



Geospatial Deep Learning with arcgis.learn

Andrew Chapkowski Twitter: @achapkowski

2020 ESRI FEDERAL GIS CONFERENCE | WASHINGTON, D.C.



A decorative background graphic at the bottom of the slide features a dark blue gradient with white contour lines. Overlaid on this are several large, semi-transparent cyan plus signs of varying sizes and orientations. In the lower-left corner, there is a small inset map showing a colorful land use or zoning classification. To the right, a vertical teal bar contains the text "SEE WHAT OTHERS CAN'T".

SEE
WHAT
OTHERS
CAN'T

Session Overview

- AI, Machine Learning & Deep Learning
- Deep Learning Workflow
- Training Models
 - ArcGIS Pro
 - `arcgis.learn`
- Types of models and their applications

Caffe

Object Tracking

CNTK

Object Detection

PyTorch

Artificial Intelligence

Computer Vision

scikit-learn

Random Forest

Machine
Learning

Neural Networks

Cognitive
Computing

TensorFlow

Natural Language
Processing

GeoAI

Deep
Learning

Data Science

Keras

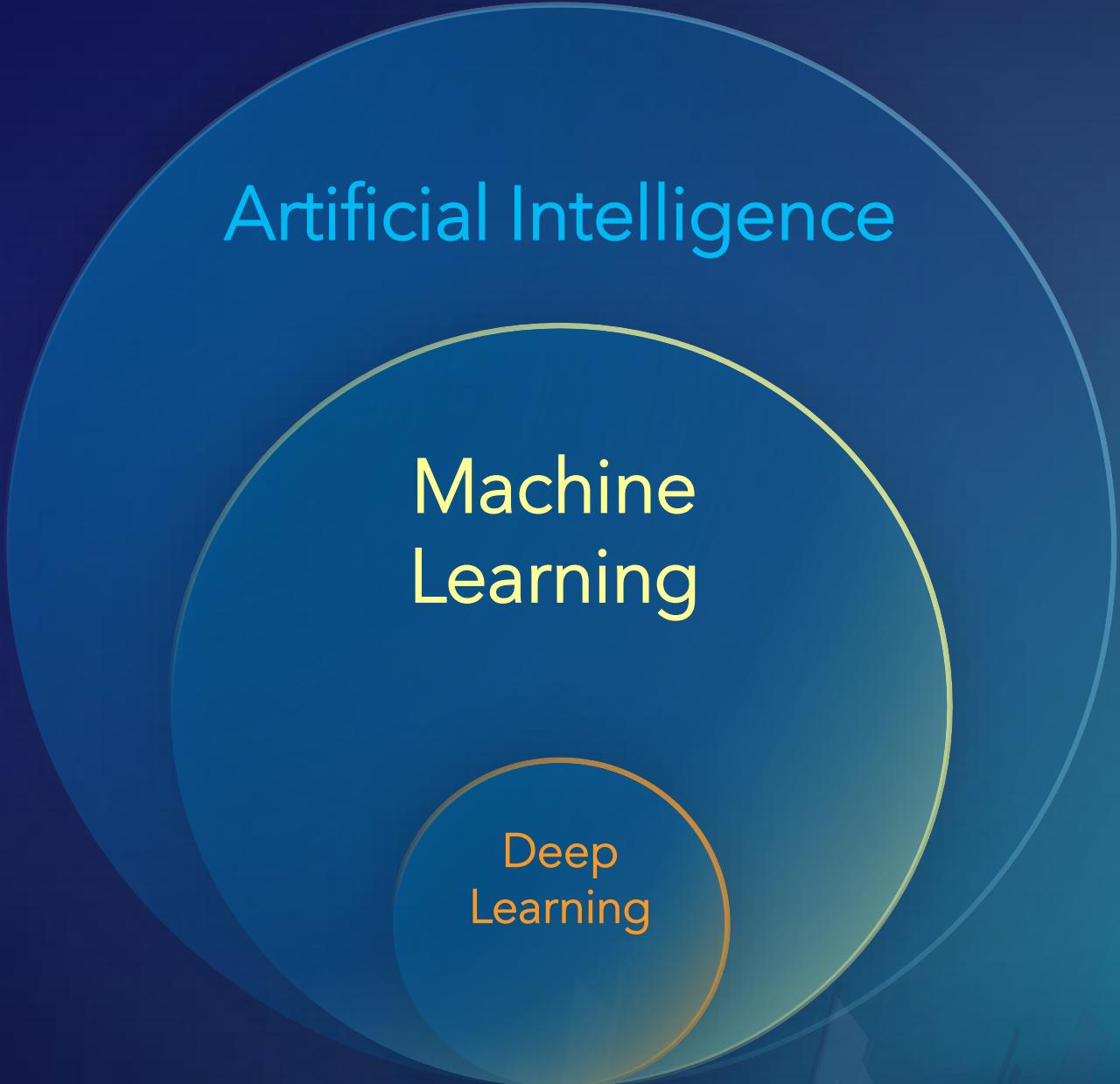
Dimensionality Reduction

Support Vector Machines

Artificial Intelligence

Machine Learning

Deep
Learning

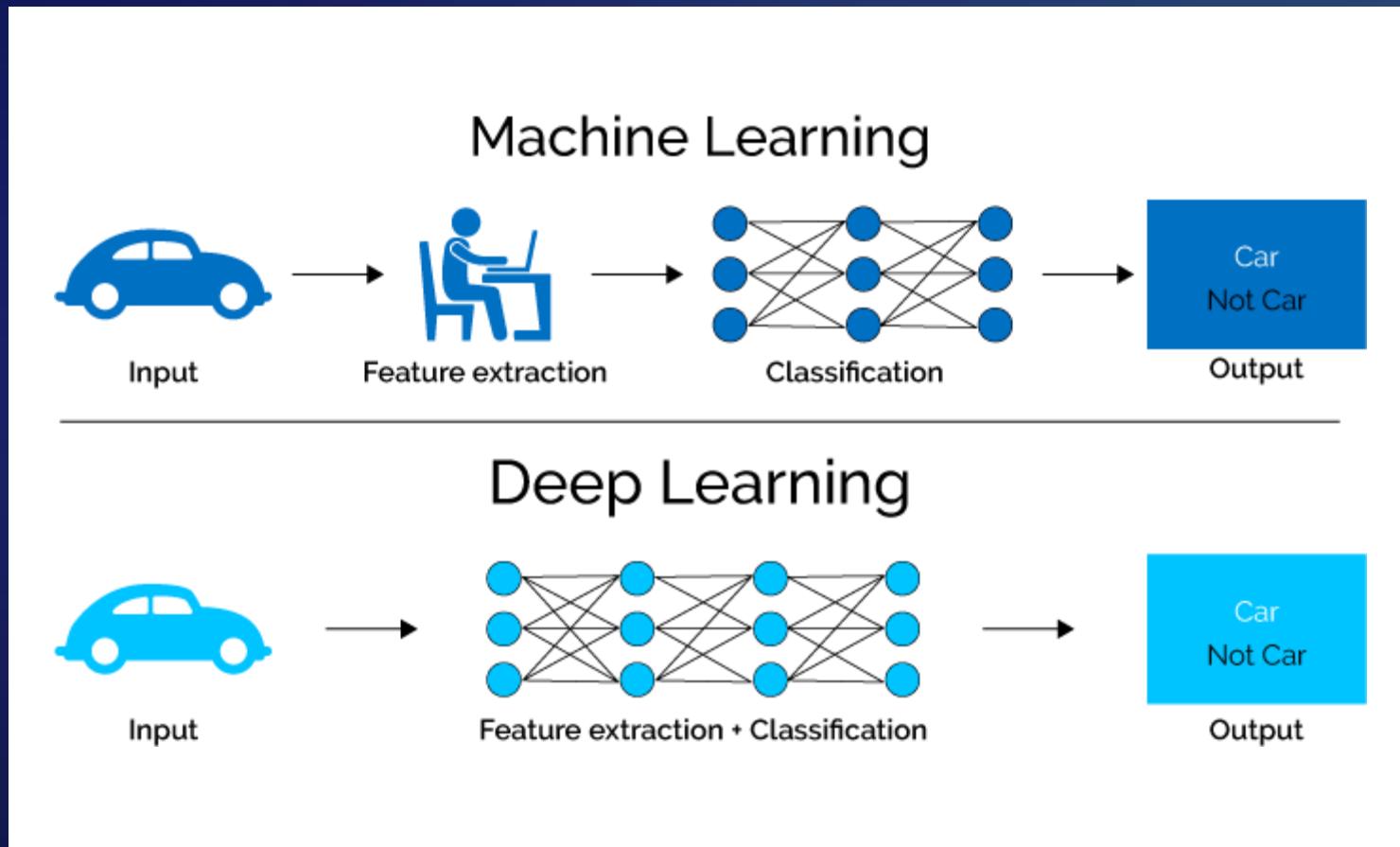


Artificial Intelligence

Machine
Learning

Deep
Learning

Contrasting Machine Learning with Deep Learning



Machine Learning in ArcGIS

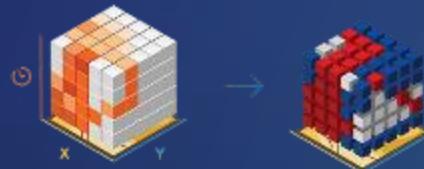
Classification

- Pixel & Object Based
- Image Segmentation
- Maximum Likelihood
- Random Trees
- Support Vector Machine



