GeoAI with ArcGIS Pro

Cedric Despierre Corporon | Rami Alouta

GIS for a Sustainable World Conference

GeoAI With ArcGIS Pro

Agenda

- What is GeoAI
- What can it do?
- Where can you apply GeoAI (Sneak peak on the GP tools)
- A deeper dive into GeoAI within ArcGIS
- Getting Started | Environment & Licensing
- GeoAI in ArcGIS Pro
  - GUI Demo
- Advanced GeoAI Workflows with Python
  - Notebook Demo

What is GeoAI?

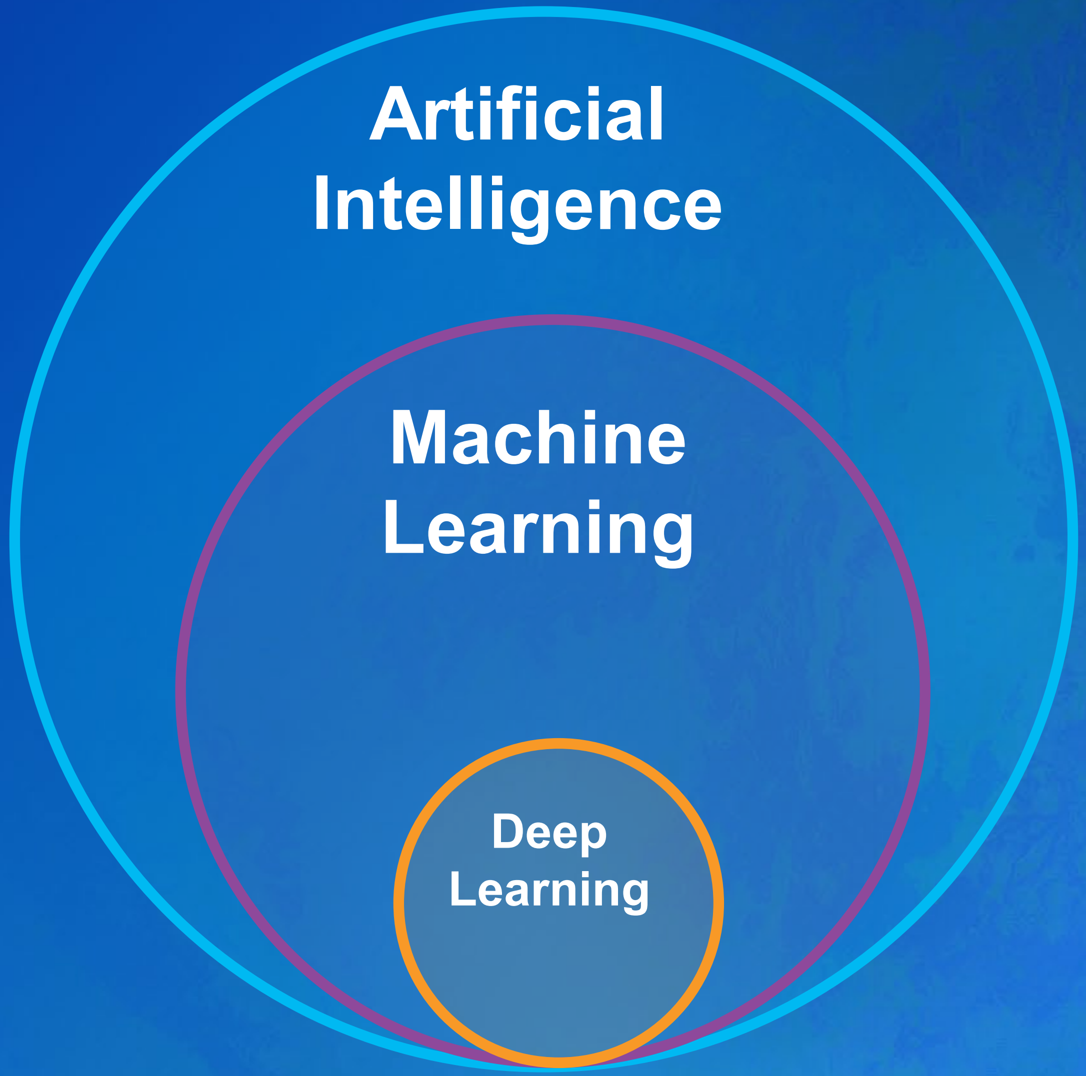
What Is AI?

Summary:

Really, machine learning (ML)

Machine learning is about extracting patterns from data to derive rules, instead of these rules being explicitly programmed.

Deep learning is a type of ML using deep neural networks to find complex patterns especially in unstructured data (such as images, text, voice, and lidar).



GeoAI in ArcGIS

Things to know

Object Tracking

PyTorch

CNTK

Object Detection

Artificial Intelligence

Computer Vision

scikit-learn

Random Forest

Neural Networks

Machine Learning

Natural Language Processing

Cognitive Computing

Caffe

TensorFlow

Data Science

GeoAI

Deep Learning

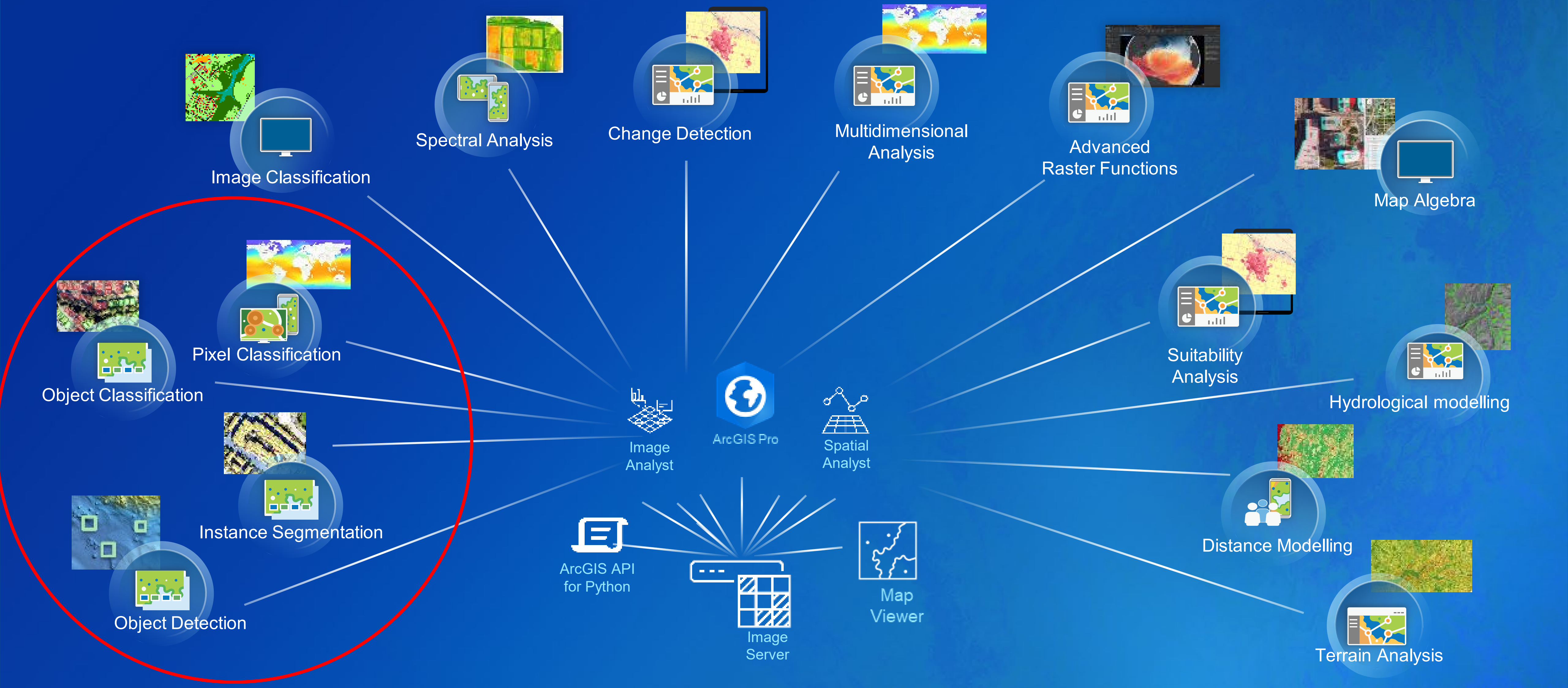
fast.ai

Keras

Dimensionality Reduction

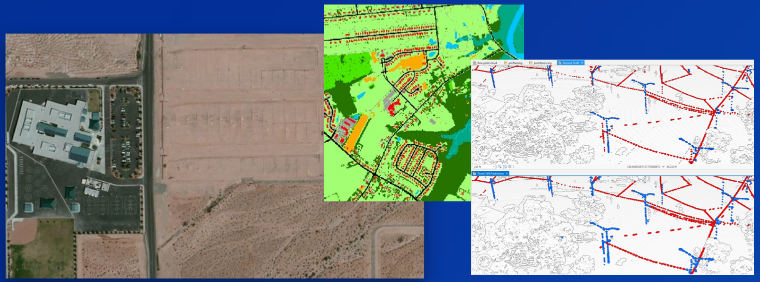
Support Vector Machines

Image Analysis Extracting Information From Imagery

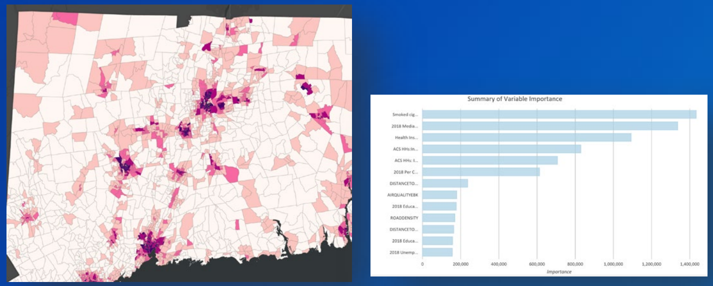


What can GeoAI Do?

