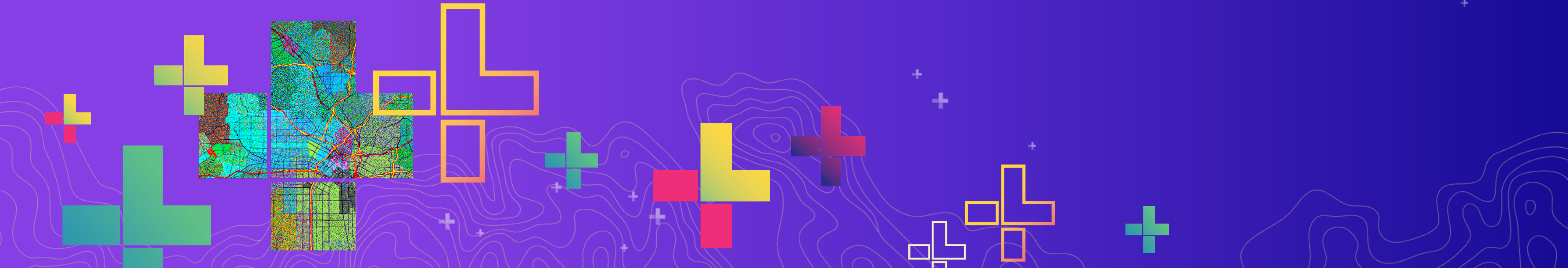
Geospatial Deep Learning with ArcGIS

Rohit Singh
Vinay Viswambharan

2020 ESRI DEVELOPER SUMMIT | Palm Springs, CA



Session Overview

- AI, Machine Learning & Deep Learning
- Deep Learning Workflow
- Training Models
  - ArcGIS Pro
  - arcgis.learn
- Types of models and their applications
- Scalable deep learning with Image Server
- Learning Resources

Caffe

Object Tracking

CNTK

Object Detection

Artificial Intelligence

PyTorch

scikit-learn

Computer Vision

Random Forest

Machine Learning

Neural Networks

Natural Language Processing

Cognitive Computing

TensorFlow

Data Science

GeoAI

Deep Learning

fast.ai

Keras

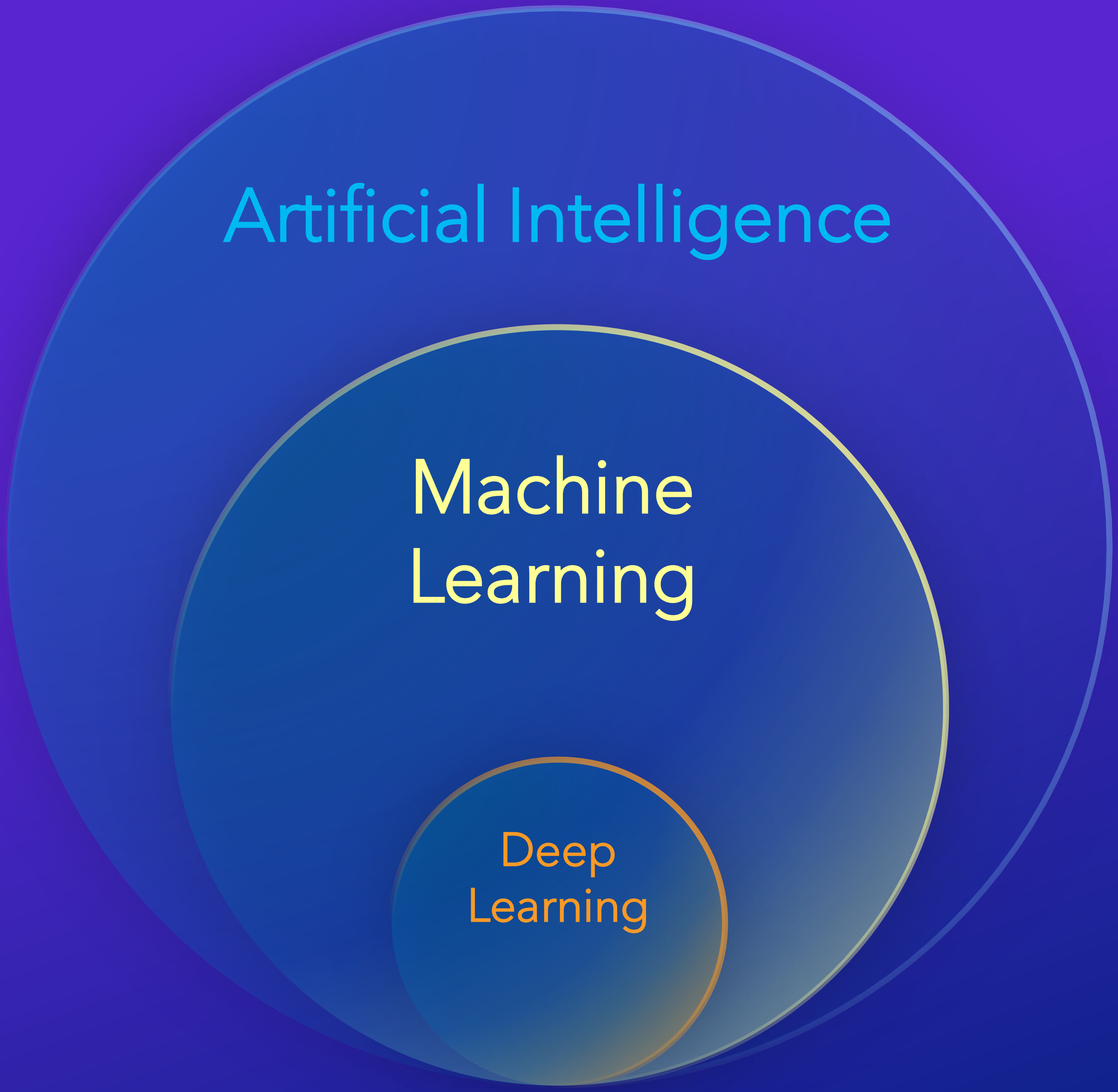
Dimensionality Reduction

Support Vector Machines

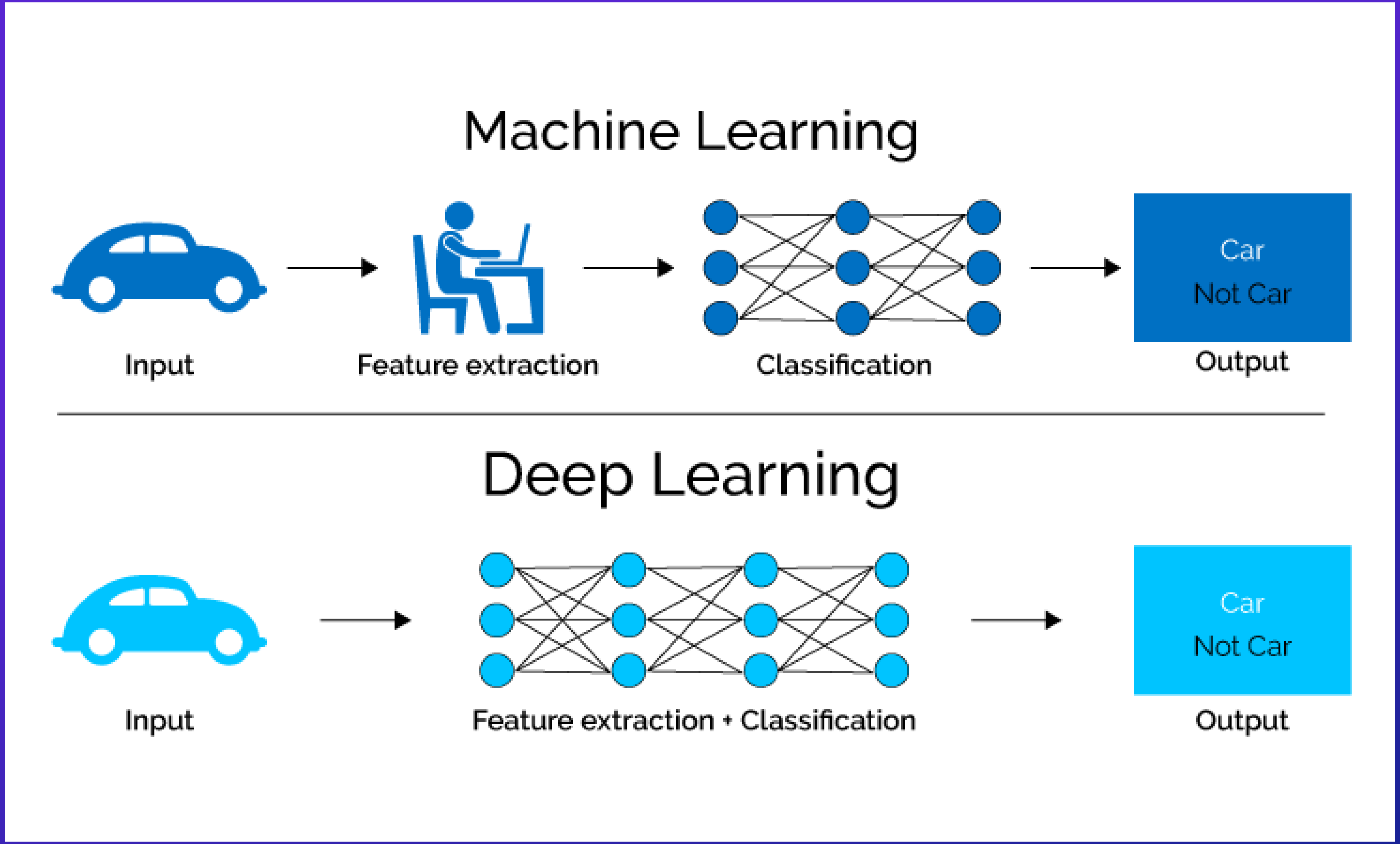
Artificial Intelligence

Machine Learning

Deep Learning


Artificial Intelligence
Machine Learning
Deep Learning

Contrasting Machine Learning with Deep Learning



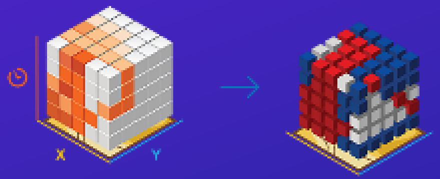
Machine Learning in ArcGIS

Classification

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

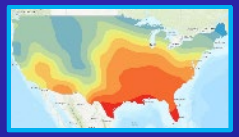
Clustering

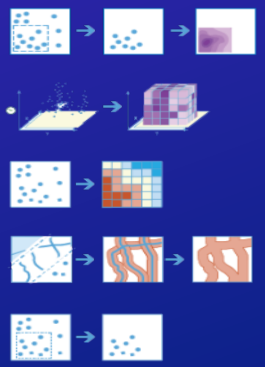
- Spatially Constrained Multivariate Clustering
- Multivariate Clustering
- Density-based Clustering
- Hot Spot Analysis
- Cluster and Outlier Analysis
- Space Time Pattern Mining



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

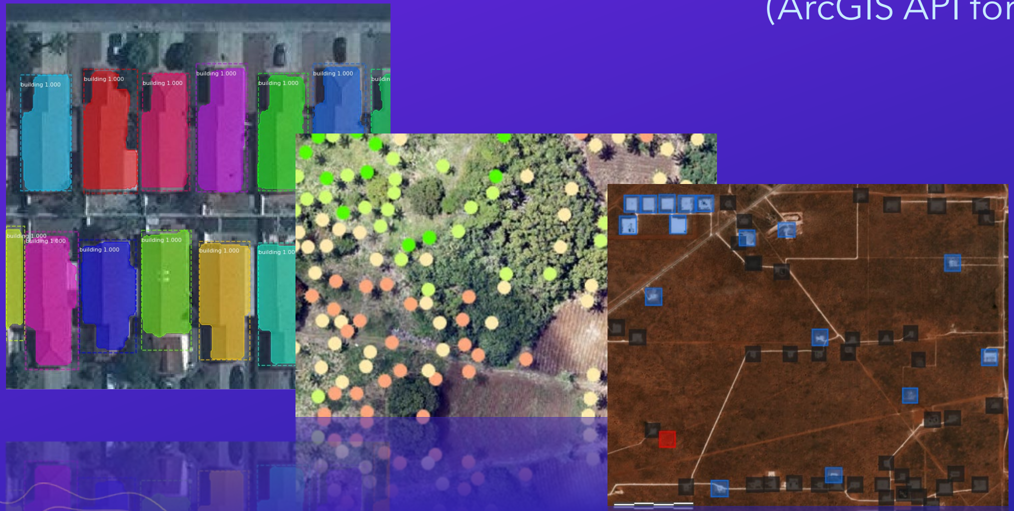
- Label Objects
- Training Samples Manager
- Export Training Samples