Clustering

- Spatially Constrained Multivariate Clustering
- Multivariate Clustering
- Density-based Clustering
- Hot Spot Analysis
- Cluster and Outlier Analysis
- Space Time Pattern Mining



Prediction

- Empirical Bayesian Kriging
- Areal Interpolation
- EBK Regression Prediction
- Ordinary Least Squares Regression and Exploratory Regression
- Geographically Weighted Regression



Deep Learning in ArcGIS

Data Preparation

- Label Objects
- Training Samples Manager
- Export Training Samples



Training

- Train Deep Learning Model
 - Object Detection
 - Object Classification
 - Pixel Classification
 - Instance Segmentation
- `arcgis.learn` module
(ArcGIS API for Python)

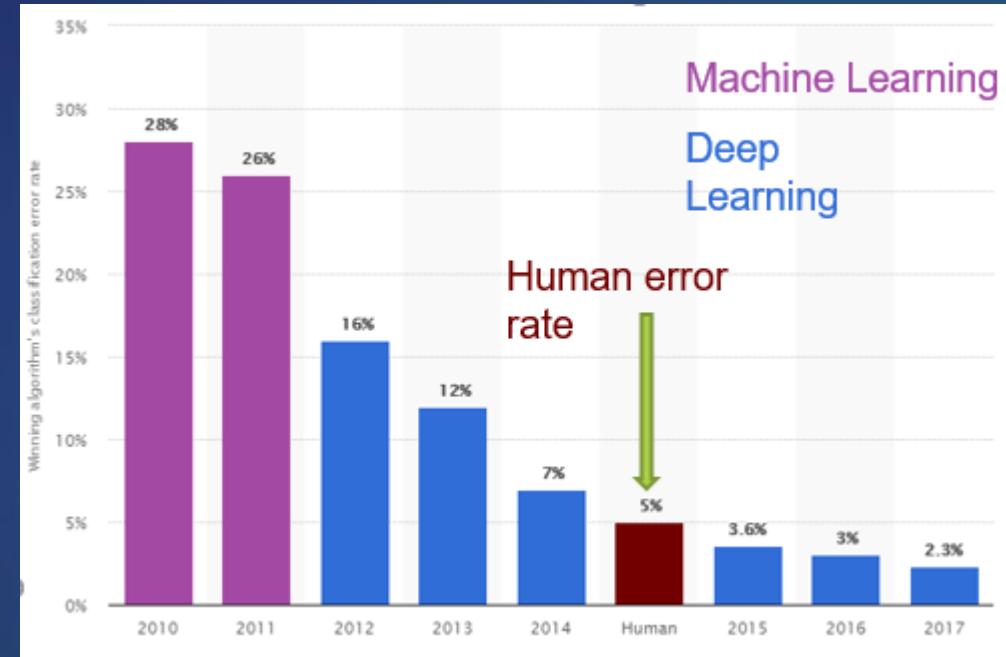


Inferencing

- Detect Objects
- Classify Pixels
- Classify Objects
- Non Maximum Suppression

Why use Deep Learning for Imagery

**Computer vision is now
almost as good, if not
better, than human vision**



ImageNet Visual Recognition Challenge error rate

Applications of Deep Learning to GIS

Impervious Surface Classification



Coconut Tree Detection



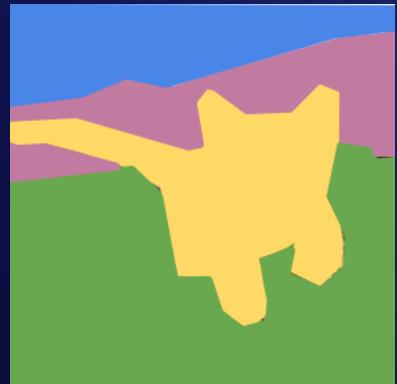
Building Footprint Extraction



Damaged House Classification



End to End Deep Learning — Wide spectrum of deep learning models



Pixel Classification



Object Detection



Instance Segmentation



Image Classification

Deep Learning with Imagery in ArcGIS

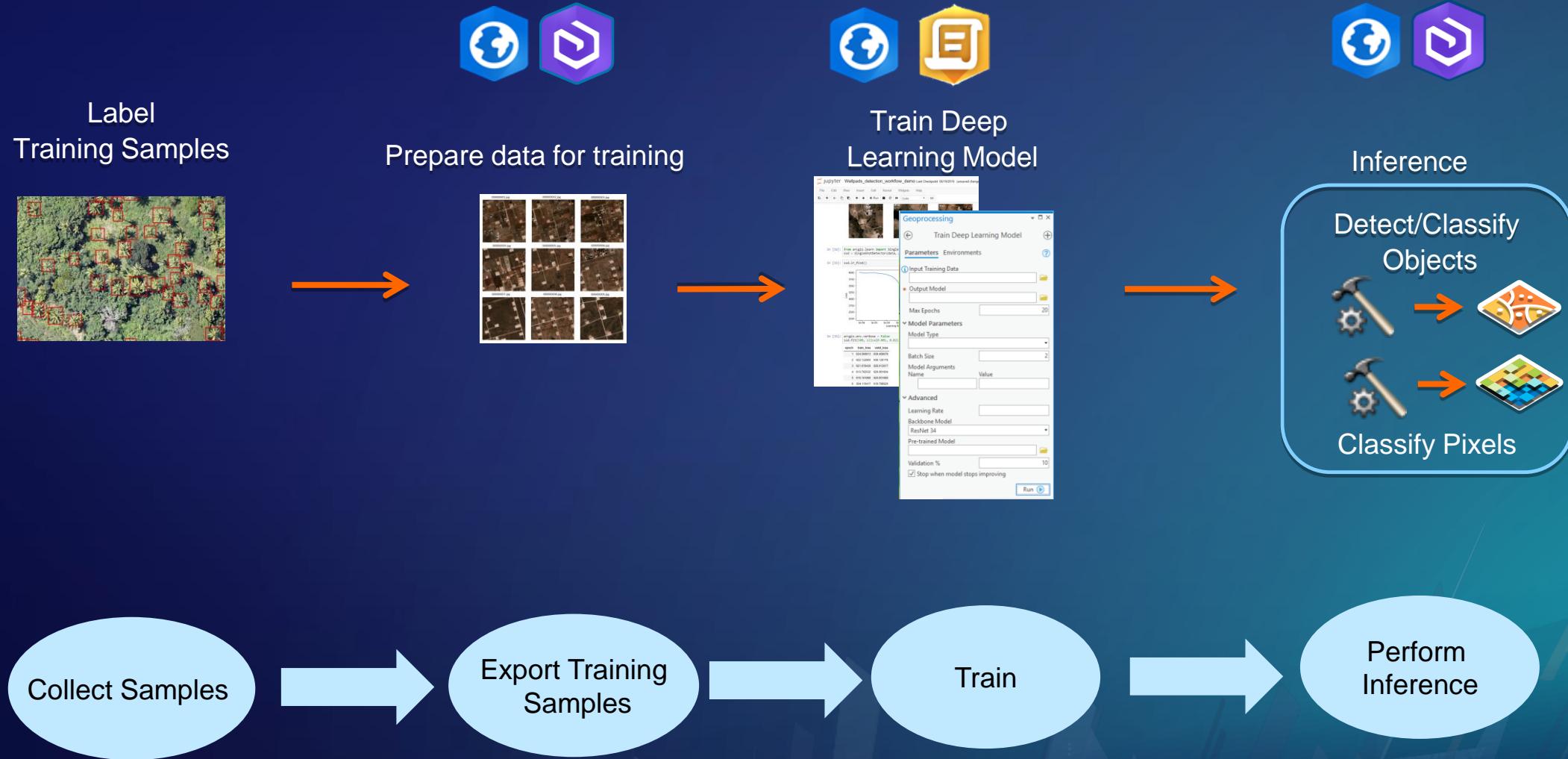
ArcGIS supports end-to-end deep learning workflows

- Tools for:
 - Labeling training samples
 - Preparing data to train models
 - Training Models
 - Running Inferencing
- Supports the key imagery deep learning categories
- Supported environments
 - ArcGIS Pro
 - Map Viewer
 - ArcGIS Notebooks/Jupyter Notebook