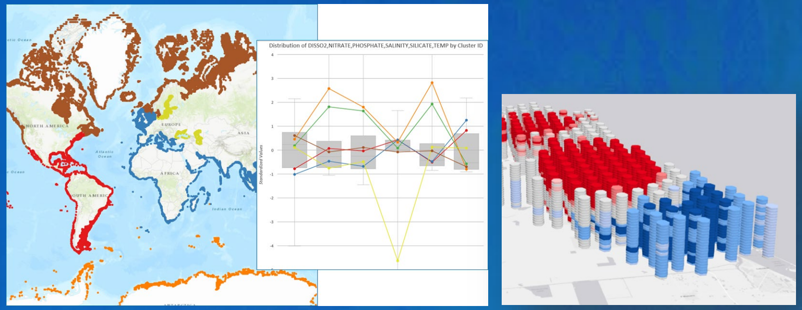
What Can Machine Learning Do?



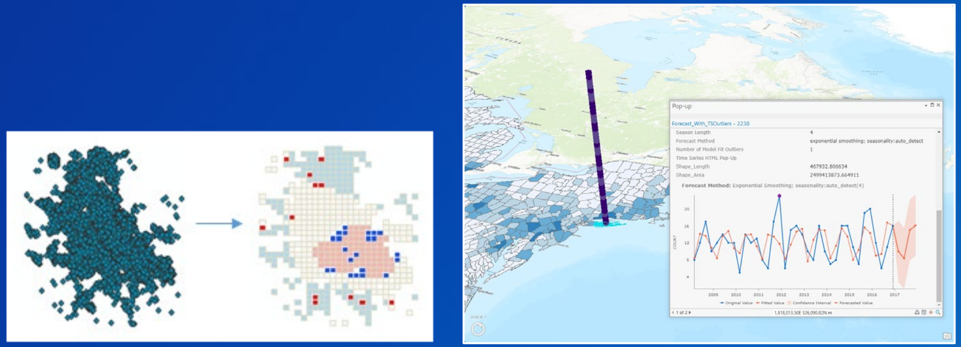
Extract features from
Imagery & LiDAR



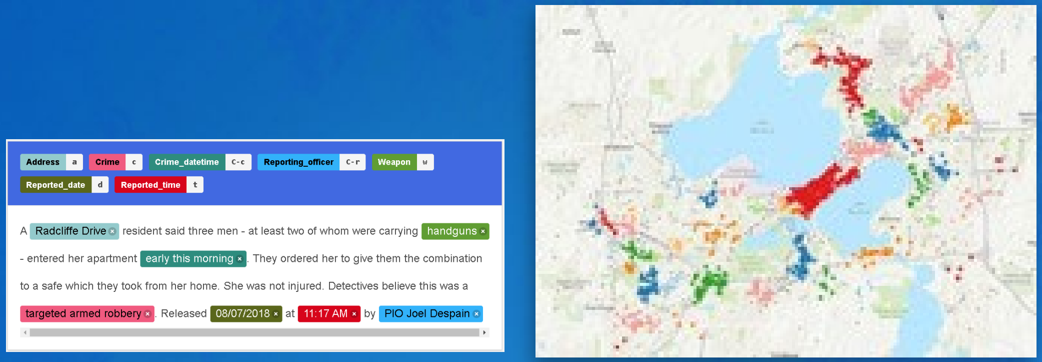
Make predictions



Find patterns & clusters



Detect anomalies



Extract insights from
unstructured text

Object Detection, Pixel Classification, Object Classification from Imagery

Building footprints

Roads

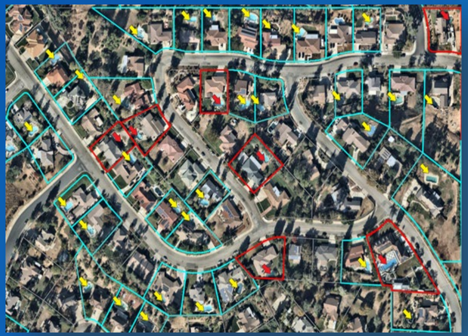
Land cover

Parcel boundaries

Palm trees

Oil pads

Swimming pools



Damaged structures

DL will detect what you can see in the imagery ……

Where can you apply GeoAI

AI is not one product. It spans the ArcGIS system.

Where we offer machine learning integration.

ArcGIS API for Python

ArcGIS Velocity

ArcGIS Notebooks

ArcGIS Pro

ArcGIS Online

ArcGIS Enterprise

ArcGIS Hub - Citizen Data Science

ArcGIS QuickCapture - Edge AI (in R&D)

ArcGIS Insights

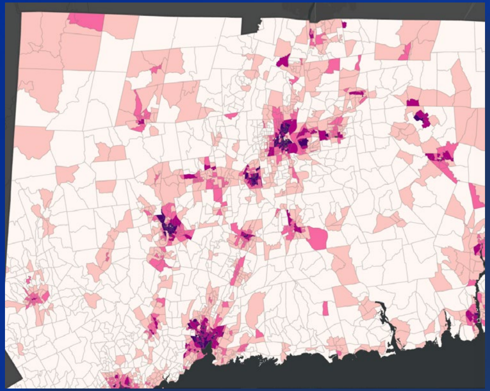
ArcGIS Pro for Intelligence

A deeper dive into GeoAI within ArcGIS

1. Predictive Analytics

Prepare Data, Make Predictions, Find Correlations, Understand Top Variables, and More