Training

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

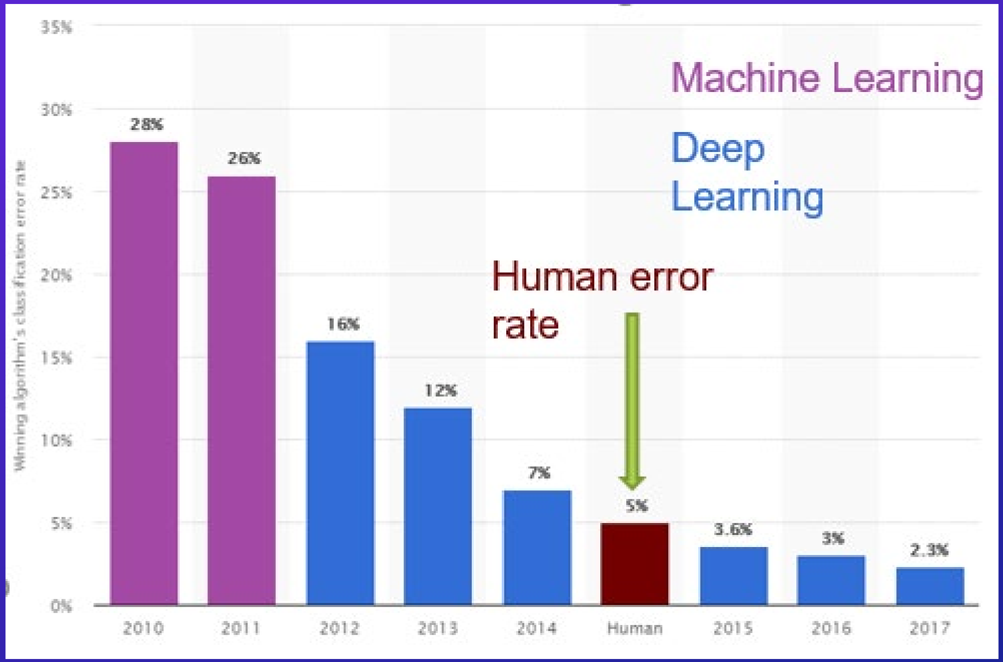
Inferencing

- 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

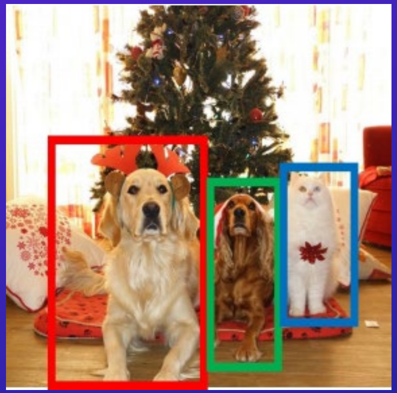
Coconut Tree Detection

Building Footprint Extraction

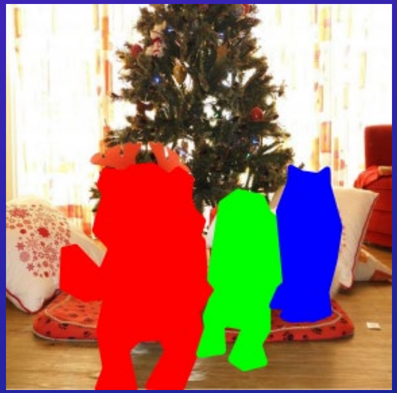
Damaged House Classification

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

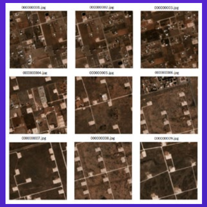
Part of ArcGIS Image Analyst

Run distributed on ArcGIS Image Server

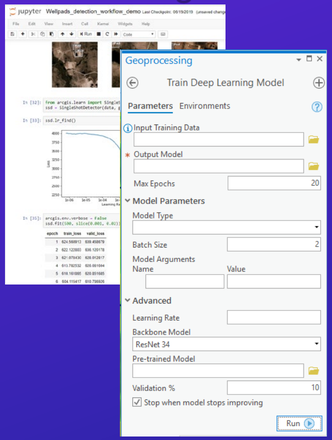
Deep Learning Workflow in ArcGIS

Label
Training Samples

Prepare data for training



Train Deep
Learning Model


jupyter
Wellpads_detection_workflow_demo
Last Checkpoint: 06/19/2019 (unsaved change
File
Edit
View
Insert
Cell
Kernel
Widgets
Help
Run
Code
Loss
Learning Ra

```
In [32]: from arcgis.learn import SingleS
         ssd = SingleShotDetector(data, g
In [33]: ssd.lr_find()
In [35]: arcgis.env.verbose = False
         ssd.fit(500, slice(0.001, 0.02))
```

epoch	train_loss	valid_loss
1	624.560913	639.458679
2	622.122803	636.120178
3	621.078430	628.012817
4	613.782532	626.081604
5	610.161865	620.851685
6	604.115417	610.798926

Geoprocessing

Train Deep Learning Model

Parameters Environments

Input Training Data:

* Output Model:

Max Epochs: 20

Model Parameters

Model Type:

Batch Size: 2

Model Arguments
Name Value

Advanced

Learning Rate:

Backbone Model: ResNet 34

Pre-trained Model:

Validation %: 10

☑ Stop when model stops improving

Run

Inference

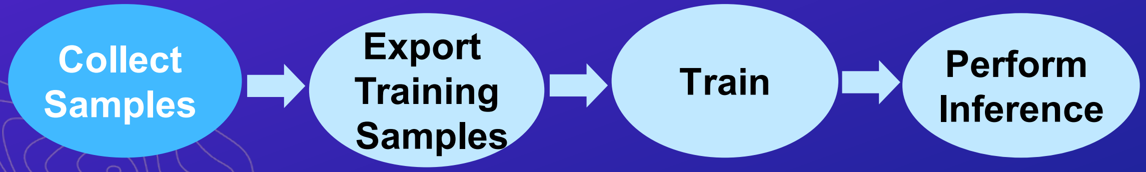
Detect/Classify
Objects

Classify Pixels

Collect Samples → Export Training Samples → Train → Perform Inference

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



Collect Samples → Export Training Samples → Train → Perform Inference

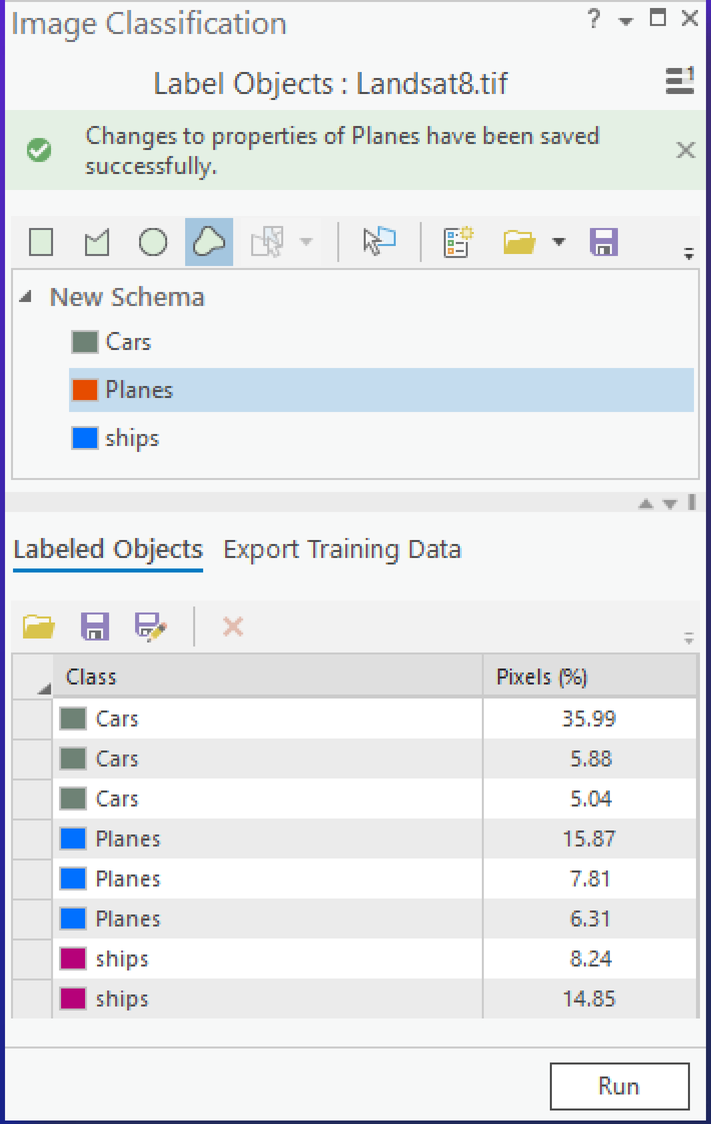


Image Classification

Label Objects : Landsat8.tif

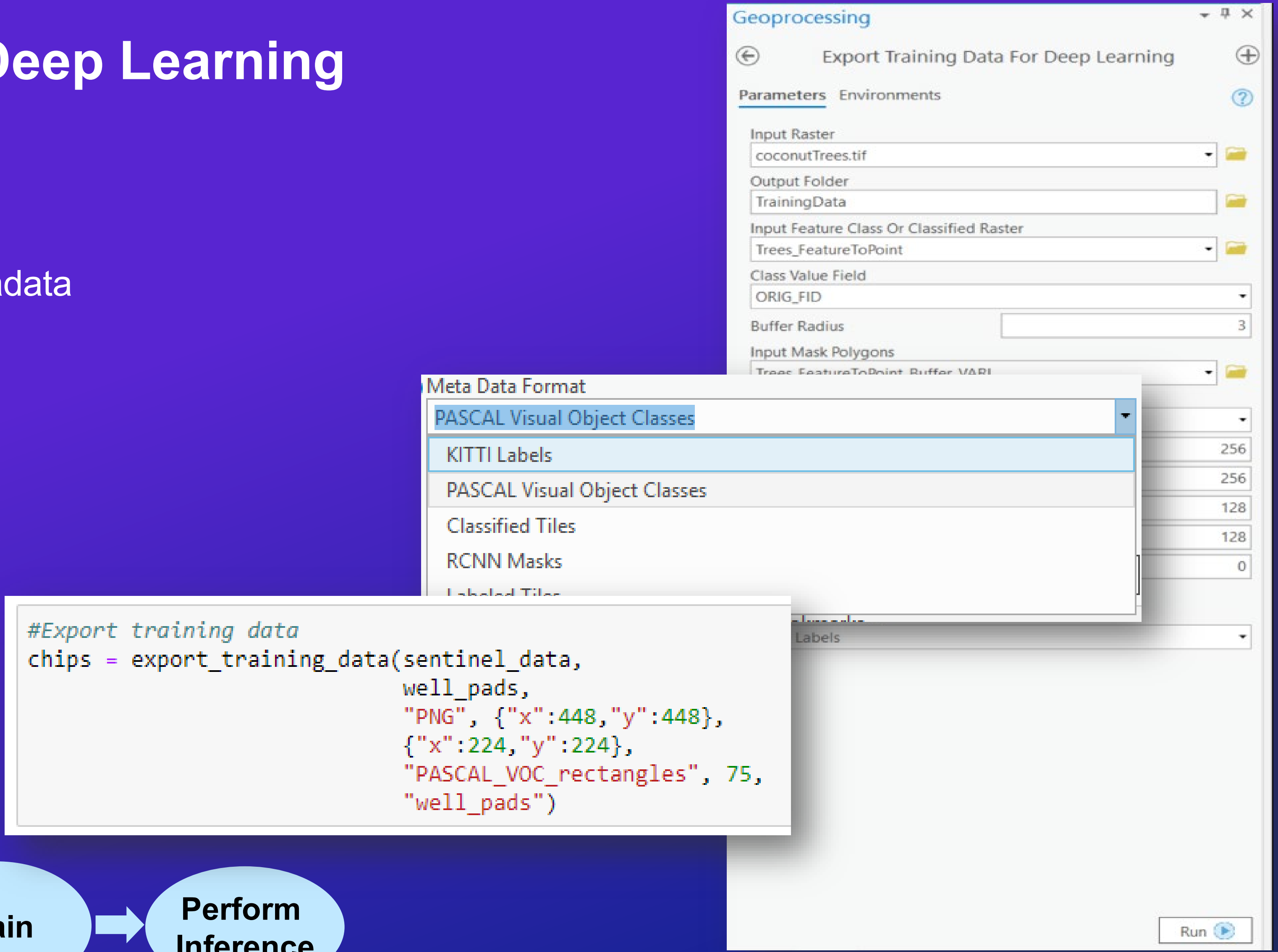
Changes to properties of Planes have been saved successfully.