Part of ArcGIS Image Analyst
Run distributed on ArcGIS Image Server

Deep Learning Workflow in ArcGIS



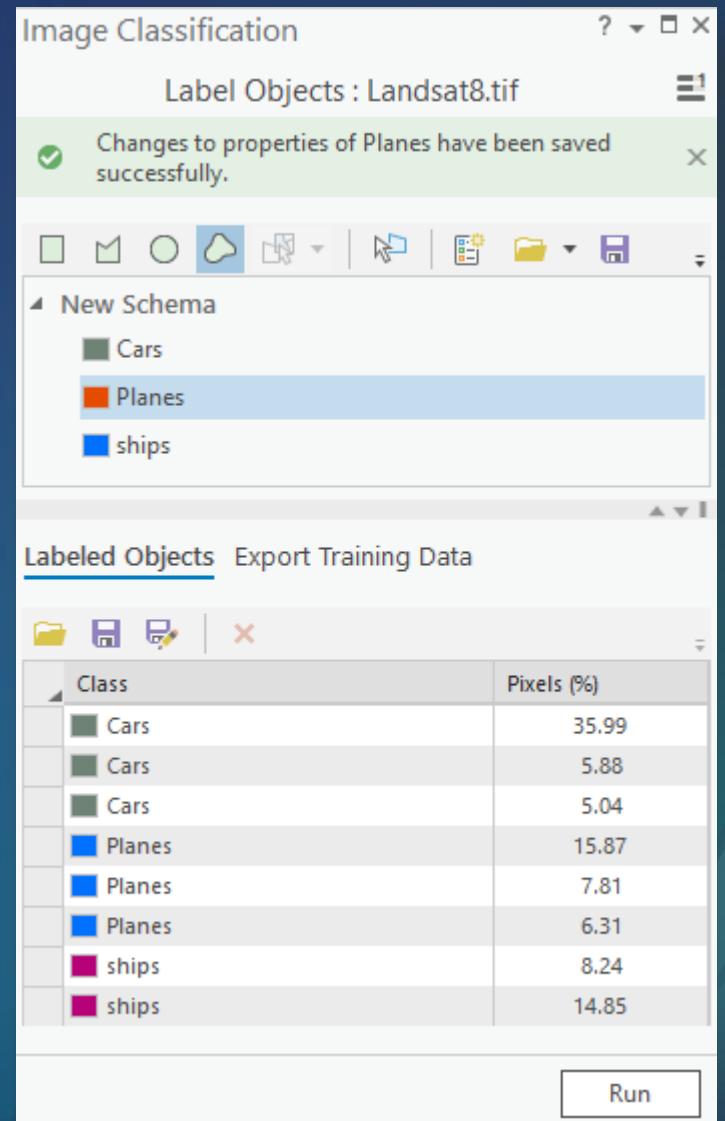
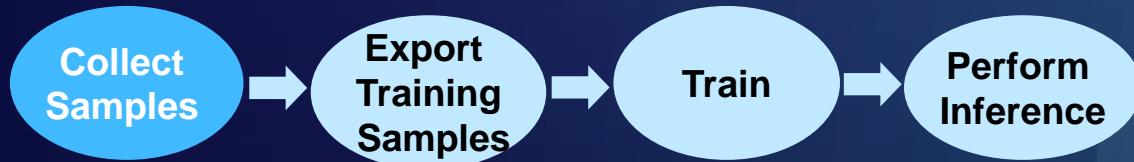
Collect Training Samples / Label data

- Different methods

- **Label Objects for Deep Learning – ArcGIS Pro (2.5)**
- Training sample manager – ArcGIS Pro
- Feature editing
 - ArcGIS Pro
 - Map Viewer
 - JS Web Apps

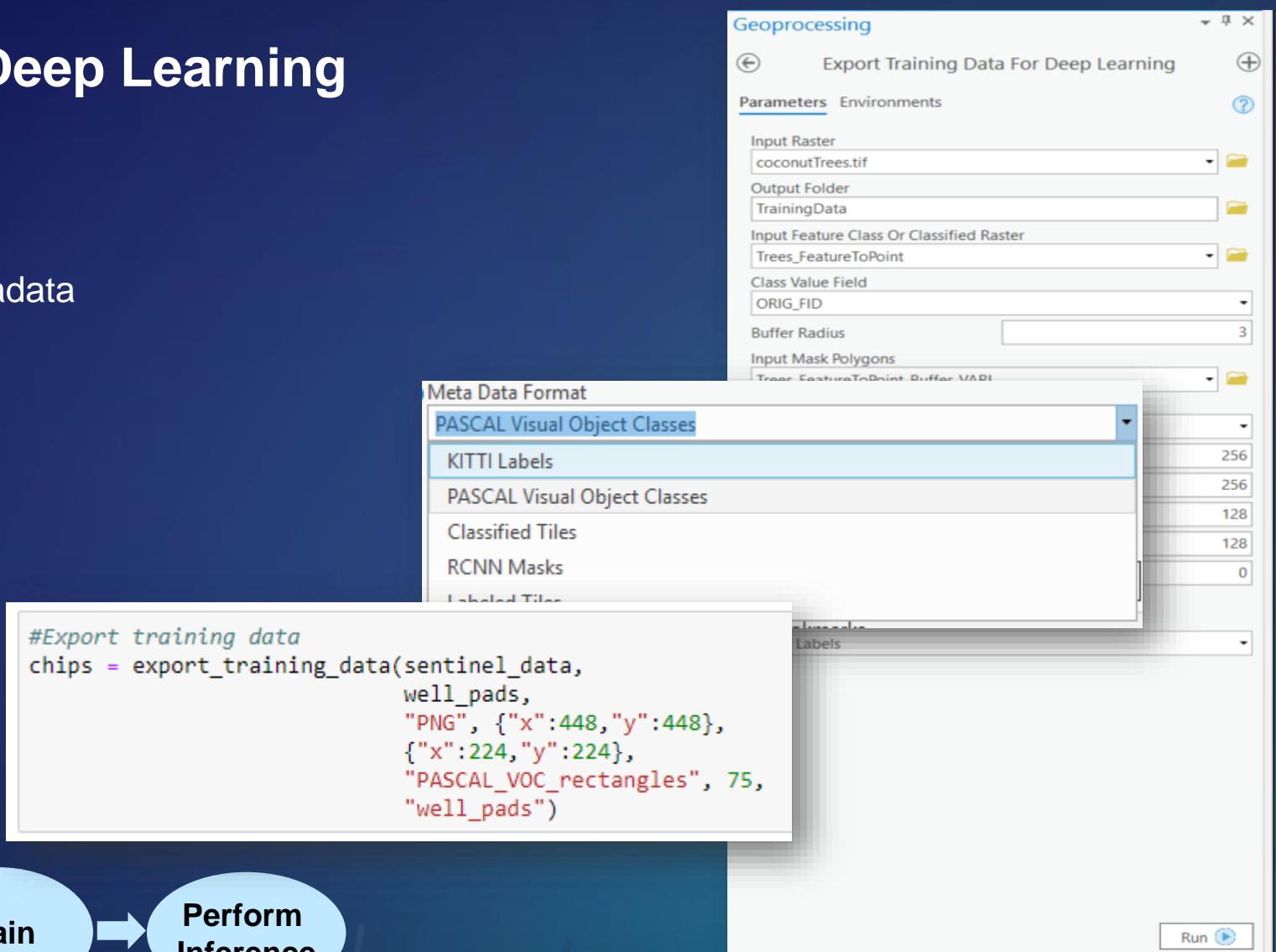
- Different data models

- Feature class (local single user)
- Feature services (collaborative experience)
- Classified thematic rasters