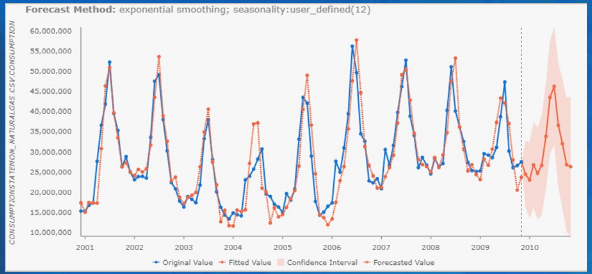
Predicting asthma rates



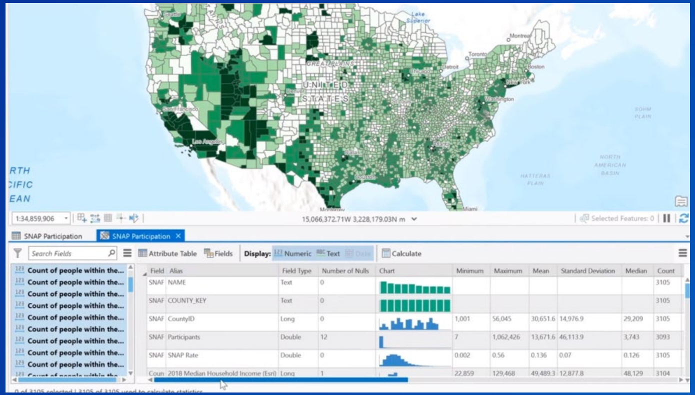
Predicting house pricing

Predicting sea habitat

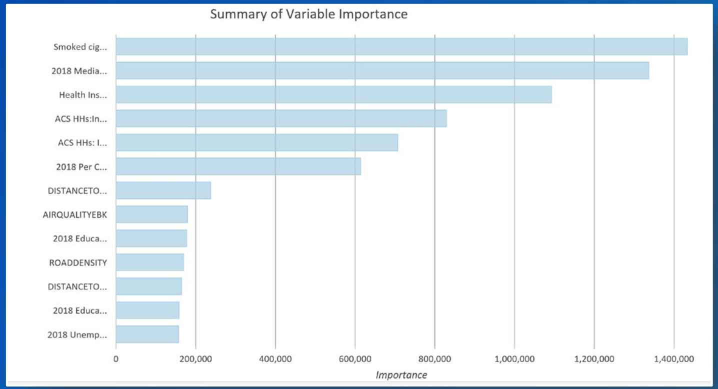
Time series forecasting



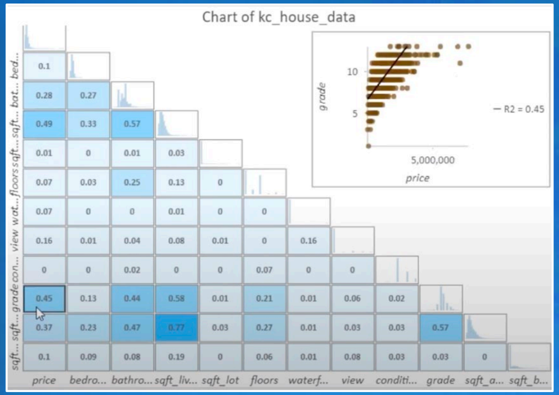
Data engineering



Variable importance

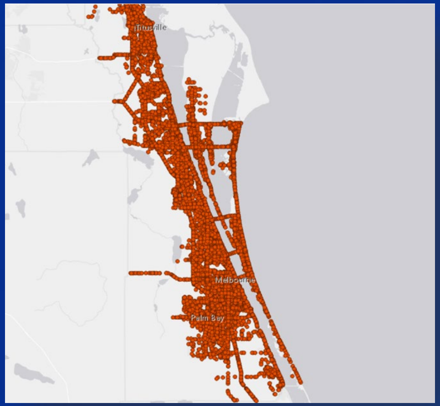


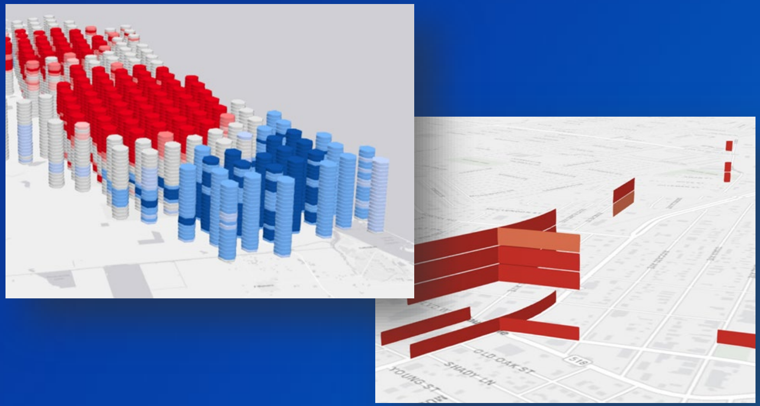
Variable correlation



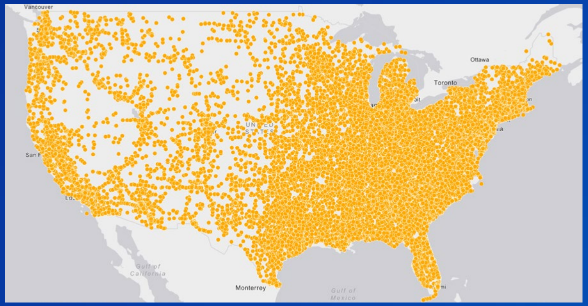
2. Pattern Mining and Clustering

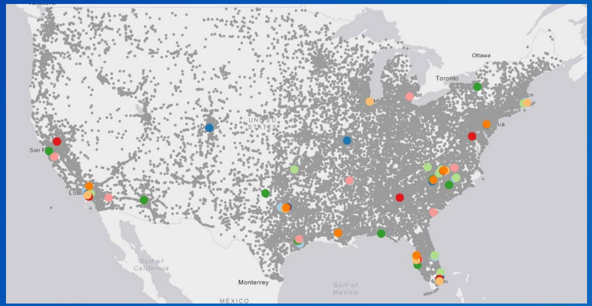
Understand Natural Groupings in Data That Are Statistically Significant



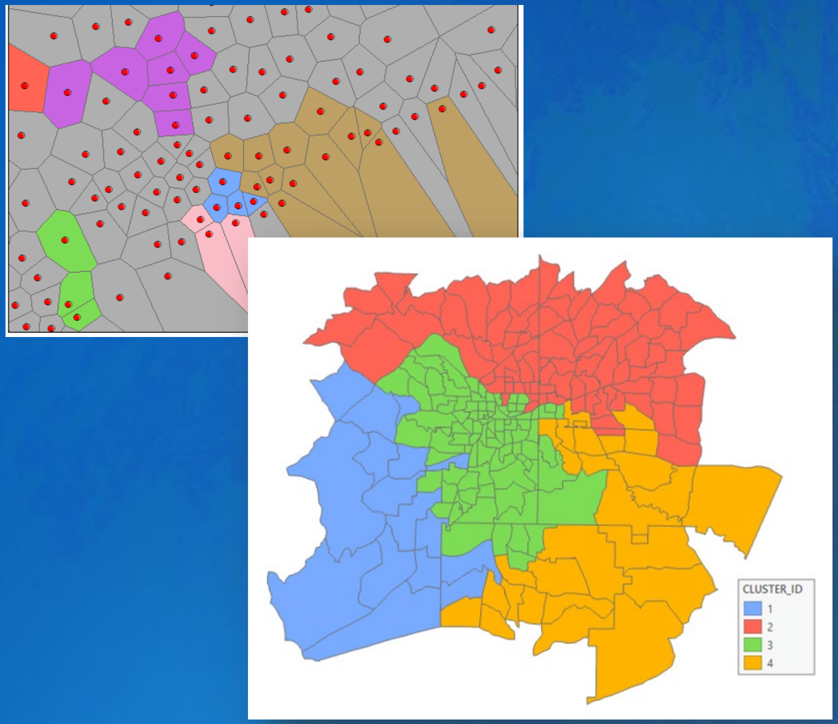


Emerging and fading hotspots for Crashes using **SpaceTime Pattern Mining Toolbox**





Most important fatal crashes clusters using **Density Based Clustering (DBScan)**

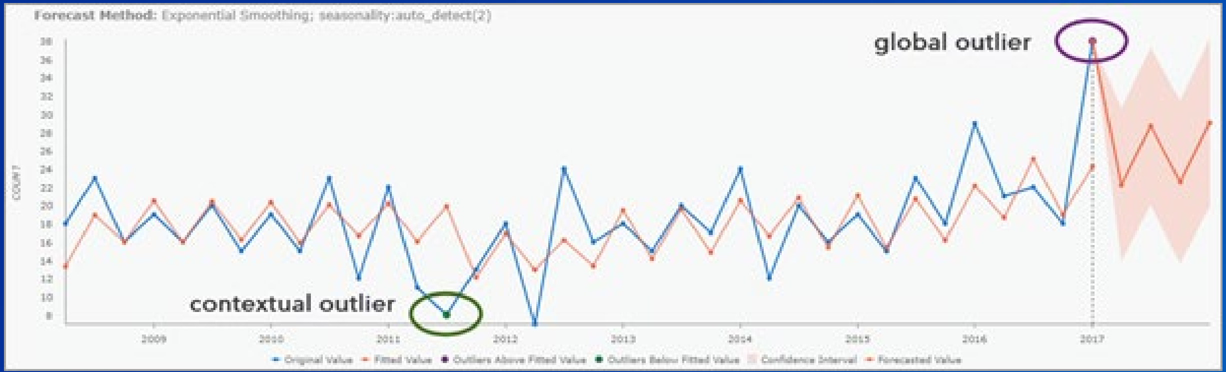


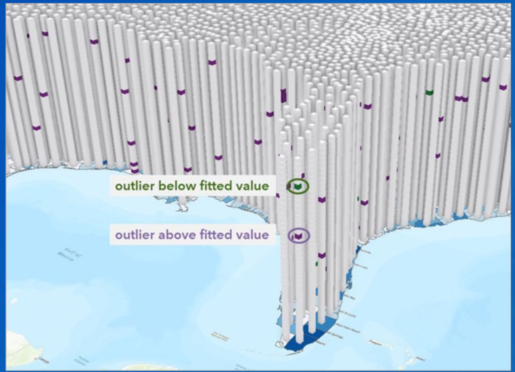
Find spatially contiguous clusters for animal territories using **Spatially Constrained Multivariate Clustering**

3. Anomaly Detection

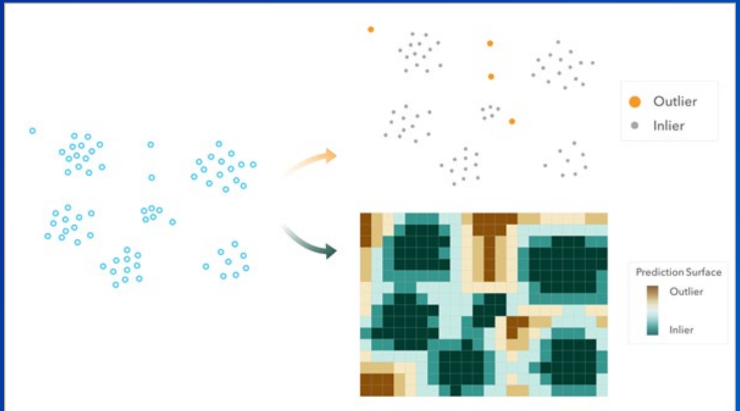
Spatial, Temporal, and Spatiotemporal Outliers

Time series outliers (temporal)

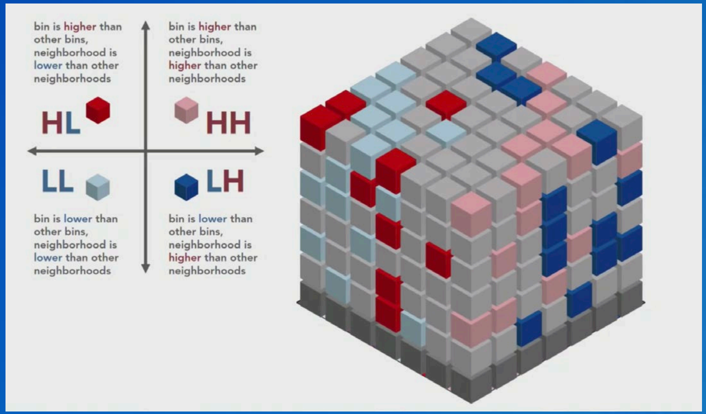


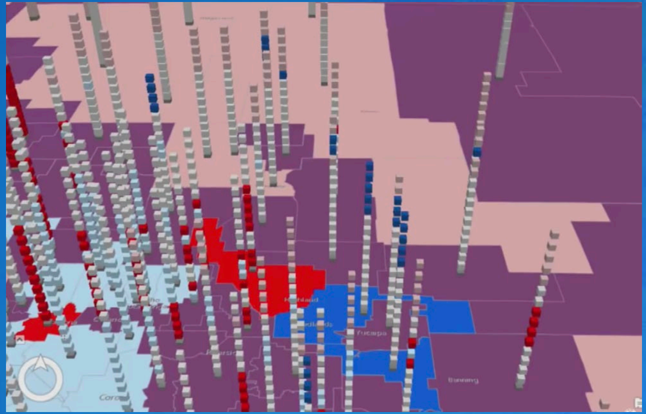


Spatial outliers (spatial)



Local outlier analysis (spatiotemporal)