New Schema
- Cars
- Planes
- ships

Class	Pixels (%)
Cars	35.99
Cars	5.88
Cars	5.04
Planes	15.87
Planes	7.81
Planes	6.31
ships	8.24
ships	14.85

Export Training Data for Deep Learning

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



Geoprocessing

Export Training Data For Deep Learning

Parameters | Environments

Input Raster: coconutTrees.tif

Output Folder: TrainingData

Input Feature Class Or Classified Raster: Trees_FeatureToPoint

Class Value Field: ORIG_FID

Buffer Radius: 3

Input Mask Polygons: Trees_FeatureToPoint_Buffer_VARI

Meta Data Format: PASCAL Visual Object Classes

- KITTI Labels
- PASCAL Visual Object Classes
- Classified Tiles
- RCNN Masks
- Labeled Tiles

256

256

128

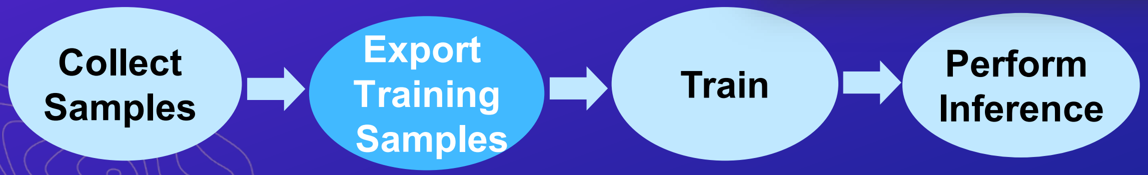
128

0

Labels

Run

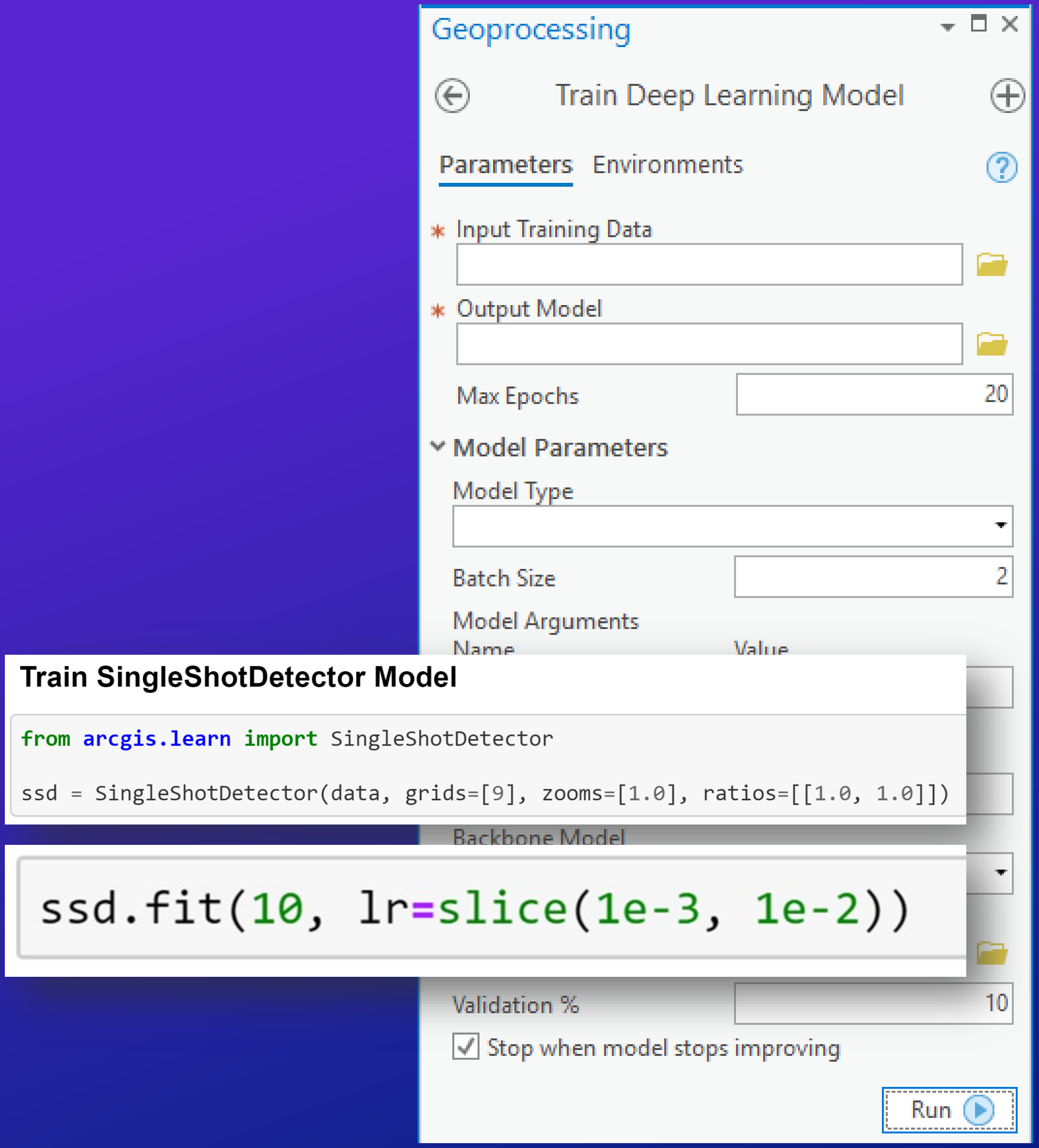
```
#Export training data
chips = export_training_data(sentinel_data,
                             well_pads,
                             "PNG", {"x":448,"y":448},
                             {"x":224,"y":224},
                             "PASCAL_VOC_rectangles", 75,
                             "well_pads")
```



Collect Samples → Export Training Samples → Train → Perform Inference

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…



* Input Training Data

* Output Model

Max Epochs: 20

Model Parameters

Model Type

Batch Size: 2

Model Arguments

Name	Value

Backbone Model

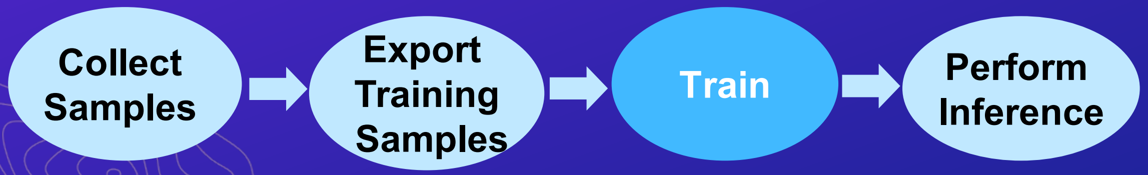
Validation %: 10

☑ Stop when model stops improving

Train SingleShotDetector Model

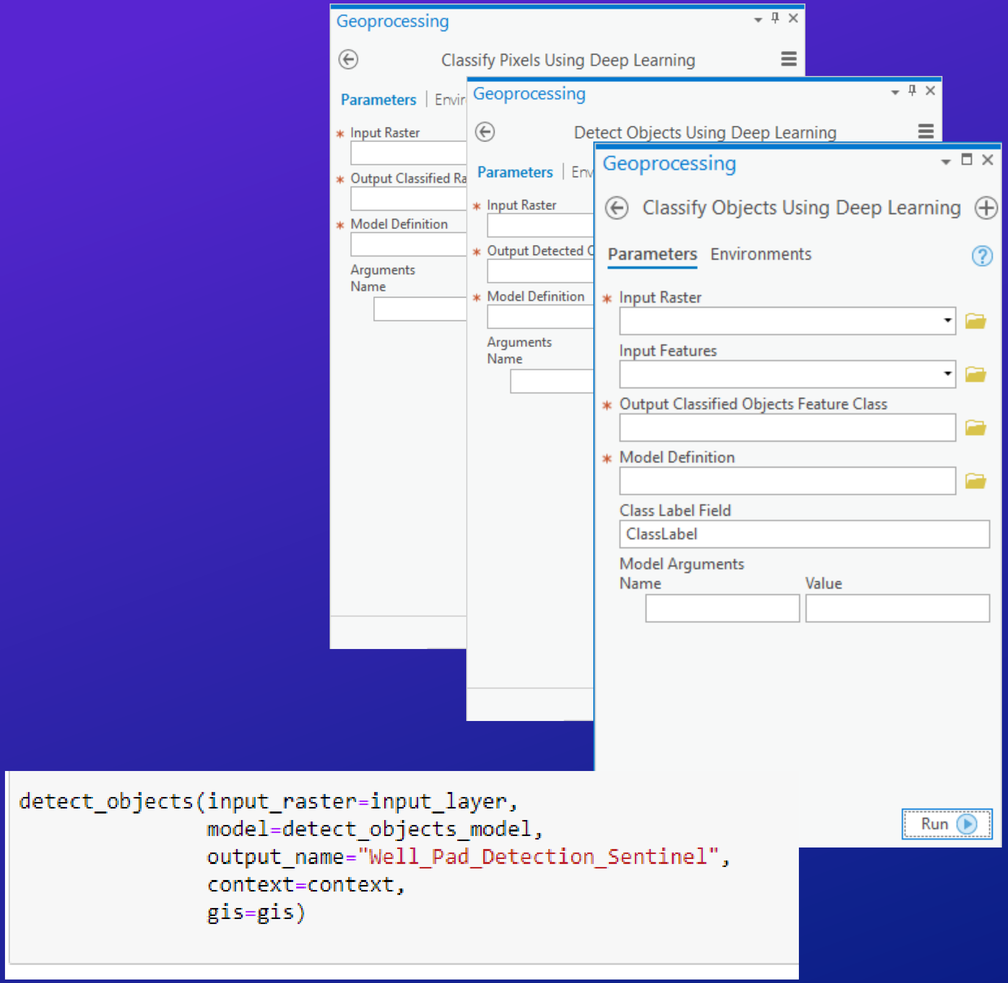
```
from arcgis.learn import SingleShotDetector

ssd = SingleShotDetector(data, grids=[9], zooms=[1.0], ratios=[[1.0, 1.0]])
```