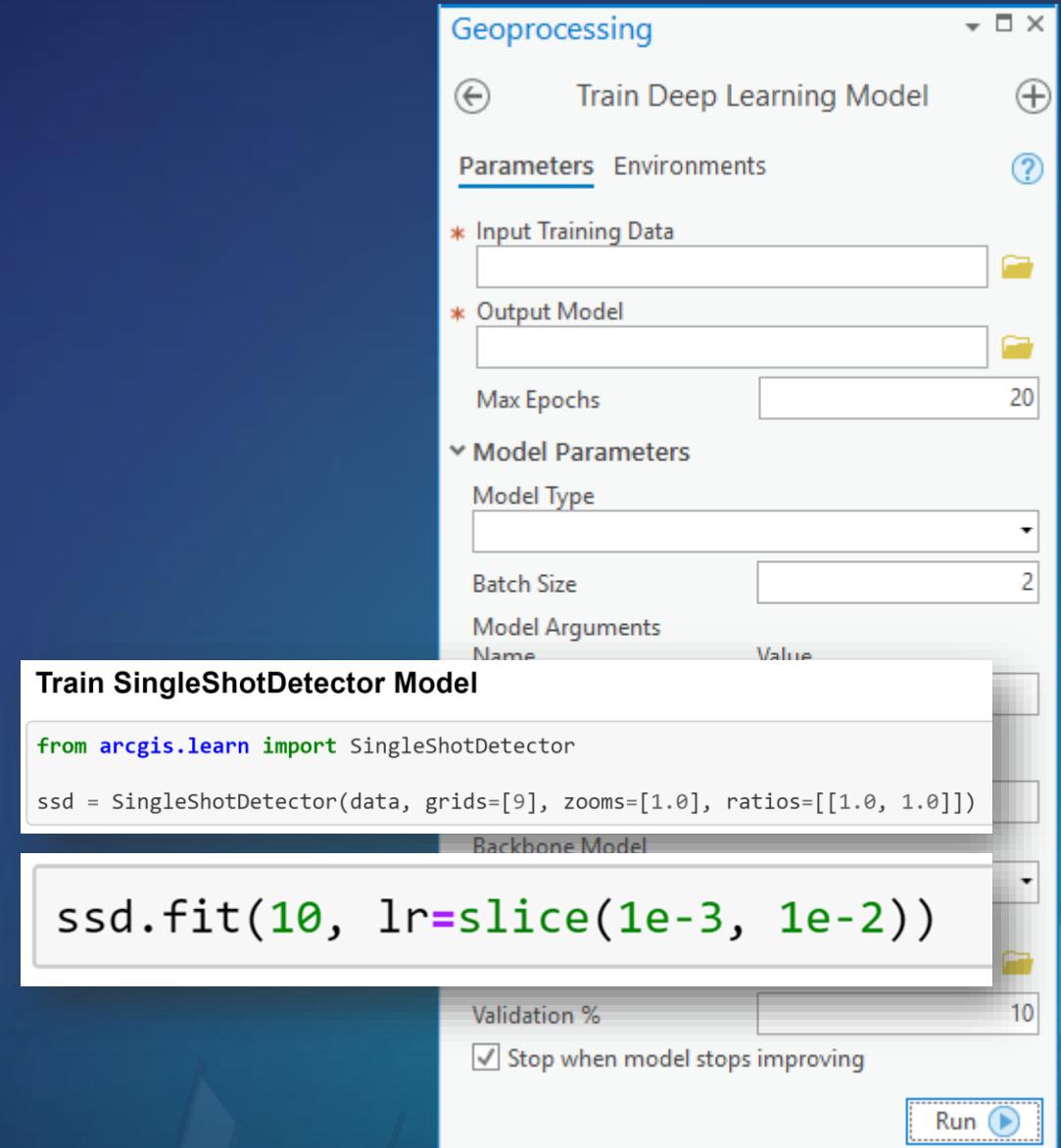
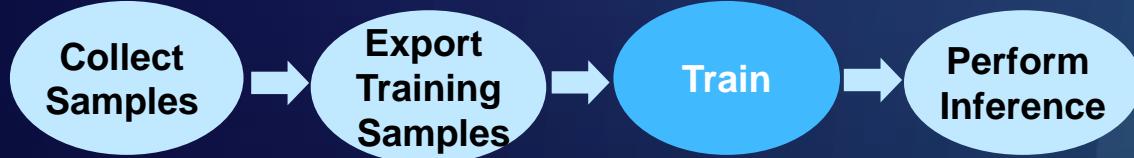
Export Training Data for Deep Learning

- Exports samples to training images
- Images have associated labels/metadata
- Writes out an ECD
- Used as inputs for model training
- Supports various formats



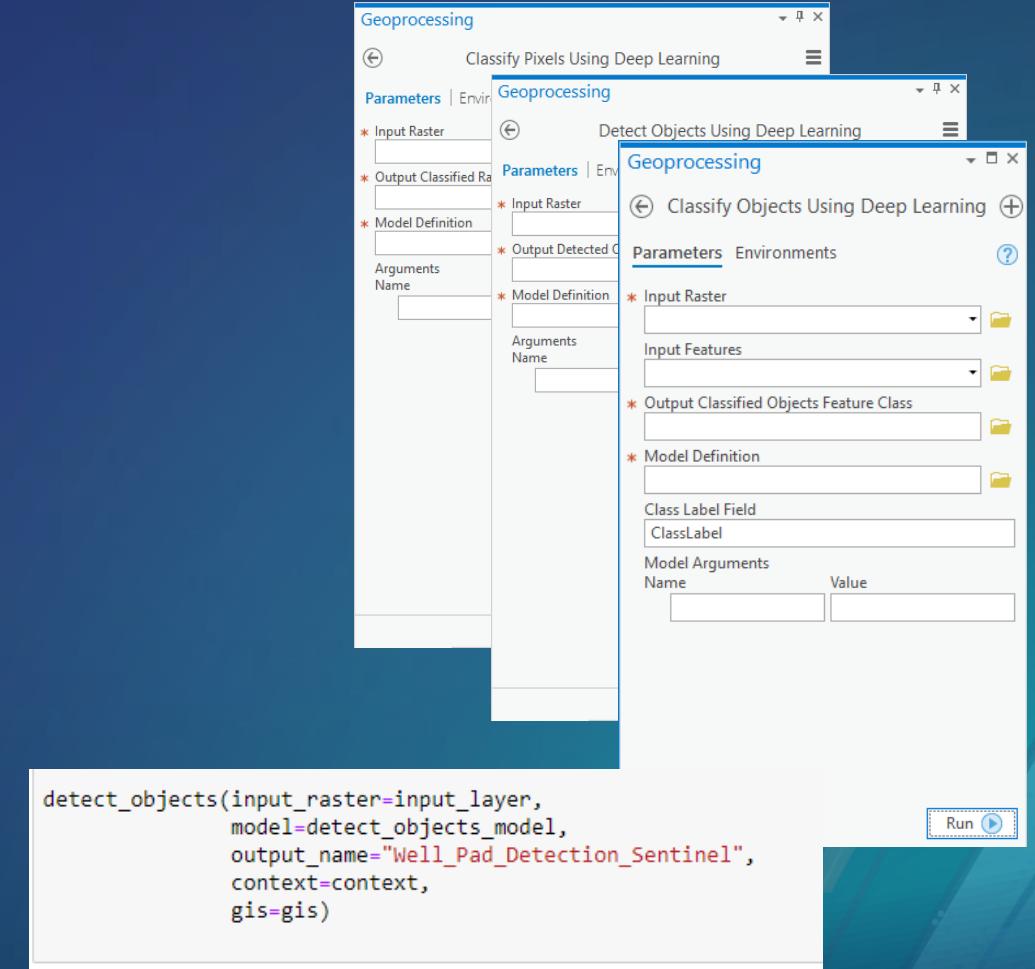
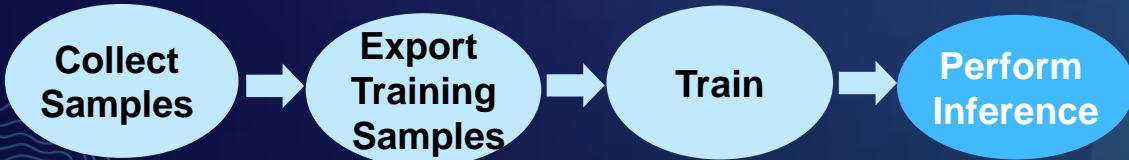
Train Deep Learning Model

- ArcGIS Pro and ArcGIS API for Python supports training
- ArcGIS Pro “Train Deep Learning Model” tool
- `arcgis.learn` module in ArcGIS API for Python
- Supported Models:
 - Object Detection - SSD, RetinaNet, MaskRCNN
 - Object Classification – Feature classifier
 - Pixel Classification – UNET, PSPNet
- External Deep Learning Frameworks
 - TensorFlow
 - CNTK...



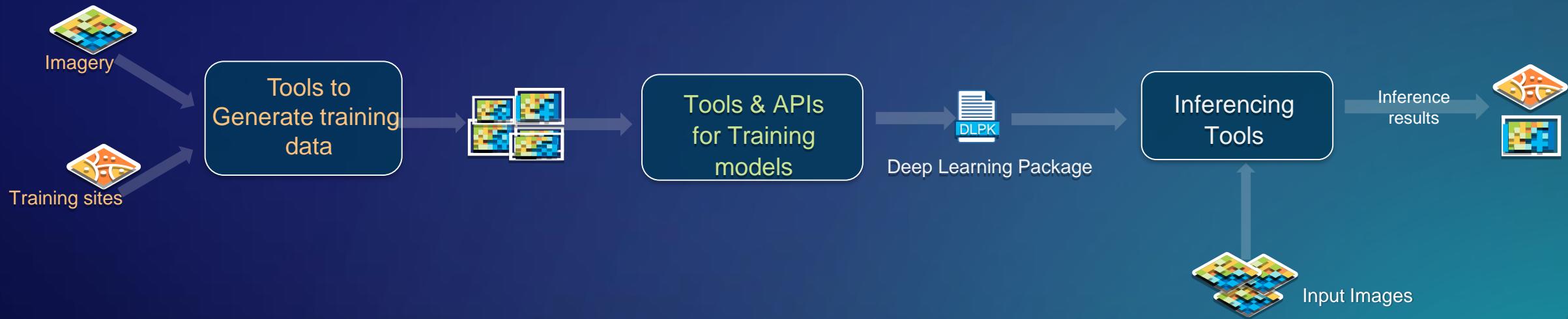
Perform Inference

- Run on desktop and enterprise
- Parallel processing using enterprise
- Types of inferencing
 - Object detection
 - Classify objects
 - Pixel classification



ArcGIS – Deep Learning Workflow

End-to-end deep learning workflow



Tools to generate training samples

- Image Analyst in ArcGIS Pro
- Image Server on ArcGIS Enterprise

Model Training

- ArcGIS Pro
- Notebooks

Inferencing

- Image Analyst in ArcGIS Pro
- Image Server on ArcGIS Enterprise

Deep Learning Package

- Zip with a .dlpk file extension
 - Created by Train Deep Learning Model tool and `arcgis.learn` (ArcGIS API for Python)
- Contents of the dlpk
 - Model definition file (.emd)
 - Deep learning model file (framework specific)
 - Python Raster Function (.py, optional if using an out-of-the-box model)
- Can be shared across your organization

The screenshot shows the ArcGIS Content page with a blue header bar. In the top right corner, there are two buttons: 'Add Item' and 'Create'. Below them is a dropdown menu with options: 'From my computer', 'From the web', 'An application', and 'All My Content' (which is highlighted). On the far right, there is a user icon labeled 'admin'.