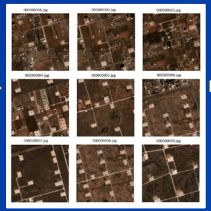
Imagery AI: End-to-End Workflow

Extract Insights from Imagery at Scale, with High Speed and Accuracy

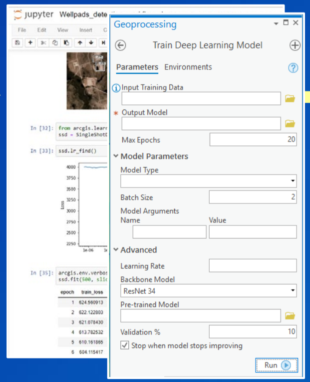
Image management

Labelling

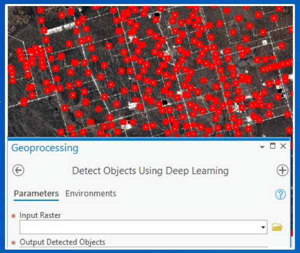
Data prep



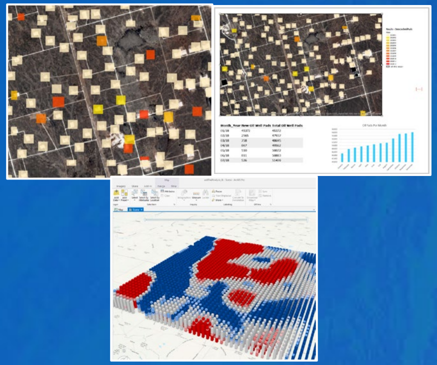
Model training



Inferencing



Analysis



Field mobility, monitoring

For Wide Range of Data Types

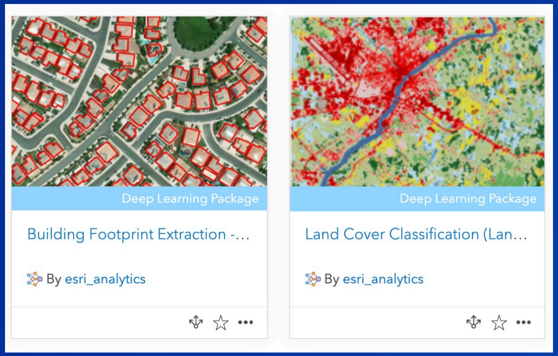
- Aerial
- Satellite
- Radar
- Lidar
- Motion imagery
- Bathymetry
- Point cloud
- Drone

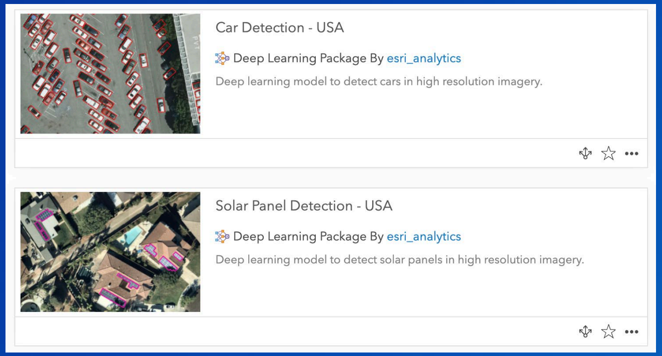
Implementing Many Tasks

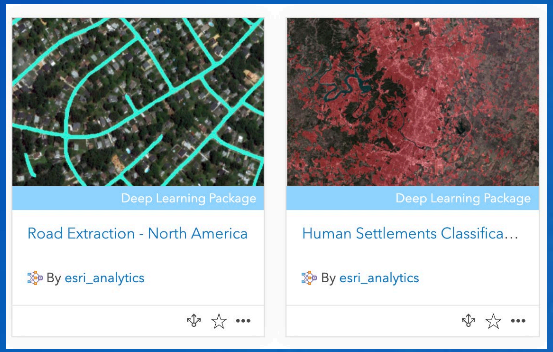
- Object classification
- Object detection
- Pixel classification
- Image translation
- Object tracking
- Scanned maps

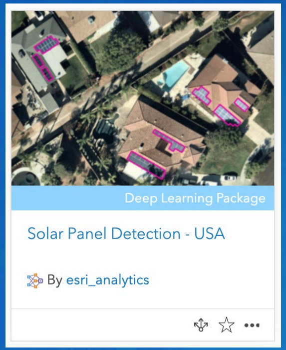
Pre-trained Models on ArcGIS Living Atlas

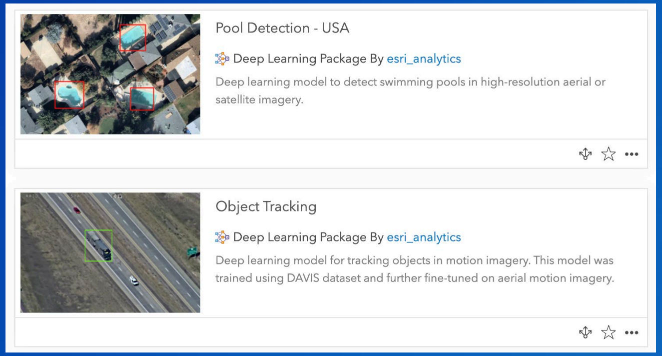
Plug-and-Play Models. No Training Needed. Easy Re-training Using Local Data.







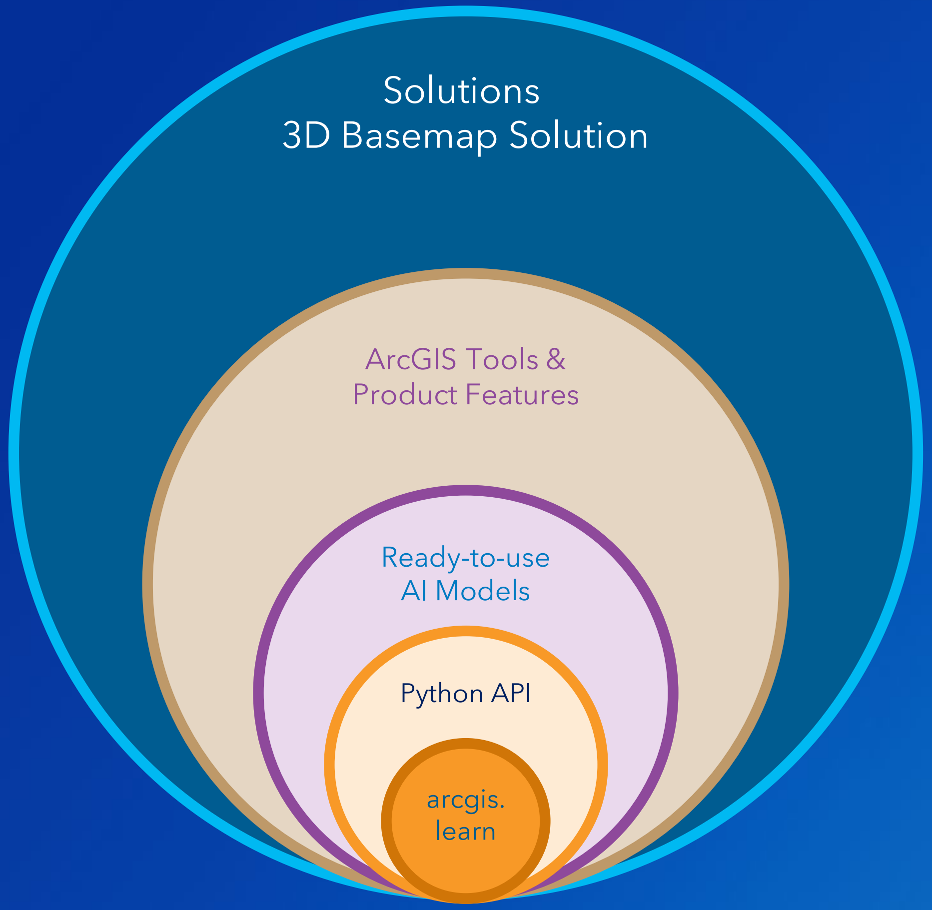


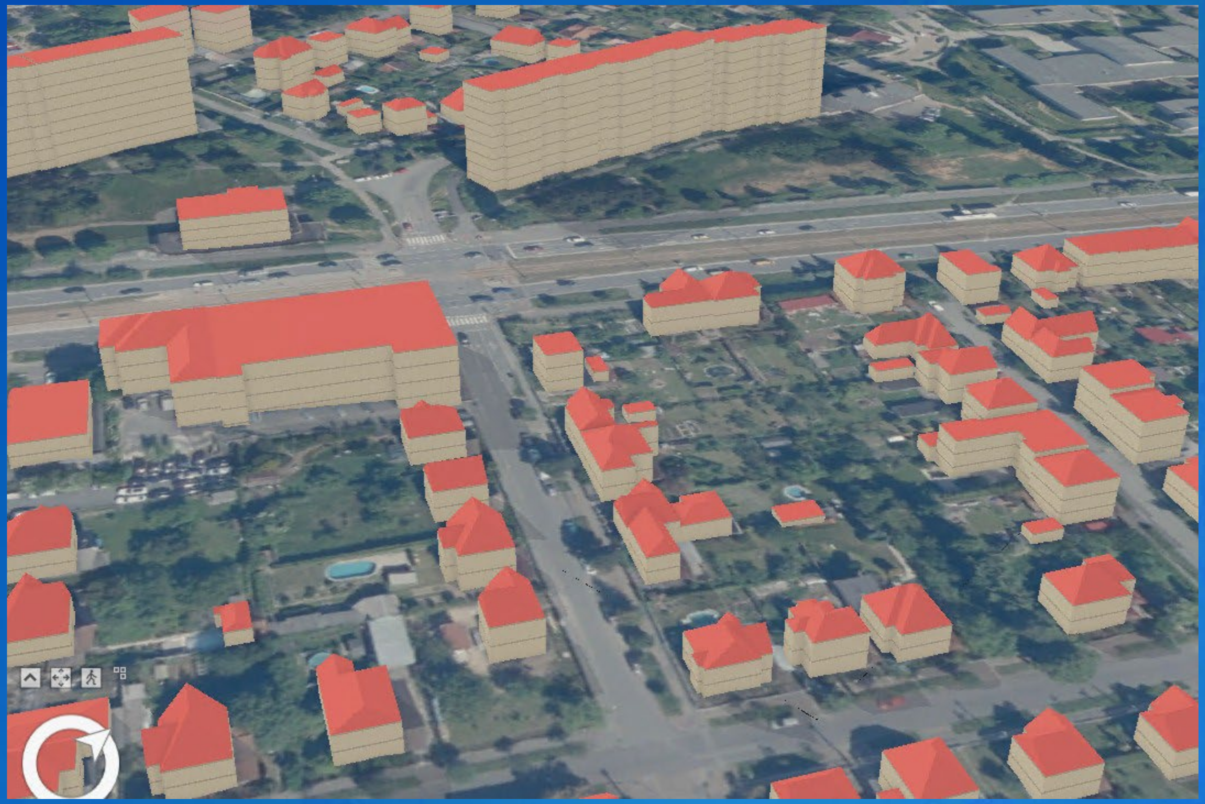


Use Models Within

- ArcGIS Pro (+ Image Analyst Extension)
- ArcGIS Enterprise (+ Image Server)
- ArcGIS Online (+ ArcGIS Image)