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



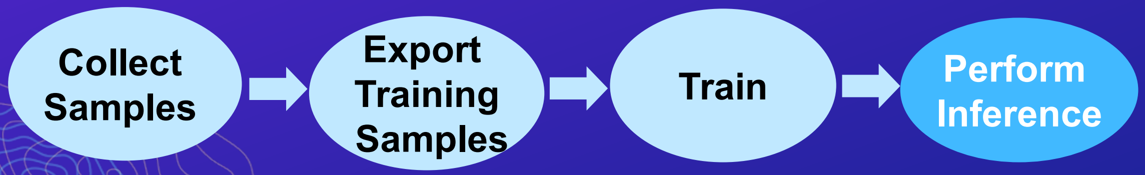
Use Deep Learning Models

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



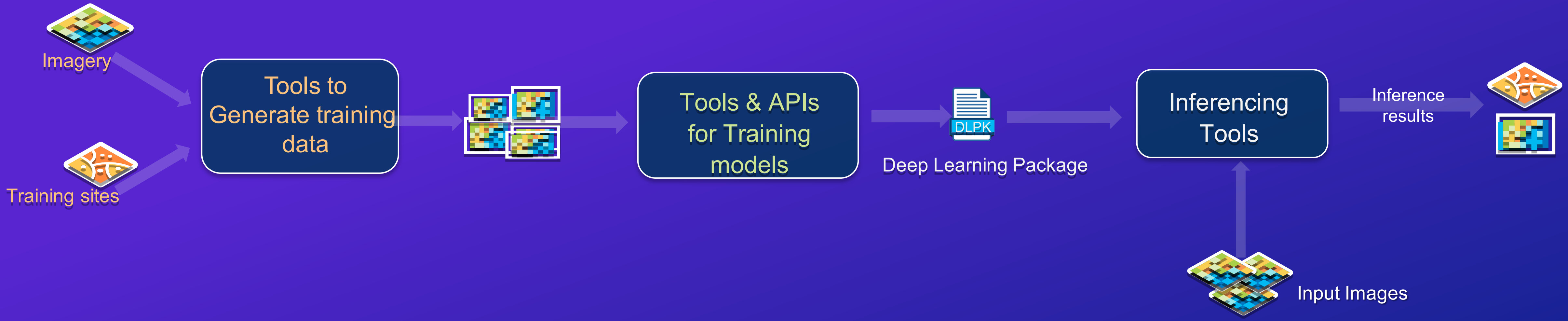
Class Label Field: ClassLabel

```
detect_objects(input_raster=input_layer,
               model=detect_objects_model,
               output_name="Well_Pad_Detection_Sentinel",
               context=context,
               gis=gis)
```



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

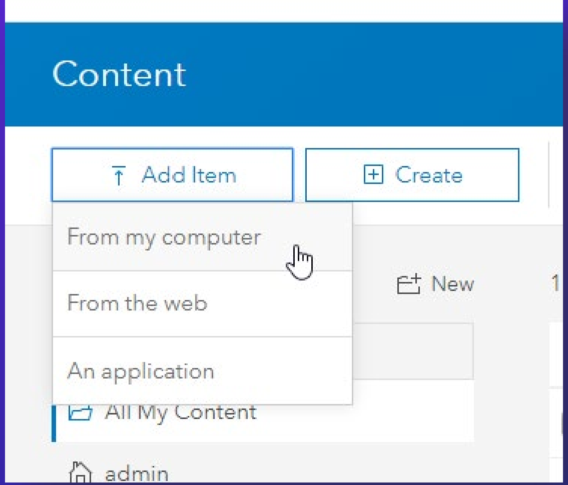
- ArcGIS Pro
- Notebooks

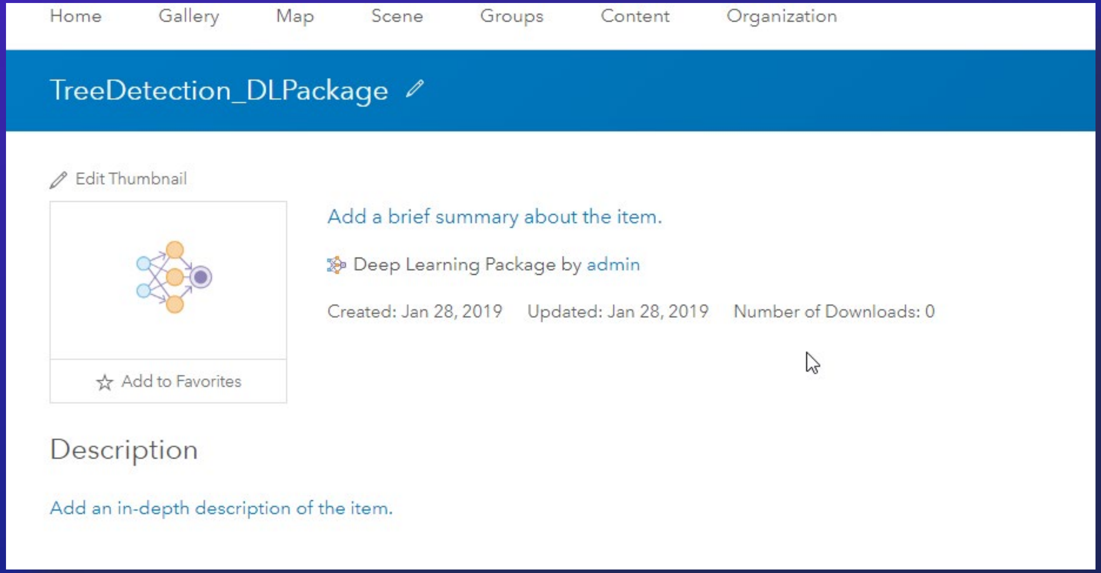
Inferencing

- 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





TreeDetection_DLPackage

Add a brief summary about the item.

Deep Learning Package by admin

Created: Jan 28, 2019 Updated: Jan 28, 2019 Number of Downloads: 0

Description

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		✓	✓	✓

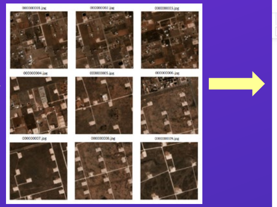
Deep Learning Workflow in ArcGIS

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

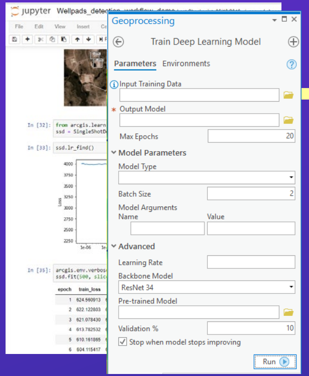
Image Management

Labelling

Data Prep



Train Model



```
In [32]: from arcgis.learn
         ssd = SingleShotD
In [33]: ssd.lr_find()
In [35]: arcgis.env.verbos
         ssd.fit(500, slic
```

epoch	train_loss
1	624.560913
2	622.122803
3	621.078430
4	613.782532
5	610.161865
6	604.115417

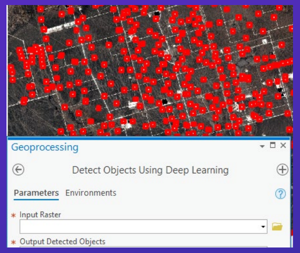
Train Deep Learning Model

Parameters | Environments

- Input Training Data:
- Output Model:
- Max Epochs: 20
- Model Parameters
  - Model Type:
  - Batch Size: 2
  - Model Arguments: Name / Value
- Advanced
  - Learning Rate:
  - Backbone Model: ResNet 34
  - Pre-trained Model:
  - Validation %: 10
  - ☑ Stop when model stops improving

Run

Inferencing

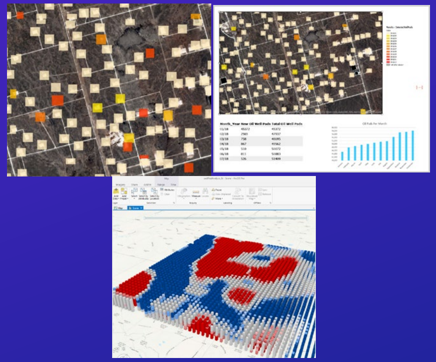


Detect Objects Using Deep Learning

Parameters | Environments

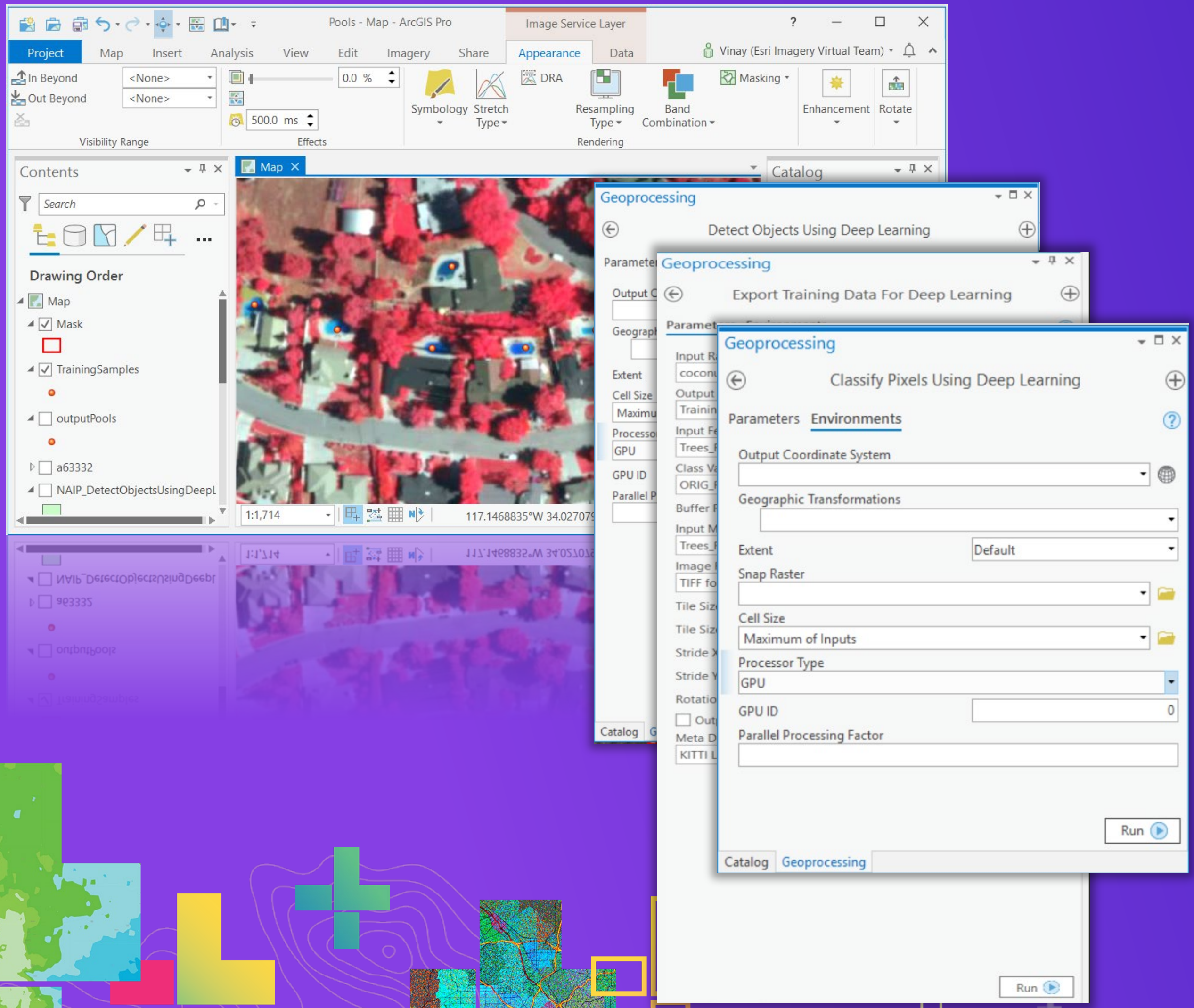
- Input Raster:
- Output Detected Objects

