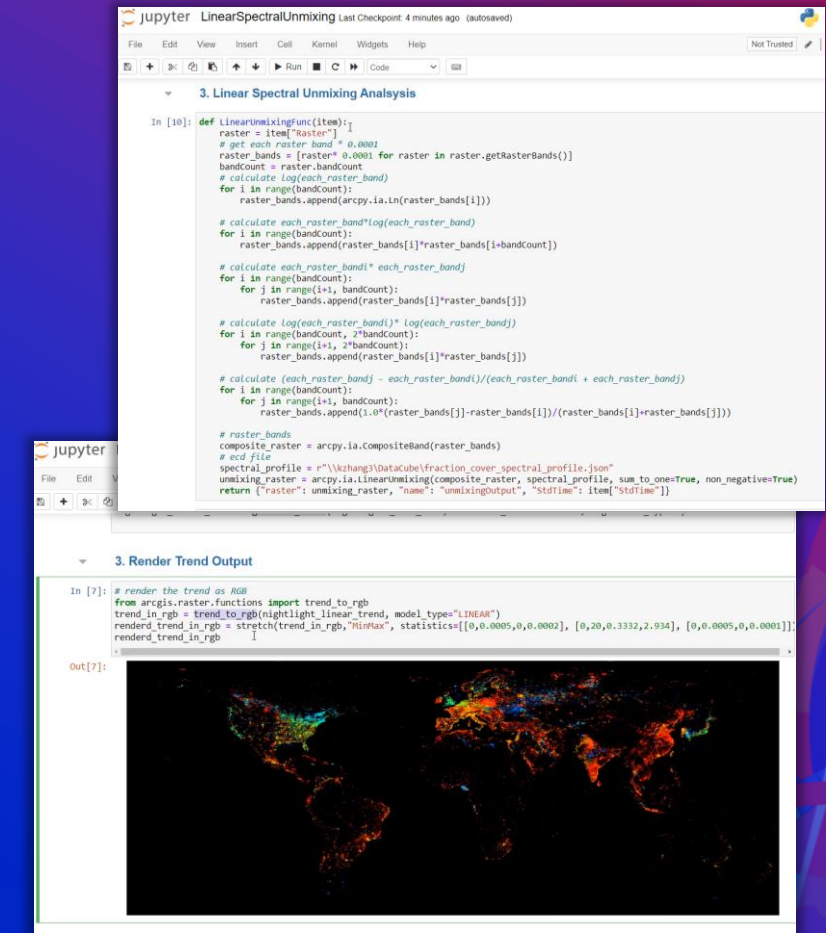
BAI
NBR
NDBI
NDSI

Statistics

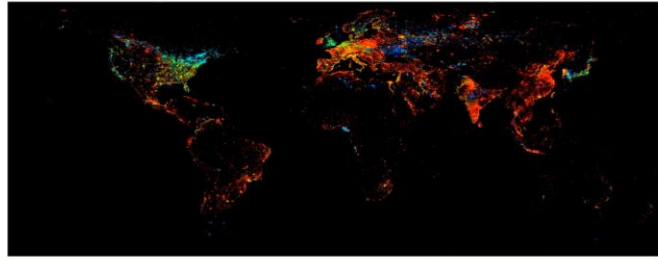
FocalStats

Correction

ApparentReflectance
Geometric
RadarCalibration
S1RadiometricCalibration
S1ThermalNoiseRemoval



The screenshot displays a Jupyter Notebook interface with two code cells. The top cell, titled '3. Linear Spectral Unmixing Analysis', contains a Python function `LinearSpectralUnmixingFunc` that processes a raster by calculating logarithmic values for each band and then combining them into a composite raster. The bottom cell, titled '3. Render Trend Output', uses `arcgis.raster.functions` to generate a trend from the unmixed raster and then stretches the result for visualization. The output of the second cell is a world map where different land cover types are represented by various colors (red, green, blue, yellow) against a black background, indicating the trend of change over time.

```
In [10]: def LinearSpectralUnmixingFunc(item):  
    raster = item["raster"]  
    # get each raster band * 0.0001  
    raster_bands = [raster * 0.0001 for raster in raster.getRasterBands()]  
    bandcount = raster.bandcount  
    # calculate log(each_raster_band)  
    for i in range(bandcount):  
        raster_bands.append(arcgpy.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_band)  
    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_band)*log(each_raster_band)  
    for i in range(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 = arcgpy.ia.CompositeBand(raster_bands)  
    # end file  
    spectral_profile = r"\\kzhang3\Datacube\fraction_cover_spectral_profile.json"  
    unmixing_raster = arcgpy.ia.LinearUnmixing(composite_raster, spectral_profile, sum_to_one=True, non_negative=True)  
    return {"raster": unmixing_raster, "name": "unmixingoutput", "stime": item["stime"]}  
  
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")  
renderd_trend_in_rgb = stretch(trend_in_rgb, "MinMax", statistics=[0,0.0005,0,0.0002], [0,20,0.3332,2.934], [0,0.0005,0,0.0001])  
renderd_trend_in_rgb  
  
Out[7]: 
```


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

File Edit View Insert Cell Kernel Widgets Help

Next Trusted Python 3.7

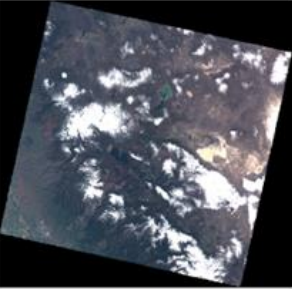
id	name	type	description	size	created	modified	deleted	status	
397	4048	AMD_AWS_Earth_LandUse_CAT_Raster	OBJECTID = 4048	LC08_L1TP_045033_20171126_20171207_01_T1	0.0	1200.0	30.0	120.0	1
398	4049	AMD_AWS_Earth_LandUse_CAT_Raster	OBJECTID = 4049	LC08_L1TP_045033_20171214_20171223_01_T1	0.0	1200.0	30.0	120.0	1
399	4050	AMD_AWS_Earth_LandUse_CAT_Raster	OBJECTID = 4050	LC08_L1TP_045033_20171230_20180103_01_T1	0.0	1200.0	30.0	120.0	1

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((quality_raster == 2720) | (quality_raster == 2724) | (quality_raster == 2728) | (quality_raster == 2732))
    masked_clean_band_list = []
    for bandRaster in raster.getRasterBands():
        arcpy.ia.Con is the go tool, will persist data
        masked_clean_band = arcpy.ia.Apply(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"])

cleaned_rc = rc2.map(clean)
```


Out[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]:



Demo – arcpy.ia module

Explore NDVI data through time



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®