3D Basemap Solution – LOD2 Building Extraction

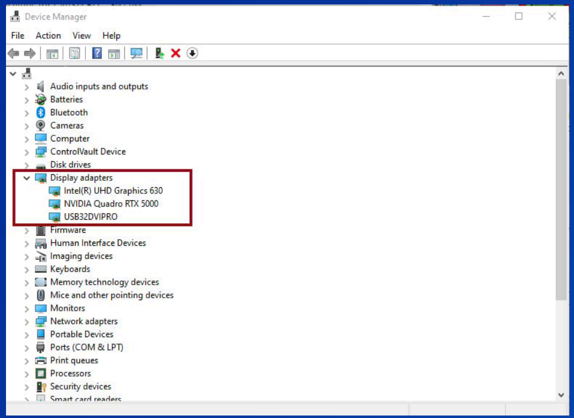




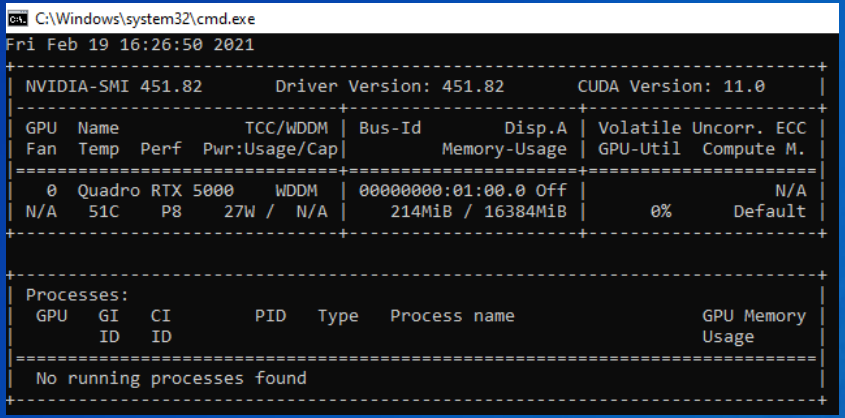
Getting Started | Environment & Licensing

Getting Started

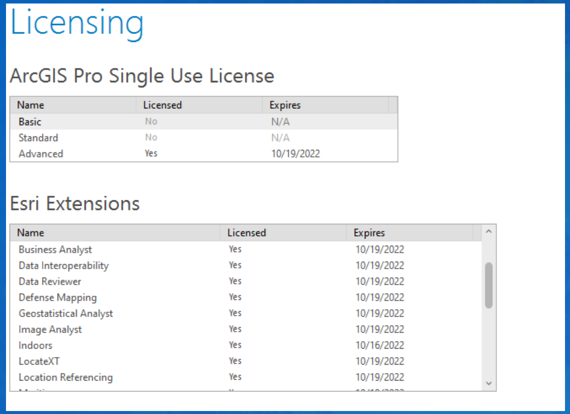
Prerequisites



GPU availability, specifications, and compatibility

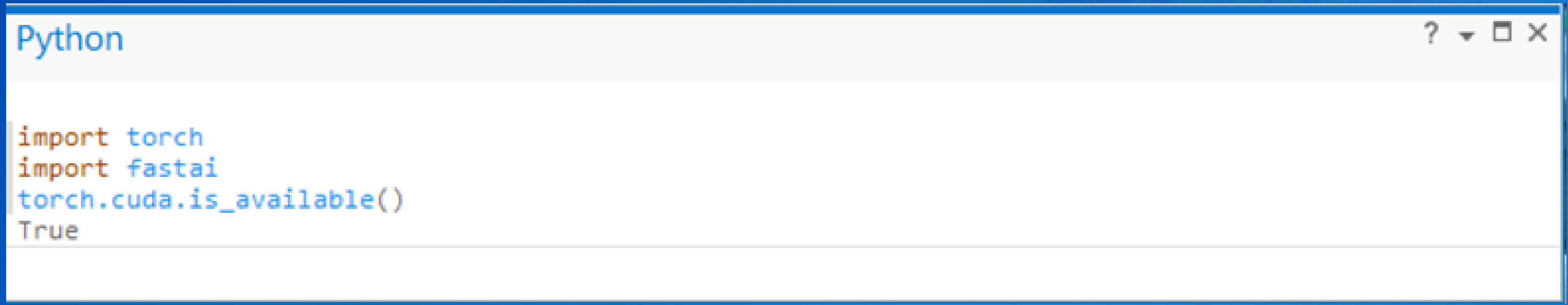


CUDA | nvidia-smi



ArcGIS Pro availability and version

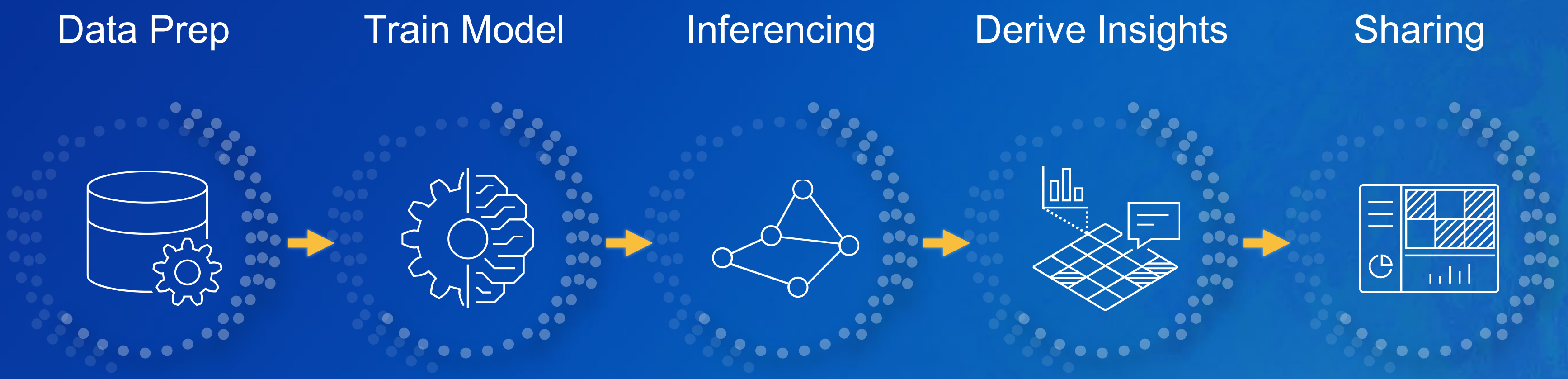
Deep learning framework for ArcGIS Pro installation