Analysis



Field Mobility, Monitoring

ArcGIS being used for each step of the deep learning workflow



Classify Pixels Using Deep Learning

Parameters Environments

Output Coordinate System

Geographic Transformations

Extent: Default

Snap Raster

Cell Size: Maximum of Inputs

Processor Type: GPU

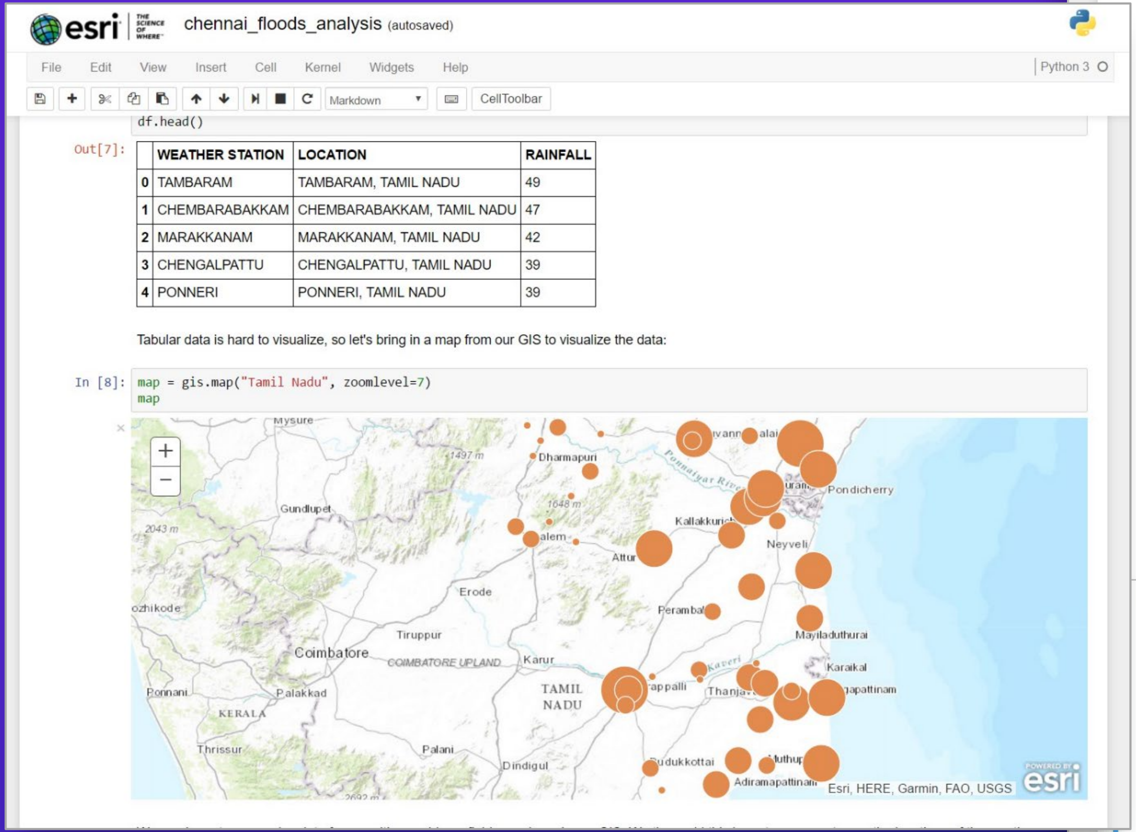
GPU ID: 0

Parallel Processing Factor

Demo

Using Deep Learning Tools in ArcGIS Pro – Well pad detection

ArcGIS + Notebooks = ♥



chennai_floods_analysis (autosaved)

```
df.head()
```

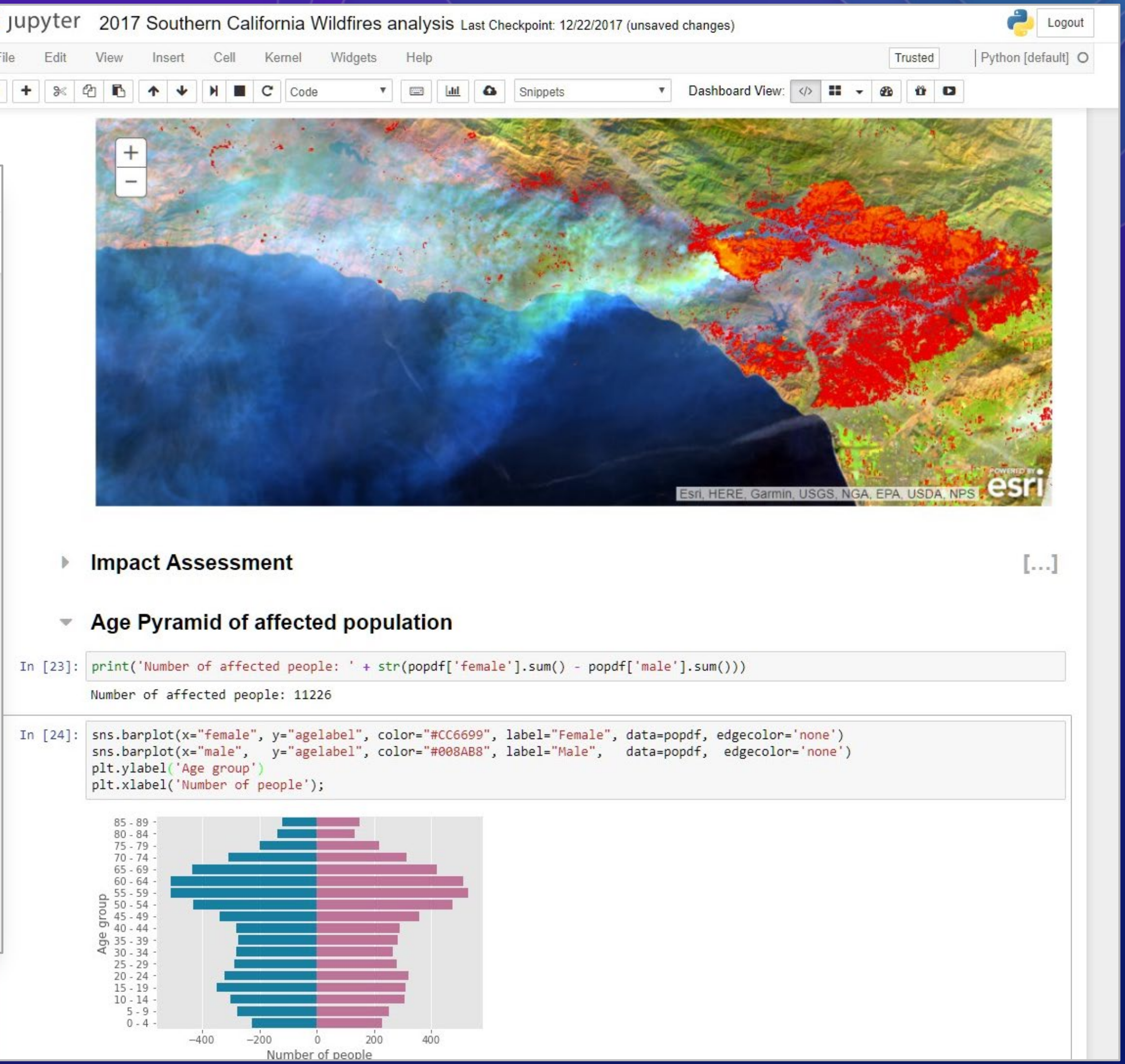
Out[7]:

	WEATHER STATION	LOCATION	RAINFALL
0	TAMBARAM	TAMBARAM, TAMIL NADU	49
1	CHEMBARABAKKAM	CHEMBARABAKKAM, TAMIL NADU	47
2	MARAKKANAM	MARAKKANAM, TAMIL NADU	42
3	CHENGALPATTU	CHENGALPATTU, TAMIL NADU	39
4	PONNERI	PONNERI, TAMIL NADU	39

Tabular data is hard to visualize, so let's bring in a map from our GIS to visualize the data:

In [8]:

```
map = gis.map("Tamil Nadu", zoomlevel=7)
map
```



2017 Southern California Wildfires analysis Last Checkpoint: 12/22/2017 (unsaved changes)

Impact Assessment

Age Pyramid of affected population

In [23]:

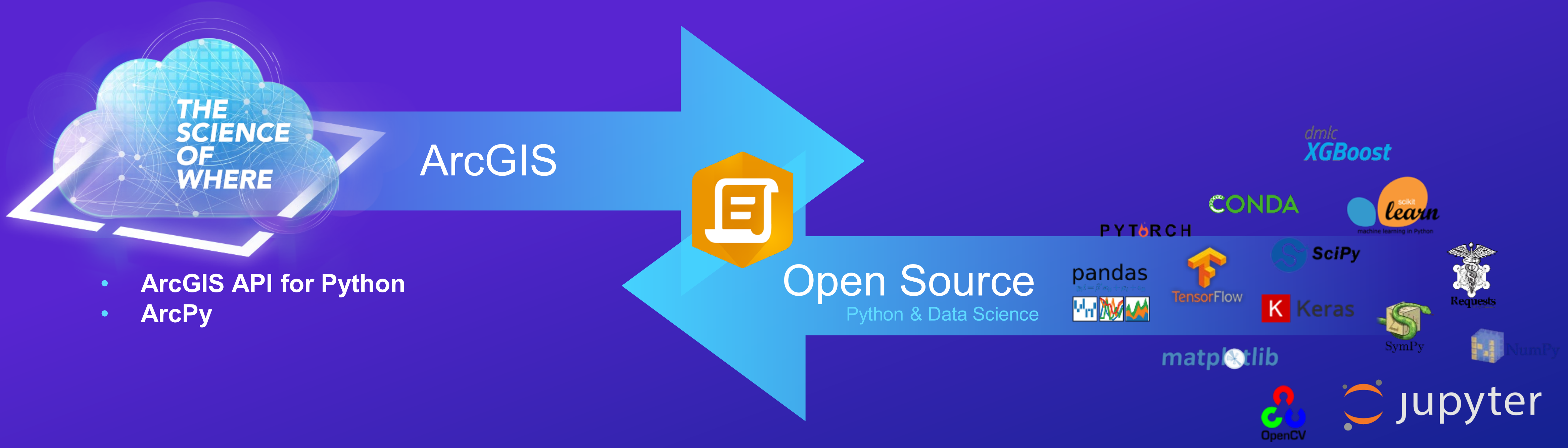
```
print('Number of affected people: ' + str(popdf['female'].sum() - popdf['male'].sum()))
```

```
Number of affected people: 11226
```

In [24]:

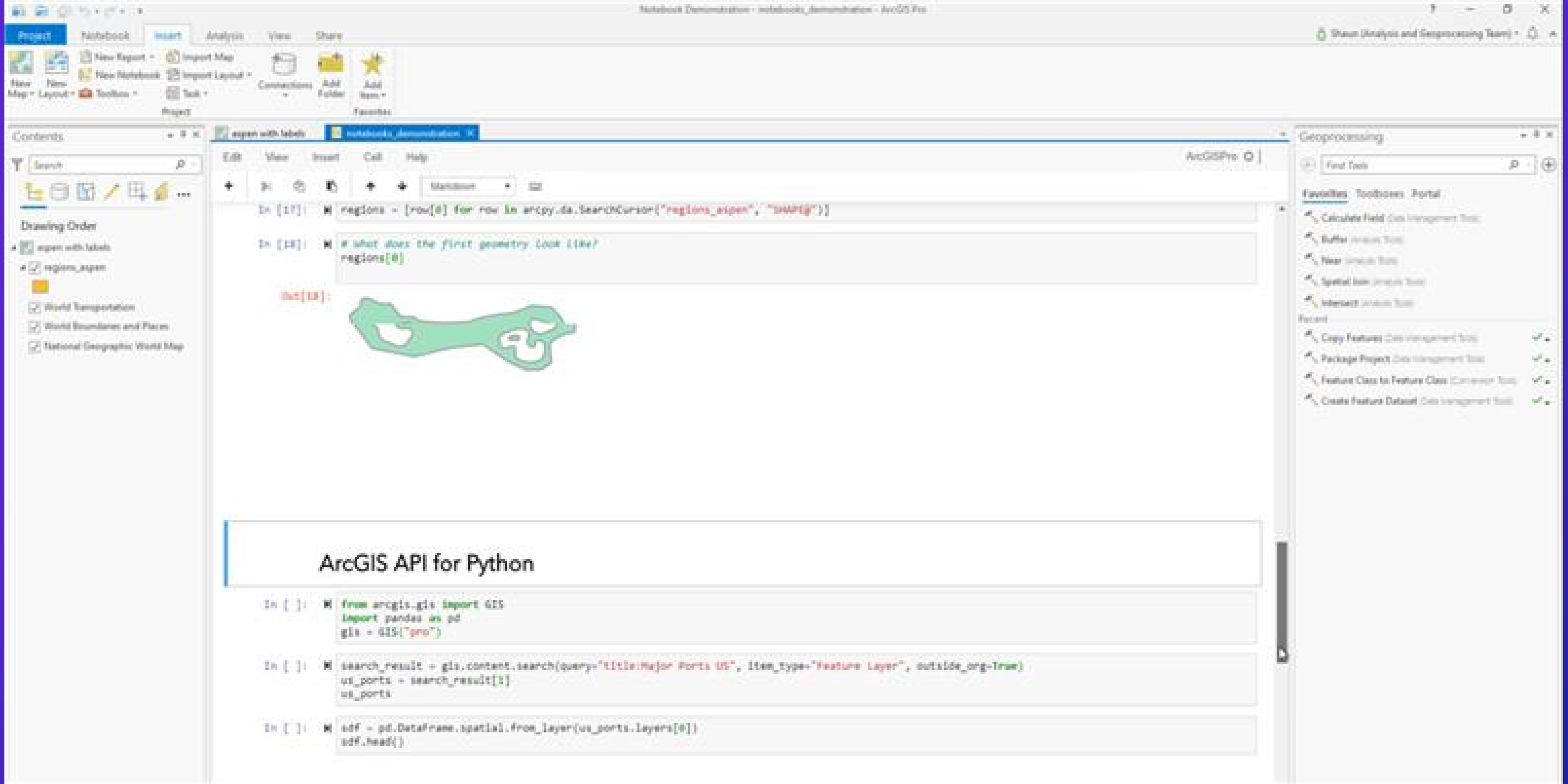
```
sns.barplot(x="female", y="agelabel", color="#CC6699", label="Female", data=popdf, edgecolor='none')
sns.barplot(x="male",   y="agelabel", color="#008AB8", label="Male",   data=popdf,  edgecolor='none')
plt.ylabel('Age group')
plt.xlabel('Number of people');
```