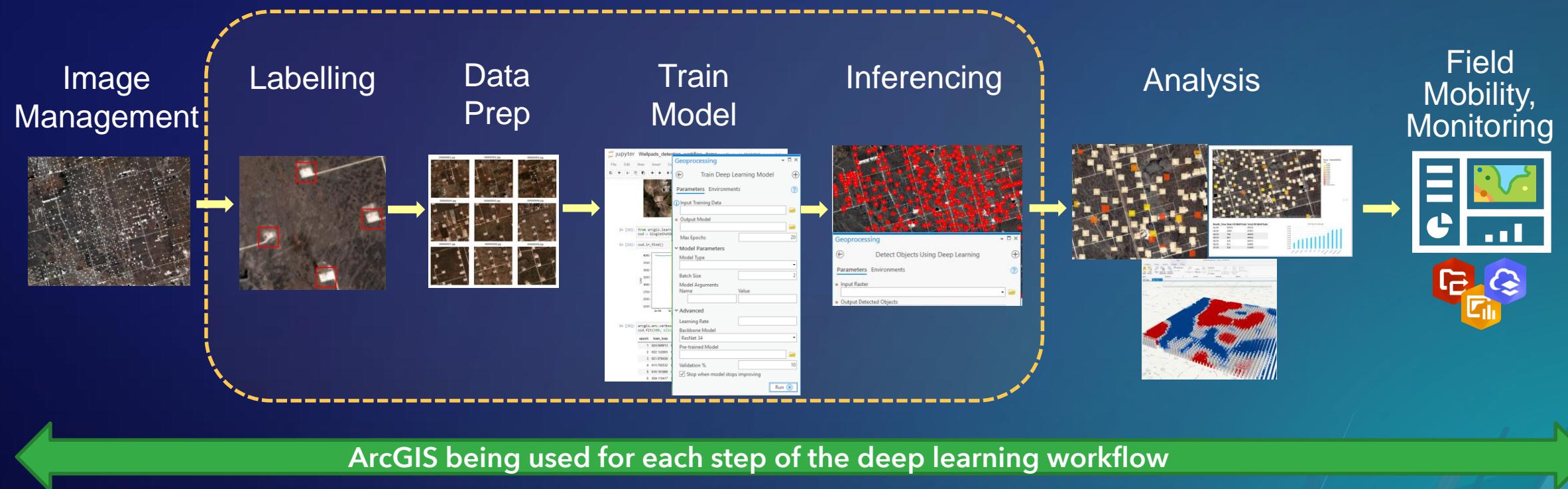
The main content area has a title 'TreeDetection_DLPackage' with a edit icon. Below the title is a thumbnail image of a neural network diagram. To the right of the thumbnail is a summary section with the text 'Add a brief summary about the item.' and 'Deep Learning Package by admin'. It also shows the item was 'Created: Jan 28, 2019' and 'Updated: Jan 28, 2019' with 'Number of Downloads: 0'. At the bottom of the summary section is a 'Description' field with the placeholder 'Add an in-depth description of the item.'

Supported Capabilities

	Labelling	Exporting Training Samples	Training Deep Learning Models	Inferencing
ArcGIS Pro	✓	✓	✓	✓
ArcGIS API for Python		✓	✓	✓
Map Viewer	✓	✗	✗	✓
Image Server		✓	✓	✓

Deep Learning Workflow in ArcGIS

End-to-end from raw imagery to structured information products



ArcGIS API for Python

[Install the API](#)

Version 1.6.1 · May 16, 2019

[Home](#) [Guide](#) [Sample Notebooks](#) [API Reference](#) [Community](#)

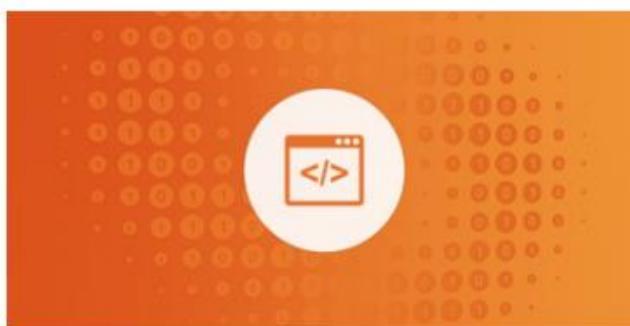
A powerful Python library for spatial analysis, mapping and GIS

ArcGIS API for Python is a Python library for working with maps and geospatial data, powered by web GIS. It provides simple and efficient tools for sophisticated vector and raster analysis, geocoding, map making, routing and directions, as well as for organizing and managing a GIS with users, groups and information items. In addition to working with your own data, the library enables access to ready to use maps and curated geographic data from Esri and other authoritative sources. It also integrates well with the scientific Python ecosystem and includes rich support for Pandas and Jupyter notebook.

[Install the API](#) | [Get started](#) | [View samples](#)

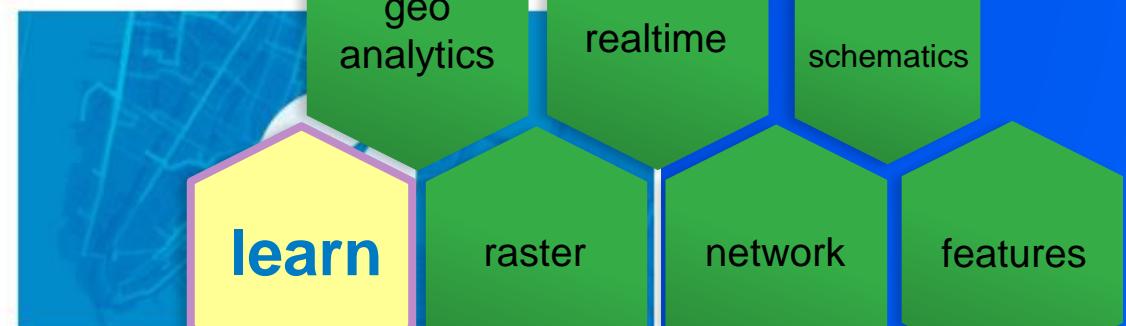
Understand your GIS

This "hello world" style notebook shows how to get started with the GIS and visualize its contents.