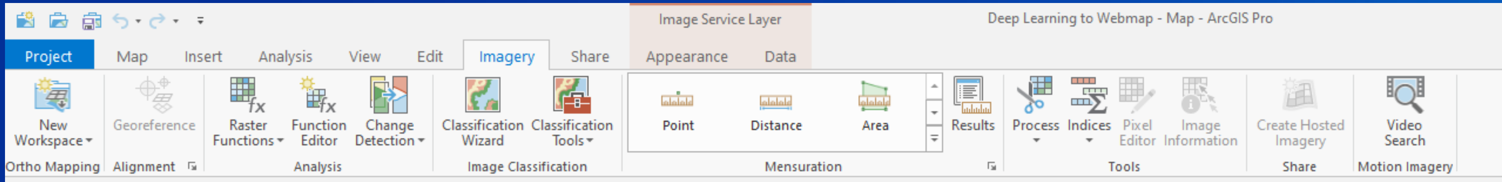
Communicating with the GPU

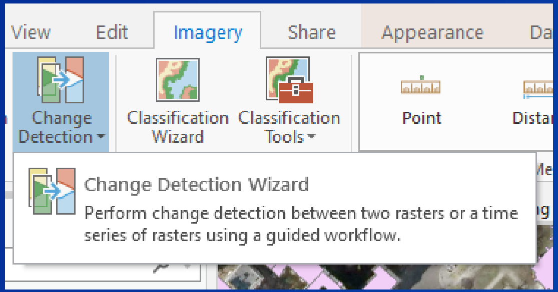
GeoAI in ArcGIS Pro

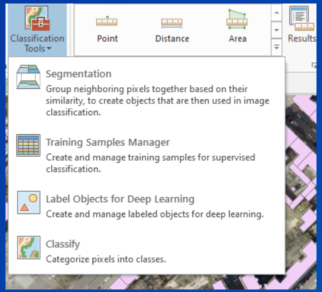
Machine Learning Lifecycle

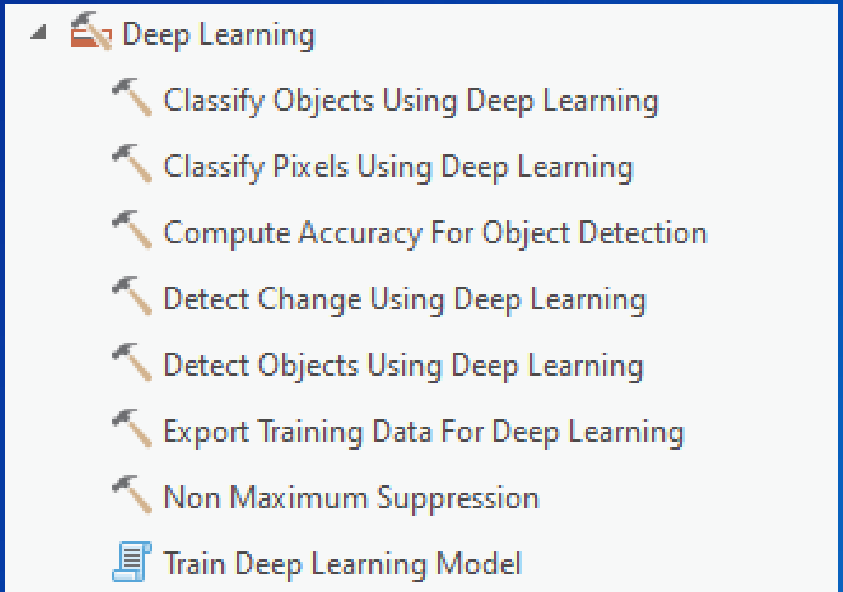


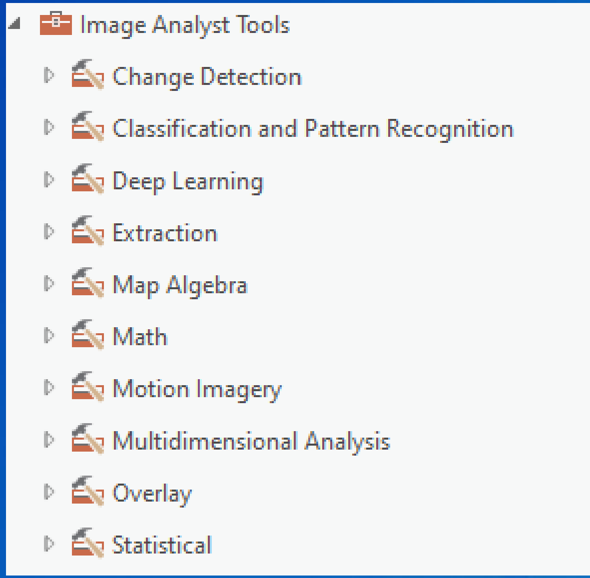
GeoAI Tools in ArcGIS Pro

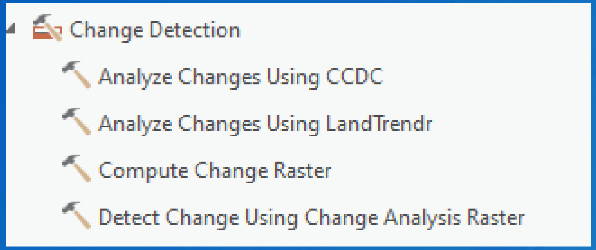


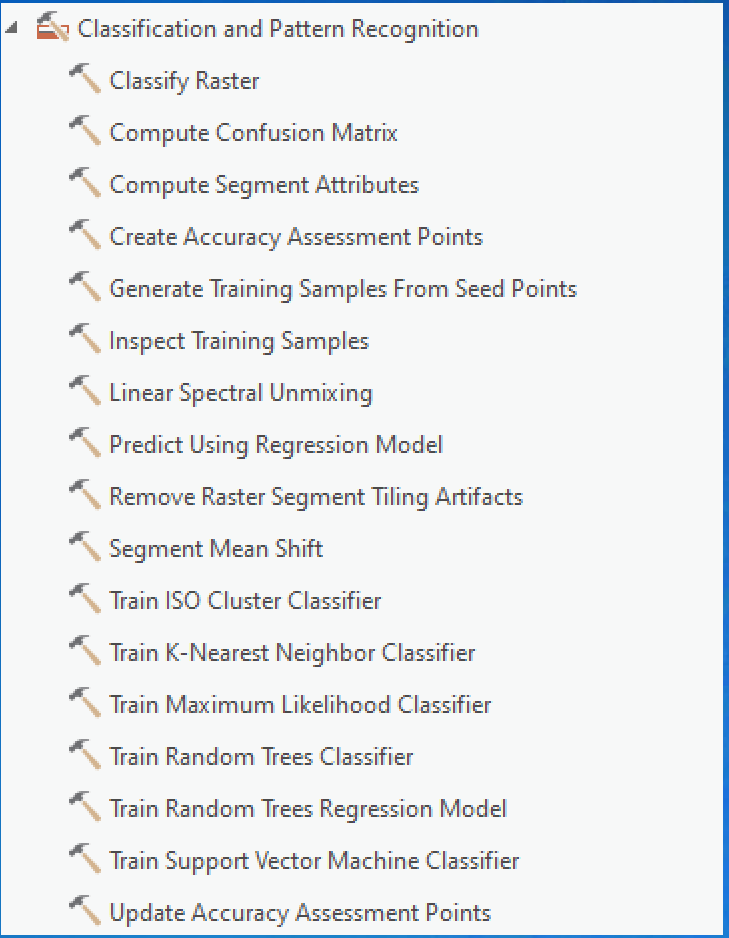








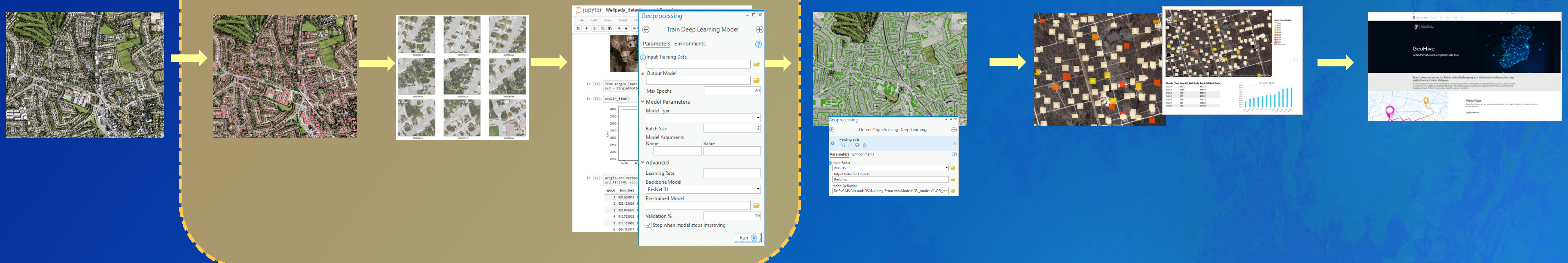




Deep Learning Workflow

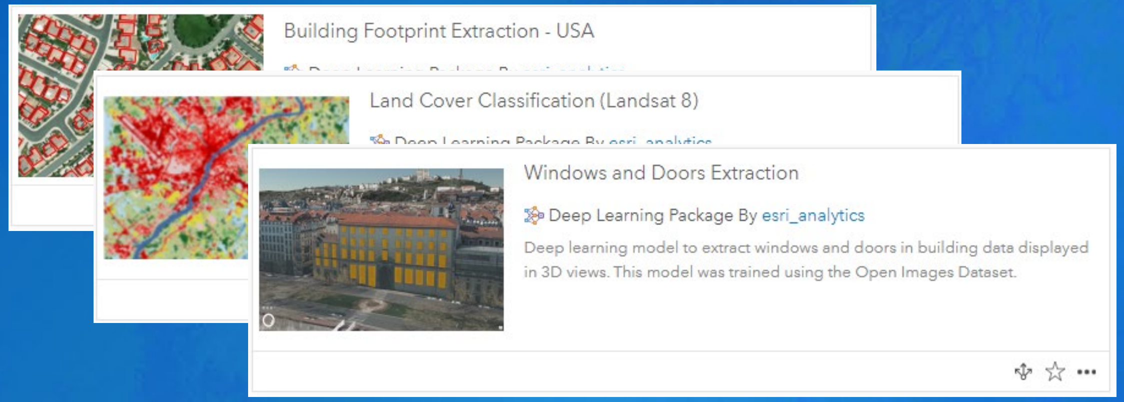
Pre-trained Models

Image Management → Labelling → Data Prep → Model Training → Inferencing → Post Processing → Sharing



Eliminates:

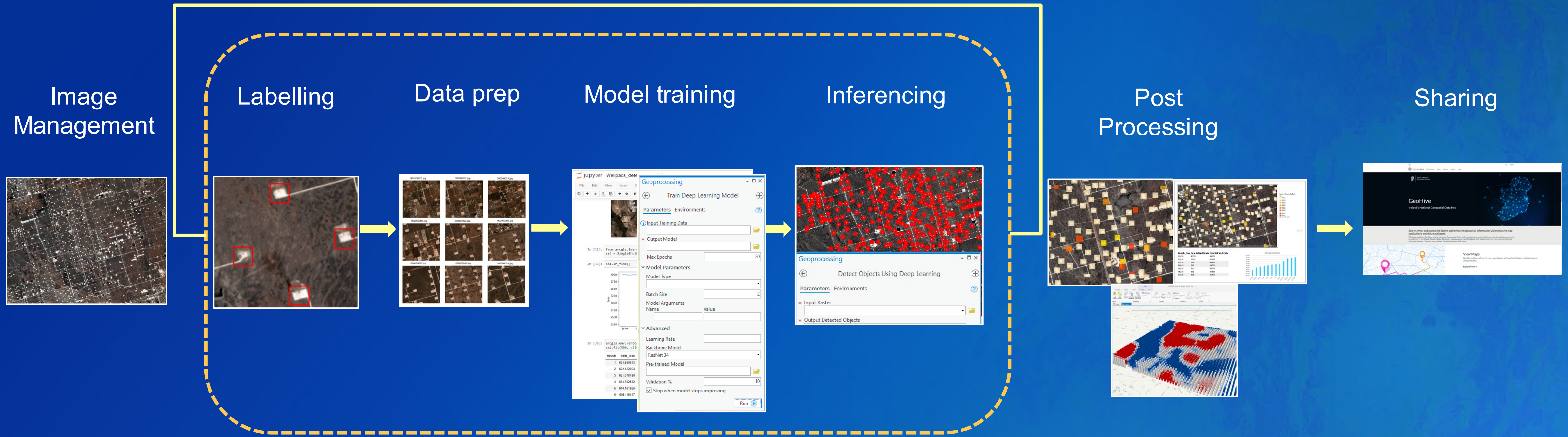
- Imagery requirements for model training
- Labelling requirements
- Training AI models
- Massive compute requirements



Applying DL with Out of the Box Model

Building Detection

Fine-Tuning Mask RCNN Building Detection Model.