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



- ArcGIS API for Python
- ArcPy

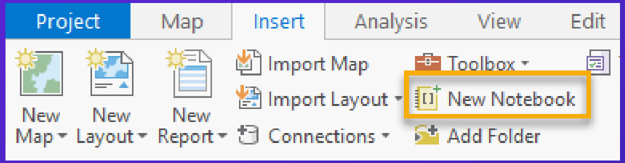
ArcGIS Pro 2.5 seamlessly integrates Python Notebooks

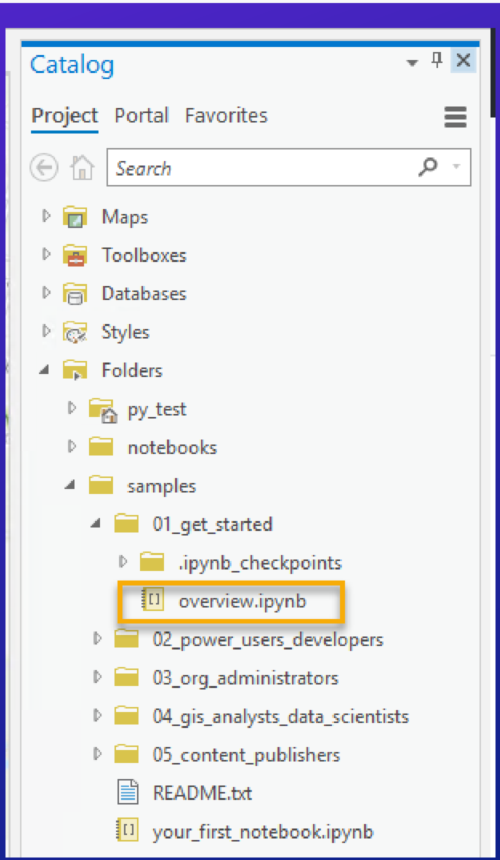


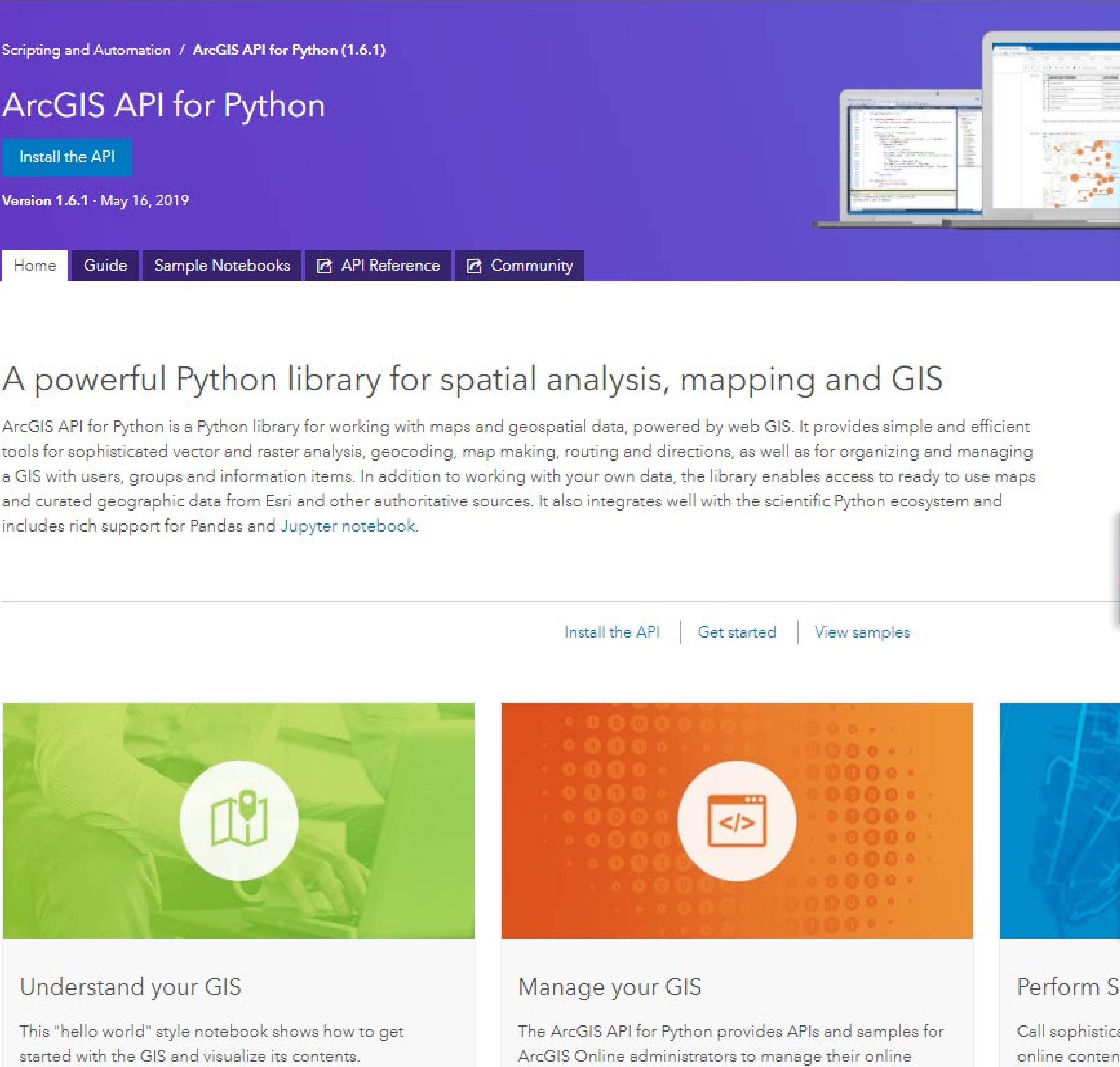
```
In [17]:  regions = [row[0] for row in arcpy.da.SearchCursor("regions_aspen", "SHAPE@")]

In [18]:  # what does the first geometry look like?
          regions[0]
```

ArcGIS API for Python

```
In [ ]:  from arcgis.gis import GIS
         import pandas as pd
         gis = GIS("pro")

In [ ]:  search_result = gis.content.search(query="title:Major Ports US", item_type="Feature Layer", outside_org=True)
         us_ports = search_result[1]
         us_ports

In [ ]:  sdf = pd.DataFrame.spatial.from_layer(us_ports.layers[0])
         sdf.head()
```







Scripting and Automation / ArcGIS API for Python (1.6.1)

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

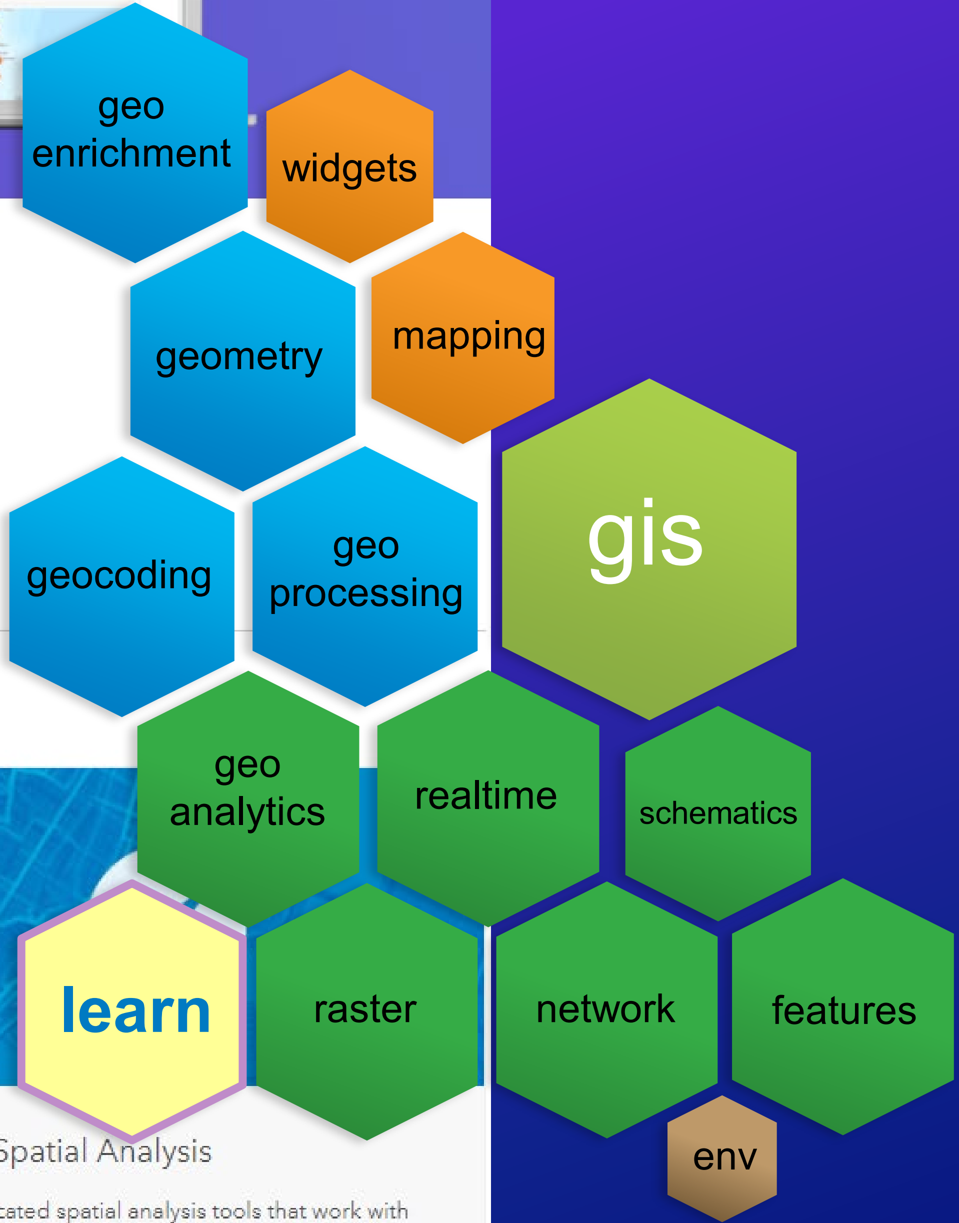
This "hello world" style notebook shows how to get started with the GIS and visualize its contents.

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.