Manage your GIS

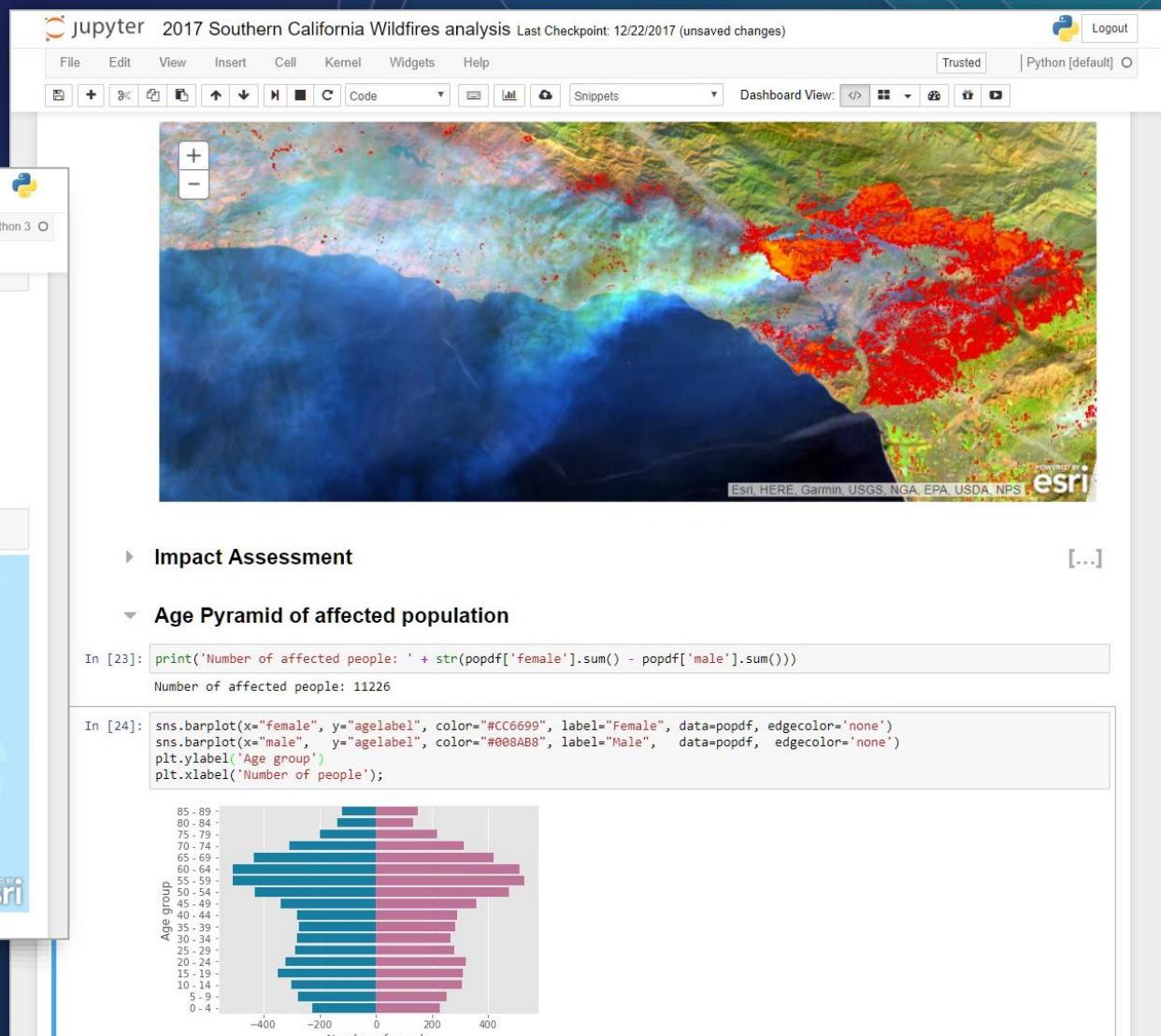
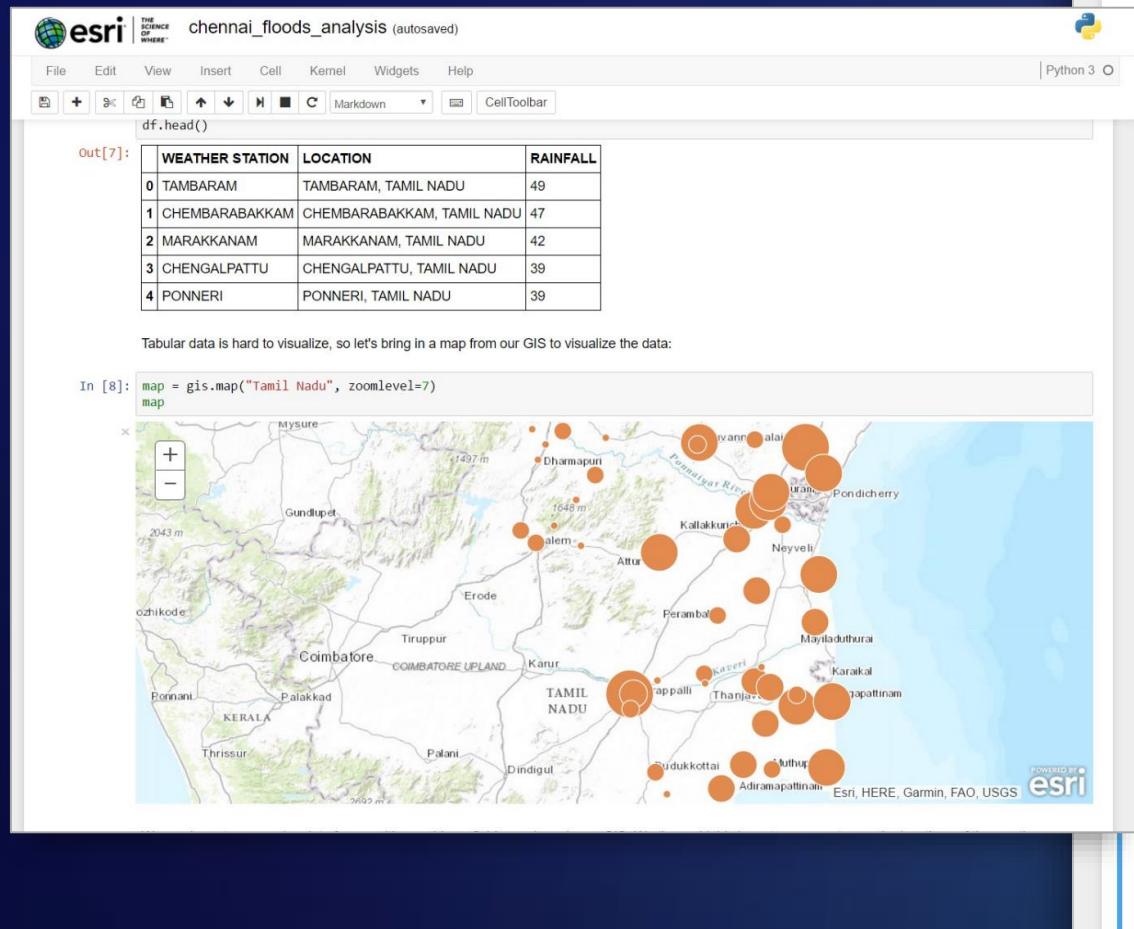
The ArcGIS API for Python provides APIs and samples for ArcGIS Online administrators to manage their online



Perform Spatial Analysis

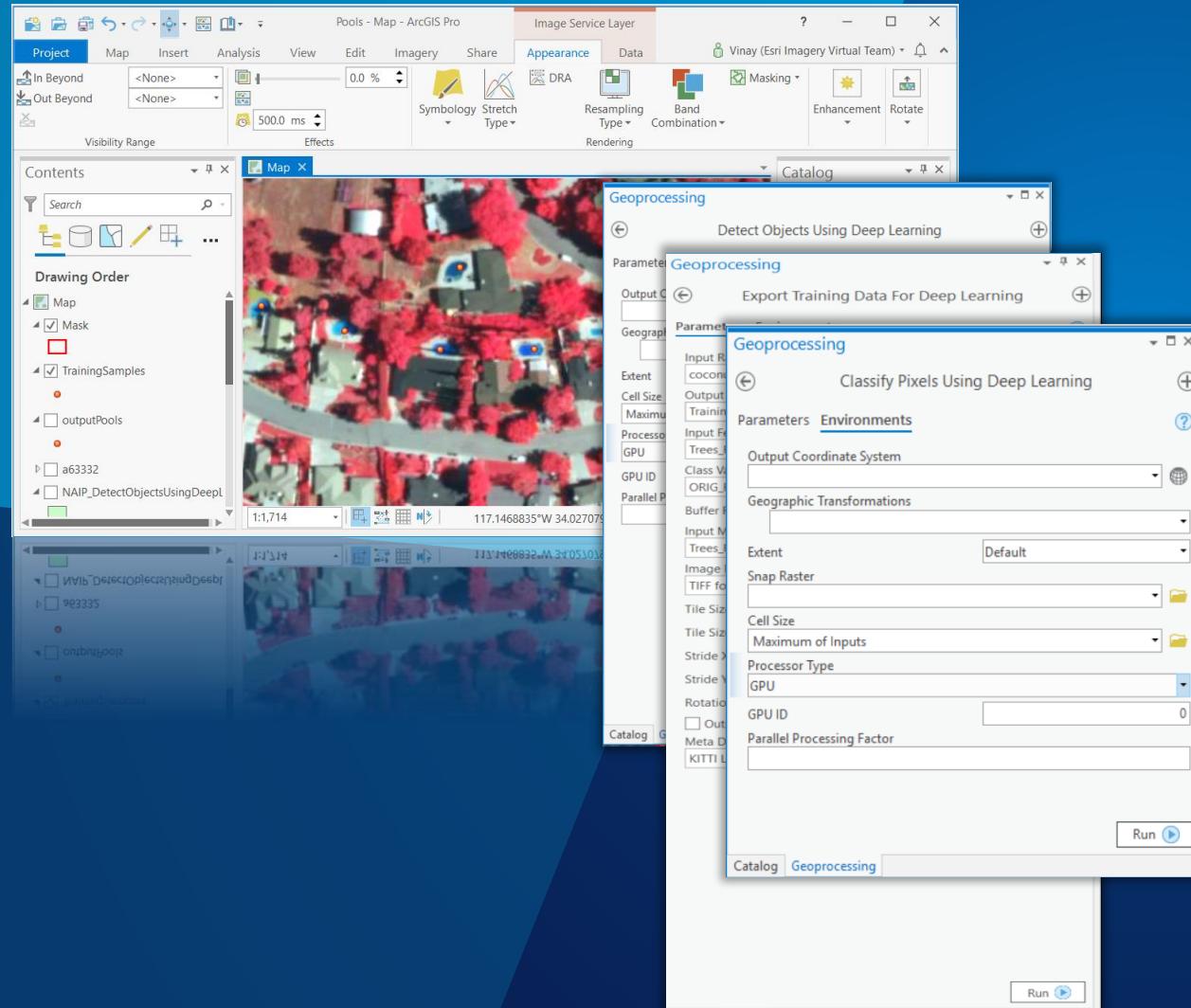
Call sophisticated spatial analysis tools that work with online content, using a few lines of code.