For Wide Range of Data Types

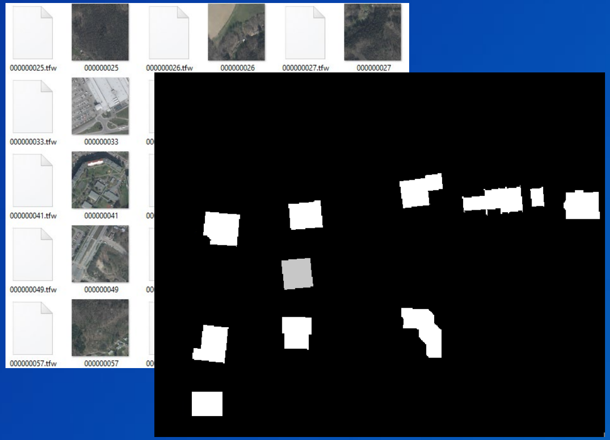
- Aerial
- Satellite
- Radar
- Lidar
- Motion imagery
- Bathymetry
- Point cloud
- Drone

Implementing Many Tasks

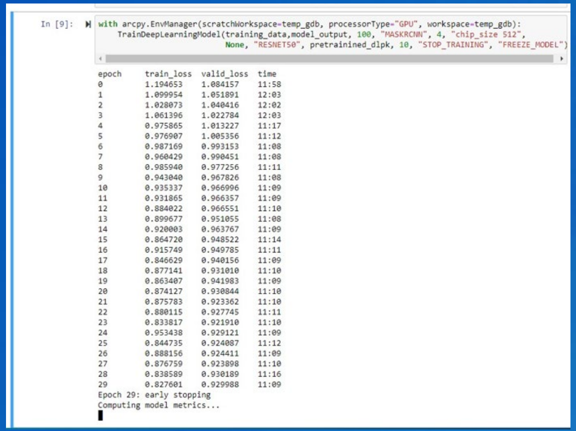
- Object classification
- Object detection
- Pixel classification
- Image translation
- Object tracking
- Scanned maps

Fine-Tuning Mask RCNN Building Detection Model.

Label with existing buildings



Data Prep



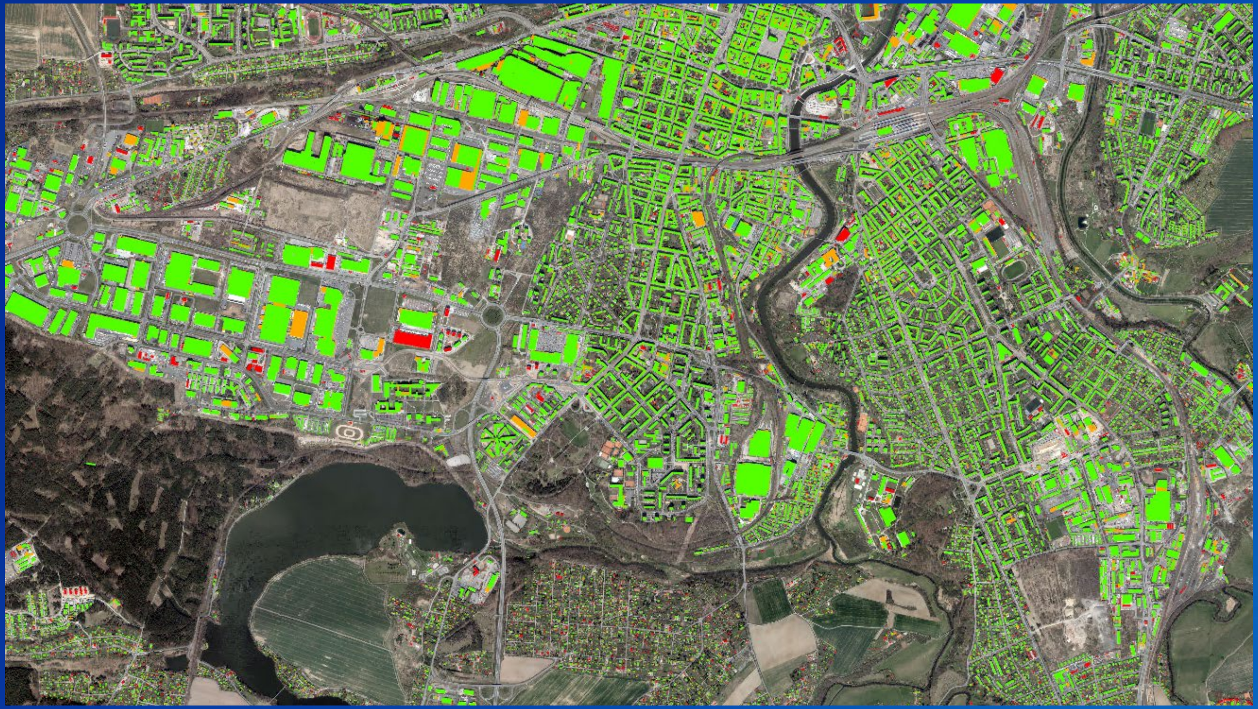
```
In [9]:  with arcpy.EnvManager(scratchWorkspace=temp_gdb, processorType="GPU", workspace=temp_gdb):
             TrainDeepLearningModel(training_data,model_output, 100, "MASKRCNN", 4, "chip_size 512",
                                    None, "RESNET50", pretrainined_dlpk, 10, "STOP_TRAINING", "FREEZE_MODEL")
```

epoch	train_loss	valid_loss	time
0	1.194653	1.084157	11:58
1	1.099954	1.051891	12:03
2	1.028073	1.040416	12:02
3	1.061396	1.022784	12:03
4	0.975865	1.013227	11:17
5	0.976907	1.005356	11:12
6	0.987169	0.993153	11:08
7	0.960429	0.990451	11:08
8	0.985940	0.977256	11:11
9	0.943040	0.967826	11:08
10	0.935337	0.966996	11:09
11	0.931865	0.966357	11:09
12	0.884022	0.966551	11:10
13	0.899677	0.951055	11:08
14	0.920003	0.963767	11:09
15	0.864720	0.948522	11:14
16	0.915749	0.949785	11:11
17	0.846629	0.940156	11:09
18	0.877141	0.931010	11:10
19	0.863407	0.941983	11:09
20	0.874127	0.930844	11:10
21	0.875783	0.923362	11:10
22	0.880115	0.927745	11:11
23	0.833817	0.921910	11:10
24	0.953438	0.929121	11:09
25	0.844735	0.924087	11:12
26	0.888156	0.924411	11:09
27	0.876759	0.923898	11:10
28	0.838589	0.930189	11:16
29	0.827601	0.929988	11:09

Epoch 29: early stopping
Computing model metrics...

Model training (fine-tuning)

Inference with fine-tuned model (71500 features)



Fine-Tunning the Deep Learning Package

Building Detection

Advanced GeoAI Workflows with Python

Automation end to end workflow with ArcGIS Pro Notebooks.

- Utilize ArcPy & Python API to automate process

deep_learning_workflow

Edit View Insert Cell Help ArcGISPro

Run Code

In [7]:
```
with arcpy.EnvManager(cellSize=0.3):
    ExportTrainingDataForDeepLearning("temp_mosaic", training_data, training_dataset, 'TIFF',
                                      '512', '512', '0', '0', "ONLY_TILES_WITH_FEATURES",
                                      'RCNN_Masks', 0, 'class', 0, None, 0, "MAP_SPACE",
                                      "PROCESS_AS_MOSAICKED_IMAGE", "NO_BLACKEN", "FIXED_SIZE")
```

In [9]:
```
with arcpy.EnvManager(scratchWorkspace=temp_gdb, processorType="GPU", workspace=temp_gdb):
    TrainDeepLearningModel(training_data,model_output, 100, "MASKRCNN", 4, "chip_size 512",
                           None, "RESNET50", pretrained_dlpk, 10, "STOP_TRAINING", "FREEZE_MODEL")
```