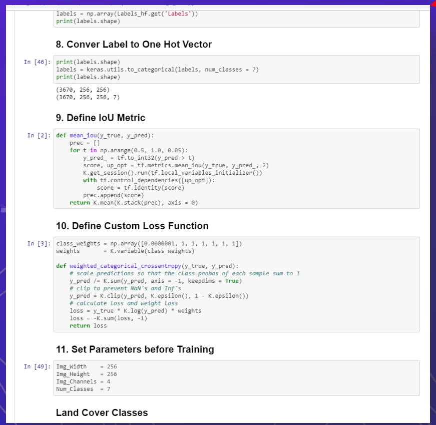
ArcGIS API for Python

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.**

Train Models

Before



```
labels = np.array(Labels_hf.get('Labels'))
print(labels.shape)
```

8. Conver Label to One Hot Vector

```
In [46]: print(labels.shape)
         labels = keras.utils.to_categorical(labels, num_classes = 7)
         print(labels.shape)

(3670, 256, 256)
(3670, 256, 256, 7)
```

9. Define IoU Metric

```
In [2]: def mean_iou(y_true, y_pred):
            prec = []
            for t in np.arange(0.5, 1.0, 0.05):
                y_pred_ = tf.to_int32(y_pred > t)
                score, up_opt = tf.metrics.mean_iou(y_true, y_pred_, 2)
                K.get_session().run(tf.local_variables_initializer())
                with tf.control_dependencies([up_opt]):
                    score = tf.identity(score)
                prec.append(score)
            return K.mean(K.stack(prec), axis = 0)
```

10. Define Custom Loss Function

```
In [3]: class_weights = np.array([0.0000001, 1, 1, 1, 1, 1, 1])
        weights       = K.variable(class_weights)

        def weighted_categorical_crossentropy(y_true, y_pred):
            # scale predictions so that the class probas of each sample sum to 1
            y_pred /= K.sum(y_pred, axis = -1, keepdims = True)
            # clip to prevent NaN's and Inf's
            y_pred = K.clip(y_pred, K.epsilon(), 1 - K.epsilon())
            # calculate loss and weight loss
            loss = y_true * K.log(y_pred) * weights
            loss = -K.sum(loss, -1)
            return loss
```

11. Set Parameters before Training

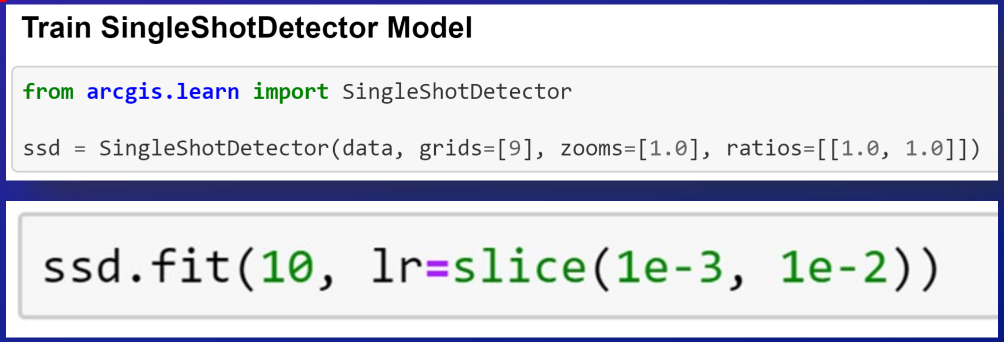
```
In [49]: Img_Width    = 256
         Img_Height   = 256
         Img_Channels = 4
         Num_Classes  = 7
```