ArcGIS + Notebooks = ❤

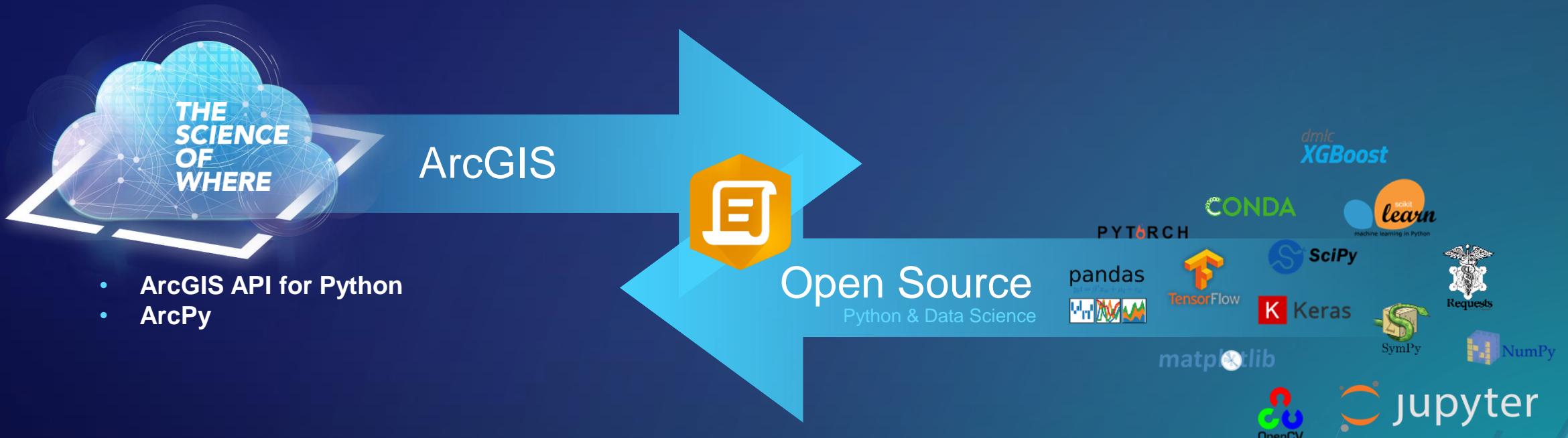


Demo

Combining Pro with Notebooks Kiln Detection in India



ArcGIS Notebooks sits at the intersection of ArcGIS and open data science



ArcGIS API for Python

arcgis.learn module

The `arcgis.learn` module in **ArcGIS API for Python** enables Python developers and data scientists to **easily train and use deep learning models with a simple, intuitive API.**



Before

- Dozens of lines of Code
- Installing External DL Frameworks
- HARD!

Train Models



After

Train SingleShotDetector Model

```
from arcgis.learn import SingleShotDetector
ssd = SingleShotDetector(data, grids=[9], zooms=[1.0], ratios=[[1.0, 1.0]])
```

```
ssd.fit(10, lr=slice(1e-3, 1e-2))
```

```

Labels = np.array(labels).T
print("Labels shape", Labels.shape)
print("Labels", Labels)

8. Convert Label to One Hot Vector
In [40]: print(label.shape)
Out[40]: (146, 7)
print(keras.utils.to_categorical(label), num_classes=7)
print(label.shape)
146 204 205
146 205 205 7

9. Define IoU Metric
In [41]: def mean_iou(true, pred):
    for t in np.unique(true):
        for p in np.unique(pred):
            if t == p:
                score = tF.metric_mean_iou(true, pred, t, p)
            else:
                score = tf.math.logical_and(
                    tf.math.equal(true, t),
                    tf.math.equal(pred, p))
                score = tf.reduce_mean(score)
    return score

10. Define Custom Loss Function
In [42]: def log_weights(a):
    weights = np.zeros((a, 1))
    weight = a * np.log(a) / np.sum(a, axis=0)
    weights[a] = weight
    return weights

def weighted_categorical_crossentropy(y_true, y_pred):
    # Since prediction is in the scale [0, 1] and each sample sum to 1
    # we need to get the log of the probability of each sample due to
    # the cross entropy loss function to prevent Nan's and Inf's
    # to do it we multiply the weight of each sample by the log of the
    # prediction plus one minus the weight times one minus the prediction
    # to calculate loss and weight loss
    # We apply a small epsilon to prevent divide by zero
    loss = -log_weights(y_true) * weights
    loss = -loss.sum(axis=0)
    return loss

11. Set Parameters before Training
In [43]: img_width = 256
img_weight = 256
grid_size = 9
num_classes = 7
Land Cover Classes

```

ArcGIS API for Python

Not just “training”!

Data Preparation

```
arcgis.learn.export_training_data  
arcgis.learn.prepare_data
```

Training DL Models

```
arcgis.learn.SingleShotDetector  
arcgis.learn.UnetClassifier  
arcgis.learn.FeatureClassifier  
arcgis.learn.PSPNetClassifier  
arcgis.learn.RetinaNet  
arcgis.learn.MaskRCNN
```

Model Management

```
arcgis.learn.list_models  
arcgis.learn.Model  
Model.install  
Model.uninstall  
Model.query_info
```

Inference APIs

```
arcgis.learn.detect_objects  
arcgis.learn.classify_pixels  
arcgis.learn.classify_objects
```

Advantages of arcgis.learn

- Closely integrated with ArcGIS
 - Directly consumes exported training data from ArcGIS (no custom preprocessing)
 - Saved models (DLPKs) are directly usable in ArcGIS
 - No custom postprocessing of model output
 - Image space to map space conversion automatically handled
 - Preserves symbology
- Consistent API (`prepare_data()`, `fit()`, `save()` to train model, `show_batch`/`show_results` to visualize)
- Performs data augmentations suitable for satellite imagery
- Extensible using fast.ai transforms, custom loss functions, model backbones
- Fast.ai goodies: Automatic learning rate finder, transfer learning, early stopping, checkpointing, one-cycle learning
- Model metrics, sample results and training details are stored along with the model
- Padding support, multi-gpu training, CPU/GPU support, predict on videos, multispectral imagery*

Things you can do today with arcgis.learn

Object Detection, Pixel Classification, Feature Classification, Instance Segmentation

Damaged Structures



Building Footprints



Land Cover



Catfish



Brick Kilns



Roads



Oil Pads



Palm trees



Refugee Camps



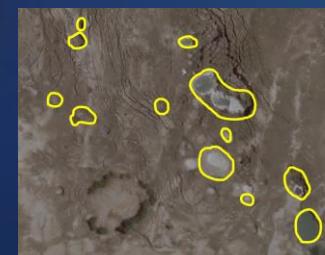
Surface-to-Air missile (SAM) sites



Swimming Pools



Sinkholes



Training Model using arcgis.learn

Types of Deep Learning Models & their applications in GIS

Image Classification

Assign a label to a given image



Cat

Object Classification

Assign a label to a given feature



Undamaged



Damaged

Models (from torchvision):

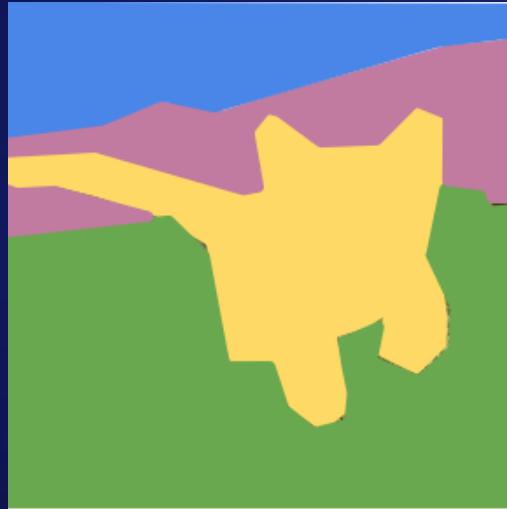
- Inception
- ResNet
- VGG...

Applications:

- Damaged building classification
- Clean or 'green' pools...

Semantic Segmentation

Assign a label to each pixel



- Cat
- Ground
- Sky

Pixel Classification



- Turf/Grass
- Building
- Water

Models:

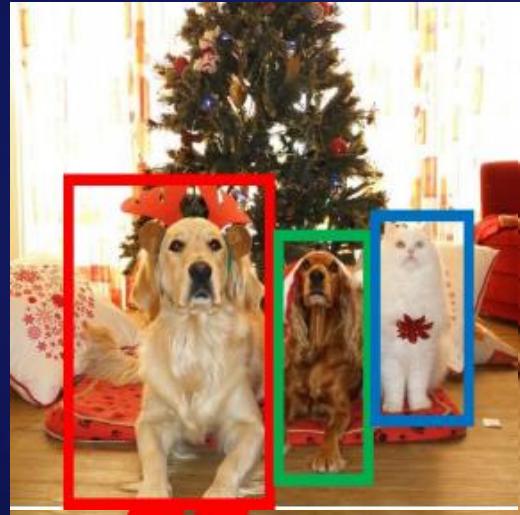
- UNetClassifier
- PSPNetClassifier

Applications:

- Land Cover Classification
- Pervious/Impervious mapping...

Object Detection

Find objects and their location (bounding boxes)



Models:

- SingleShotDetector
- RetinaNet

Applications:

- Detect trees, cars, airplanes, ...