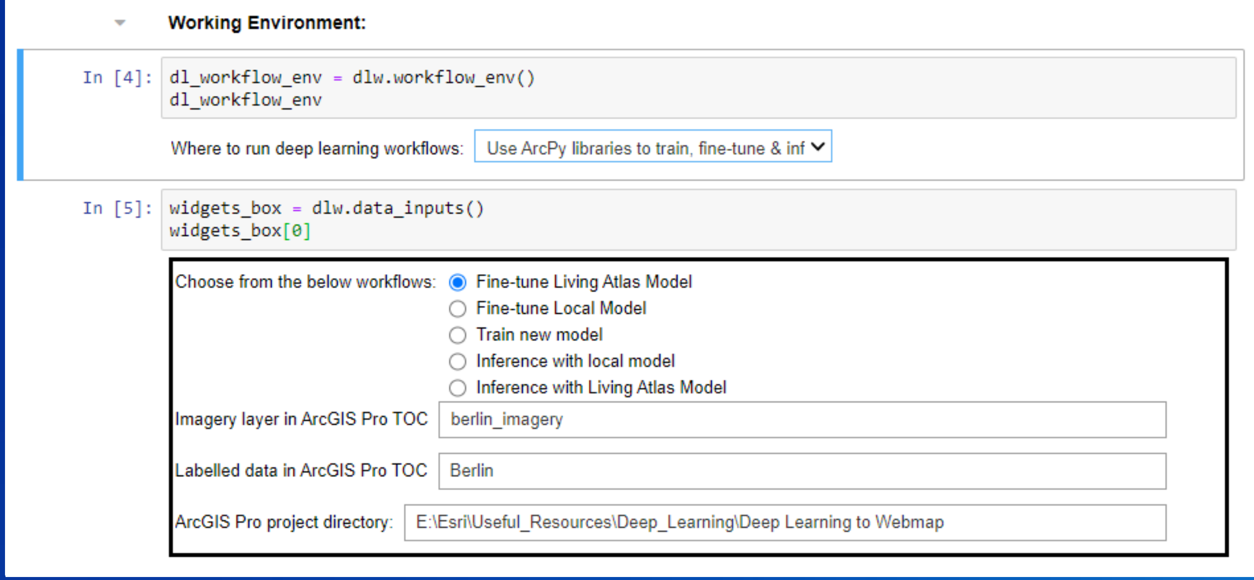
epoch	train_loss	valid_loss	time
0	1.194653	1.084157	11:58
1	1.099954	1.051891	12:03
2	1.028073	1.040416	12:02
3	1.061396	1.022784	12:03
4	0.975865	1.013227	11:17
5	0.976907	1.005356	11:12
6	0.987169	0.993153	11:08
7	0.960429	0.990451	11:08
8	0.985940	0.977256	11:11
9	0.943040	0.967826	11:08
10	0.935337	0.966996	11:09
11	0.931865	0.966357	11:09
12	0.884022	0.966551	11:10
13	0.899677	0.951055	11:08
14	0.920003	0.963767	11:09
15	0.864720	0.948522	11:14
16	0.915749	0.949785	11:11
17	0.846629	0.940156	11:09
18	0.877141	0.931010	11:10
19	0.863407	0.941983	11:09
20	0.874127	0.930844	11:10
21	0.875783	0.923362	11:10
22	0.880115	0.927745	11:11
23	0.833817	0.921910	11:10
24	0.953438	0.929121	11:09
25	0.844735	0.924087	11:12
26	0.888156	0.924411	11:09
27	0.876759	0.923898	11:10
28	0.838589	0.930189	11:16
29	0.827601	0.929988	11:09

Epoch 29: early stopping
Computing model metrics...

In [2]:
```
images_directory = r"C:\Users\ramialouta\Documents\ArcGIS\Projects\deeplearning_czech\True_Ortho"
pretrained_dlpk = r"C:\Users\ramialouta\Documents\ArcGIS\Packages\usa_building_footprints.dlpk"
model_parameters = ["padding 64;batch_size 16;threshold 0.5;return_bboxes False;tile_size 512",
                    "padding 64;batch_size 4;threshold 0.5;return_bboxes False;tile_size 1024",
                    "padding 128;batch_size 16;threshold 0.5;return_bboxes False;tile_size 512",
                    "padding 128;batch_size 4;threshold 0.5;return_bboxes False;tile_size 1024"]
training_data = r"C:\Users\ramialouta\Documents\ArcGIS\Projects\deeplearning_czech\training\buildings_training_data_v2"
footrpints_fc = r"C:\Users\ramialouta\Documents\ArcGIS\Projects\deeplearning_czech\Footprints\SURE_Pilsen_building_footprints.gdb\Pilsen_buil
model_output = r"C:\Users\ramialouta\Documents\ArcGIS\Projects\deeplearning_czech\models\retrained_buildings_v2"
```

In [3]:
```
gdb_dir = images_directory.split("\\")[:-1]
gdb = f"{gdb_dir[-1]}_temp.gdb"
final_gdb = f"gdb_dir[-1].gdb"
gdb_dir[0] = f"{gdb_dir[0]}\\"
gdb_dir =  functools.reduce(os.path.join,gdb_dir)

training_dataset = "training_dataset"


# Execute CreateFileGDB
temp_gdb = os.path.join(gdb_dir, gdb)
final_gdb= os.path.join(gdb_dir, "deeplearning_czech.gdb")

if arcpy.Exists(temp_gdb):
    featureclasses = arcpy.ListFeatureClasses()
    fsc=[]
    for fc in featureclasses:
        fc=fc.replace("_","_-")
        fsc.append(fc)
else:
    arcpy.CreateFileGDB_management(gdb_dir, gdb)
    arcpy.env.workspace = temp_gdb
    arcpy.env.overwriteOutput = True


output_detected_buildings_fc = os.path.join(temp_gdb, "detected_buildings")
raw_buildings = os.path.join(final_gdb, "raw_detected_buildings")

arcpy.env.overwriteOutput = True
```

In [4]:
```
dlpk_name_extesion = model_output.split("\\")[-1]
dlpk= os.path.join(model_output, f"{dlpk_name_extesion}.dlpk")
start_datetime = datetime.now()
detected_outputs = []
for parameter in model_parameters:
    iter_startdt = datetime.now()
    padding = parameter.split(";")[0].split(" ")[-1]
    btachsize = parameter.split(";")[1].split(" ")[-1]
    threshold = str(int(float(parameter.split(";")[2].split(" ")[-1])*100))
    tilesize = parameter.split(";")[-1].split(" ")[-1]
    output_fc = f"czech_buildings_30cm_P{padding}_B{btachsize}_Threshold{threshold}_Tile{tilesize}_{dlpk_name_extesion}"
    with arcpy.EnvManager(processorType="GPU", cellSize=0.3, workspace=final_gdb):
        arcpy.ia.DetectObjectsUsingDeepLearning("temp_mosaic", output_fc, dlpk,
                                                parameter, "NO_NMS", "Confidence",
                                                "Class", 0, "PROCESS_ITEMS_SEPARATELY")

    torch.cuda.empty_cache()
    detected_outputs.append(output_fc)
    iter_enddt = datetime.now()
    iter_minutes_diff = (iter_enddt - iter_startdt).total_seconds() / 60.0
    print(f"The processing of image: {output_fc} took {int(iter_minutes_diff)} minutes")
```



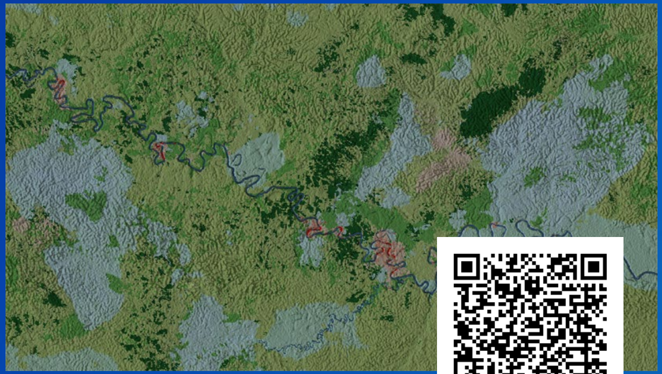
Deep Learning with Python

Deep Learning Resources

Deep Learning with ArcGIS Pro Tips & Tricks: Part 1

Scan Me

Deep Learning with ArcGIS Pro Tips & Tricks: Part 2

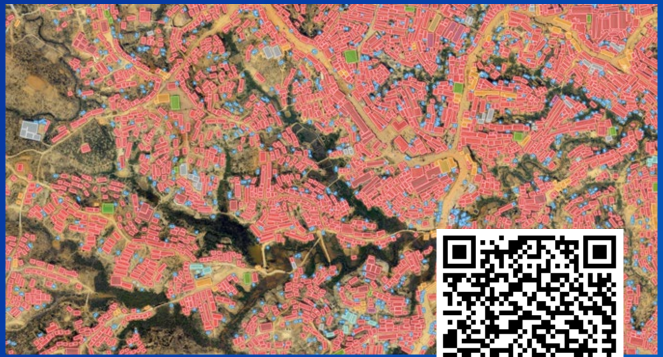


Scan Me

Deep Learning with ArcGIS Pro Part 3: QA/QC Extracted Features

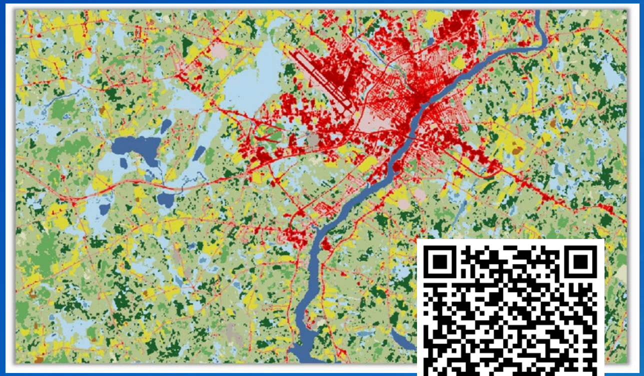
Scan Me

Deep Learning with ArcGIS Pro Part 3: QA/QC Extracted Features



Scan Me

Introducing pre-trained geospatial deep learning models



Scan Me

Pre-trained deep learning models update (February 2021)

Scan Me

Thank you to our sponsors

Co-host

esri
THE SCIENCE OF WHERE™

unitar
United Nations Institute for Training and Research

UNOSAT

esri®
THE SCIENCE OF WHERE®