Instance Segmentation

Find objects and their *precise locations* (masks or polygonal features)

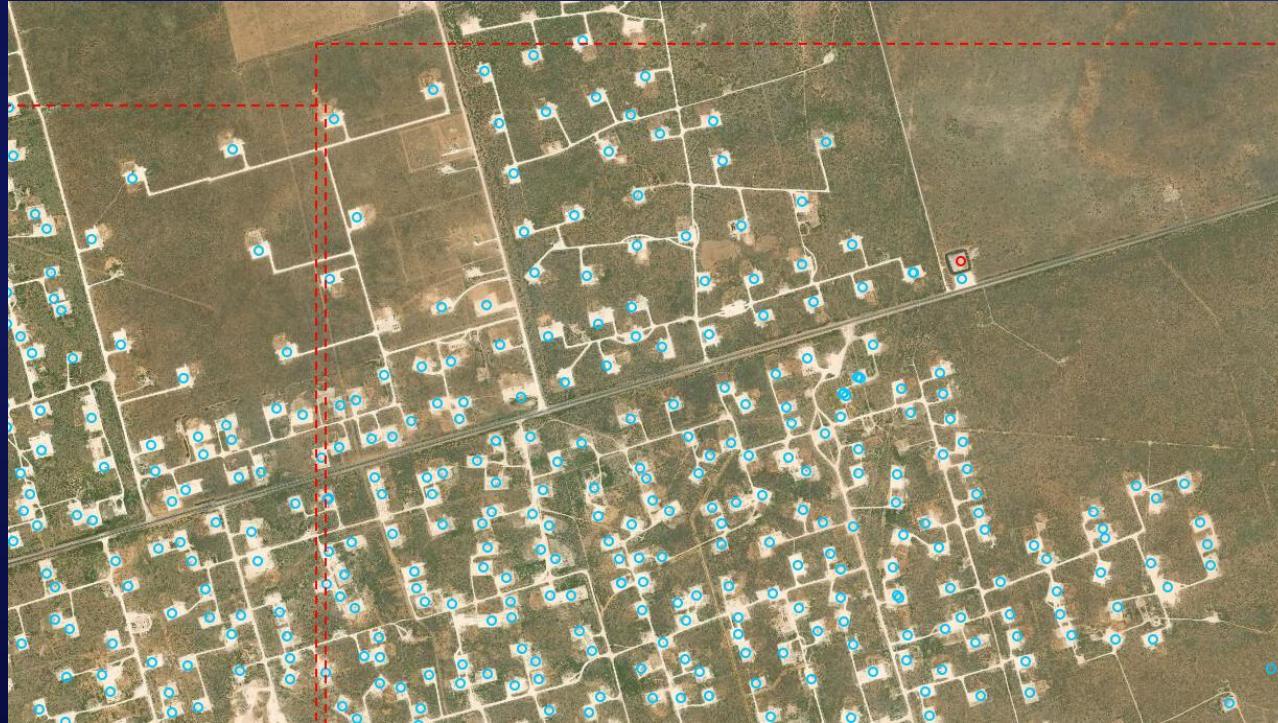


Models:

- MaskRCNN

Applications:

- Building footprint extraction



Demo

Continue Pool Detection Demo



Demo

Land Use/Land Cover



Demo

Features from Text

ArcGIS Enterprise for Scaling Deep Learning

- Leverage Raster Analytics to scale inferencing
- All desktop inferencing tools are accessible through enterprise
- Clients to invoke distributed inferencing – map viewer, pro, notebooks
- Multi GPU support
- Requires the ArcGIS Image Server license

ArcGIS Enterprise Deep Learning Tools / Services

Pro + Server Tools

Server Only Tools

[ExportTrainingDataforDeepLearning](#)

Uses a remote sensing image to convert labeled vector or raster data into deep learning training datasets. The output is a folder of image chips and a folder of metadata files.

[DetectObjectsUsingDeepLearning](#)

Runs a trained deep learning model on an input raster to produce a feature class containing the objects it finds. The features can be bounding boxes or polygons around the objects found, or points at the centers of the objects.

[ClassifyPixelsUsingDeepLearning](#)

Runs a trained deep learning model on an input raster to produce a classified raster with each valid pixel having a class label assigned.

[ClassifyObjectsUsingDeepLearning](#)

Runs a trained deep learning model on an input raster and feature class to produce a classified feature class

[TrainDeepLearningModel](#)

Enables training deep learning models

[QueryDeepLearningModelInfo](#)

Extracts the model specific settings from the model package item or model definition file.

[InstallDeepLearningModel](#)

Installs the model package item from portal to the Raster Analysis Image Server.

[UninstallDeepLearningModel](#)

Uninstalls the model package from portal to the Raster Analysis Image Server

[ListDeepLearningModels](#)

Lists all the installed model packages on the Raster Analysis Image Server

ArcGIS REST Services Directory

[Home](#) > [services](#) > [System](#) > [RasterAnalysisTools \(GPServer\)](#)

[JSON](#) | [SOAP](#)

System/RasterAnalysisTools (GPServer)

Service Description: The RasterAnalysisTools service is used by...

Tasks:

- [DetectObjectsUsingDeepLearning](#)
- [ClassifyPixelsUsingDeepLearning](#)
- [ExportTrainingDataforDeepLearning](#)
- [QueryDeepLearningModelInfo](#)
- [InstallDeepLearningModel](#)
- [UninstallDeepLearningModel](#)
- [ListDeepLearningModels](#)
- [TrainDeepLearningModel](#)
- [ClassifyObjectsUsingDeepLearning](#)

New at ArcGIS API for Python v1.7.1

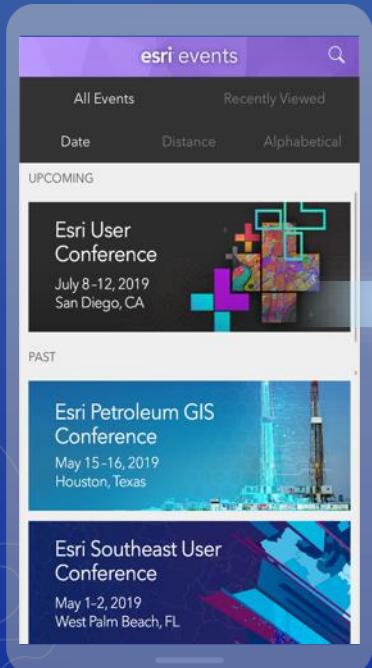
- `arcgis.learn` module enhancements
 - Single band support
 - Multispectral Imagery
- New Sample Notebooks and Learn lessons for deep learning workflows
- New model types (DeepLab, PSPNet, RetinaNet, MaskRCNN...)
- Multi GPU inference using enterprise

MORE IS COMING AT v1.8.0 and BEYOND!

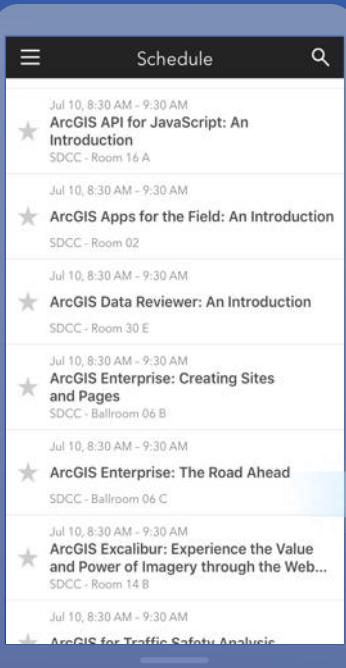


Please Share Your Feedback in the App

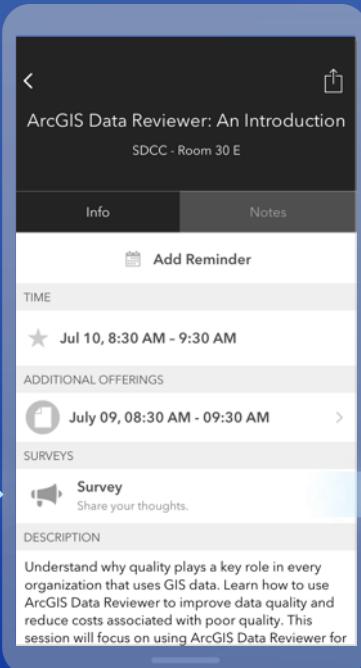
Download the Esri Events app and find your event



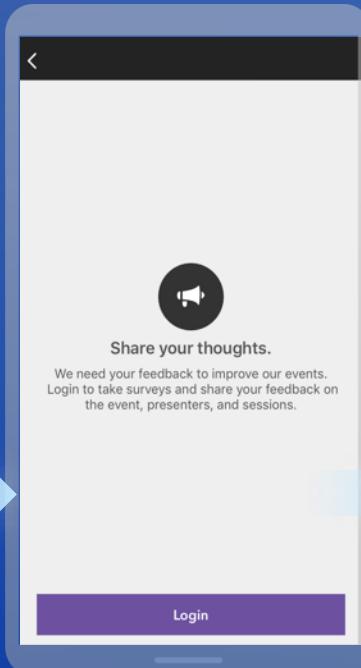
Select the session you attended



Scroll down to "Survey"



Log in to access the survey



Complete the survey and select "Submit"