Land Cover Classes

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

- 3-5 lines
- No Installation (ArcGIS Pro & Notebooks)
- EASY

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))
```

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*…

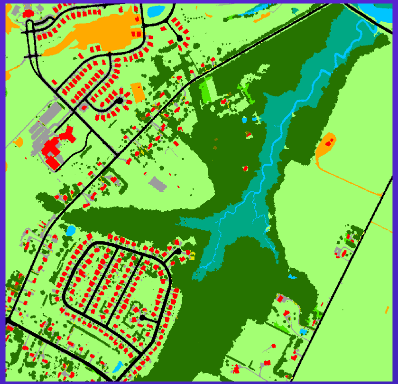
Things you can do today with arcgis.learn

Object Detection, Pixel Classification, Feature Classification, Instance Segmentation

Damaged Structures

Building Footprints

Land Cover



Catfish

Brick Kilns

Roads

Oil Pads

Palm trees

Refugee Camps

Sinkholes

Swimming Pools

Training models using `arcgis.learn`

Types of Deep Learning Models & their applications to GIS

Image Classification

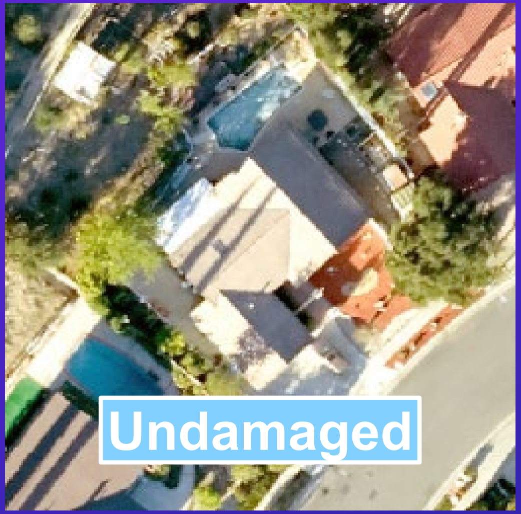
Assign a label to a given image

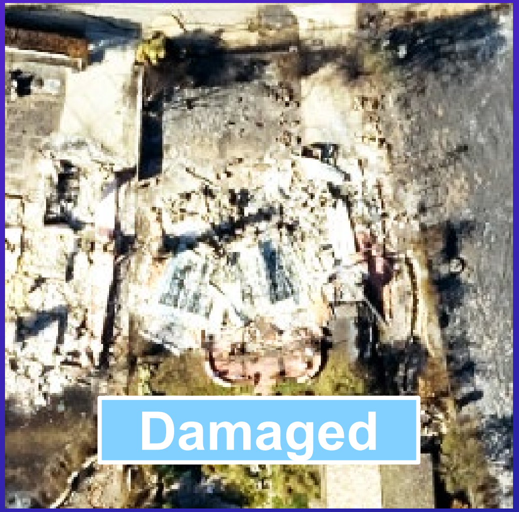
Models (from torchvision):

- Inception
- ResNet
- VGG…

Object Classification

Assign a label to a given feature





Applications:

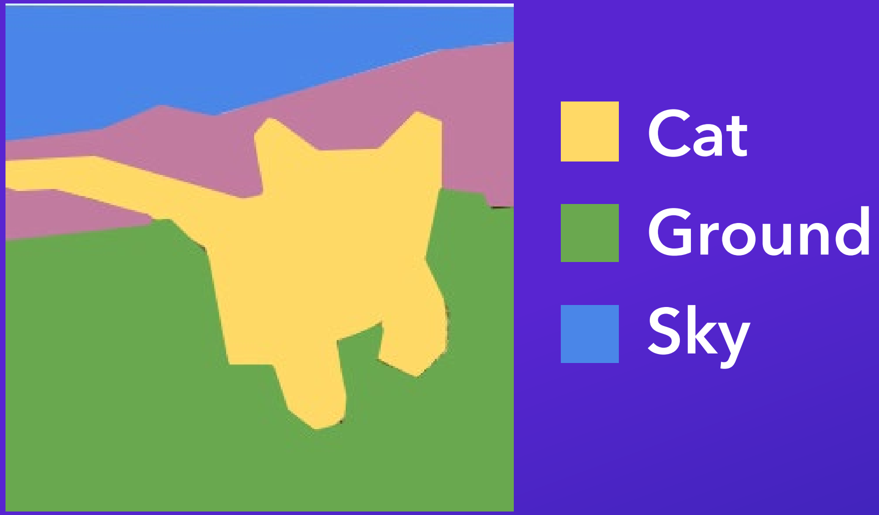
- Damaged building classification
- Clean or ‘green’ pools…

Demo

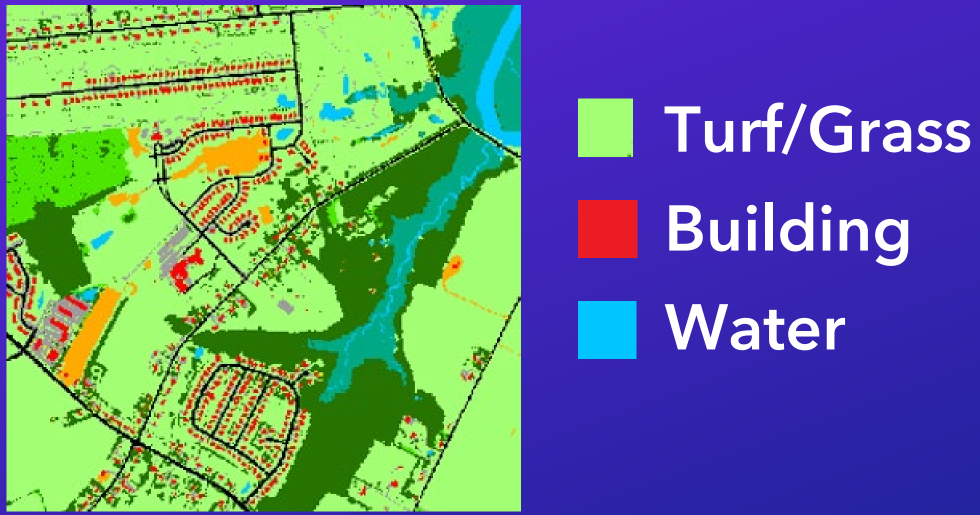
Building Damage classification

Semantic Segmentation

Assign a label to each pixel



Pixel Classification



Models:

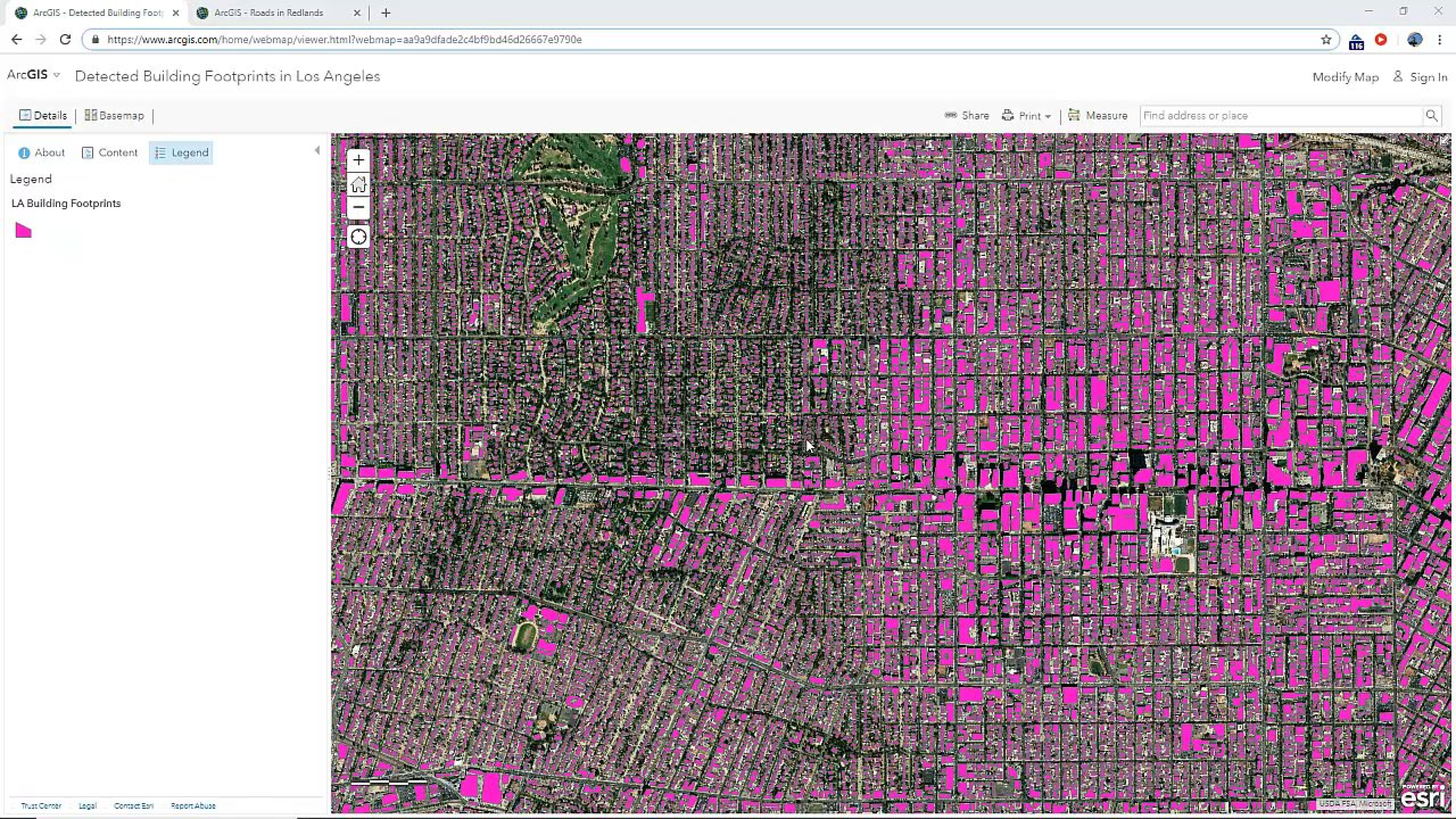
- UNetClassifier
- PSPNetClassifier

Applications:

- Land Cover Classification
- Pervious/Impervious mapping…

Demo

Building Footprint Extraction



Detected Building Footprints in Los Angeles

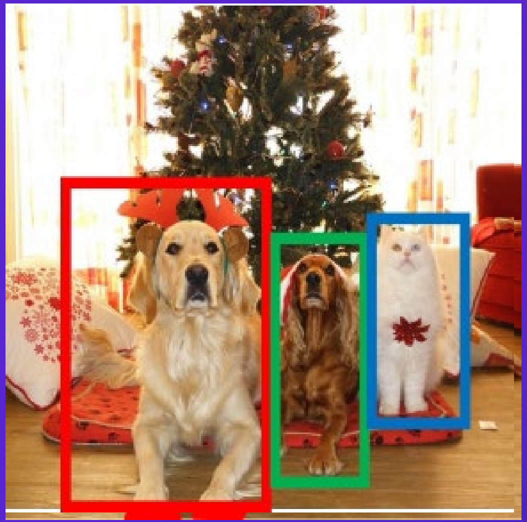
Legend

LA Building Footprints

■

Object Detection

Find objects and their location (bounding boxes)

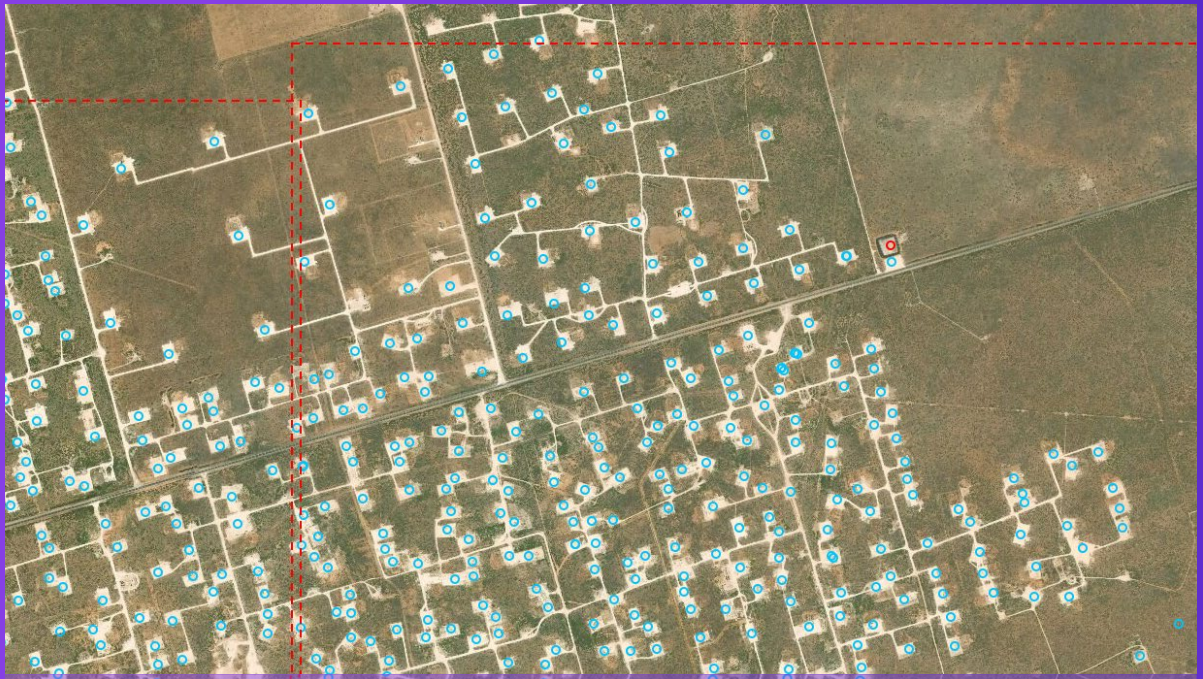


Models:

- SingleShotDetector
- RetinaNet

Applications:

- Detect trees, cars, airplanes, …

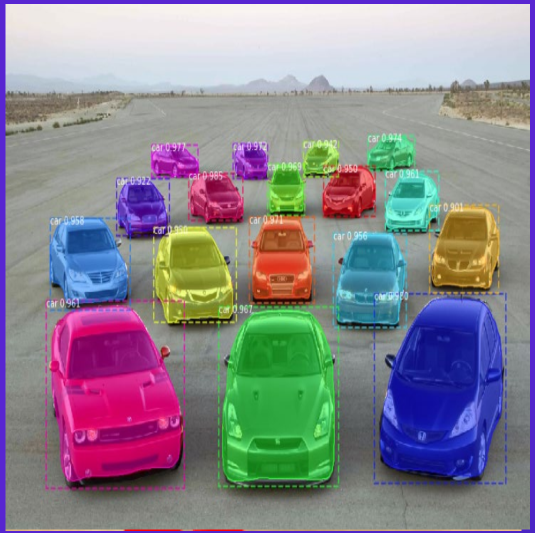


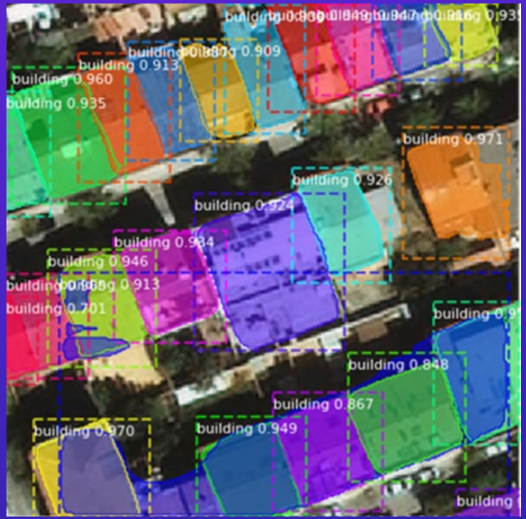
Demo

Detecting Well Pads

Instance Segmentation

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





Models:

- MaskRCNN

Applications:

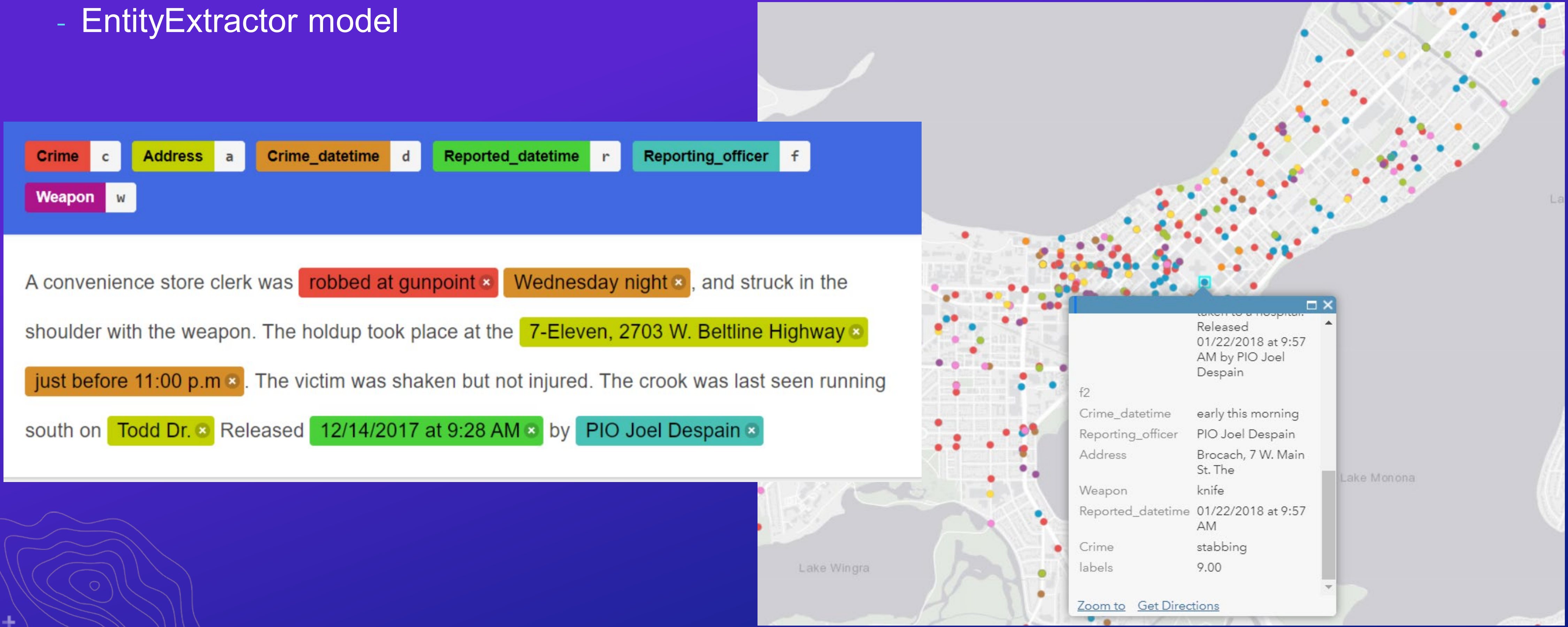
- Building footprint extraction

Demo

Contiguous Building Footprints

Text / NLP

- arcgis.learn – model for extracting location and other entities
  - EntityExtractor model



A convenience store clerk was robbed at gunpoint Wednesday night, and struck in the shoulder with the weapon. The holdup took place at the 7-Eleven, 2703 W. Beltline Highway just before 11:00 p.m. The victim was shaken but not injured. The crook was last seen running south on Todd Dr. Released 12/14/2017 at 9:28 AM by PIO Joel Despain

Released 01/22/2018 at 9:57 AM by PIO Joel Despain

f2

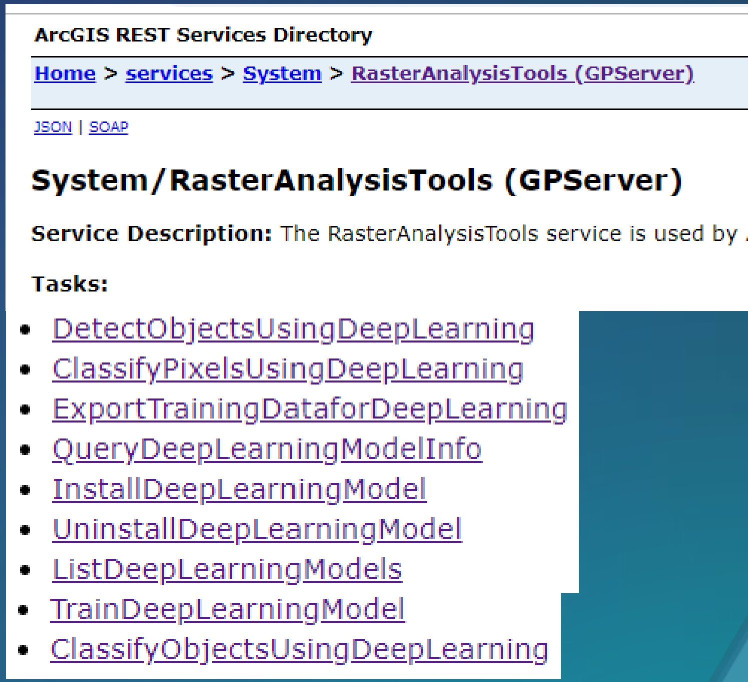
Crime_datetime	early this morning
Reporting_officer	PIO Joel Despain
Address	Brocach, 7 W. Main St. The
Weapon	knife
Reported_datetime	01/22/2018 at 9:57 AM
Crime	stabbing
labels	9.00

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	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*
Server Only Tools	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

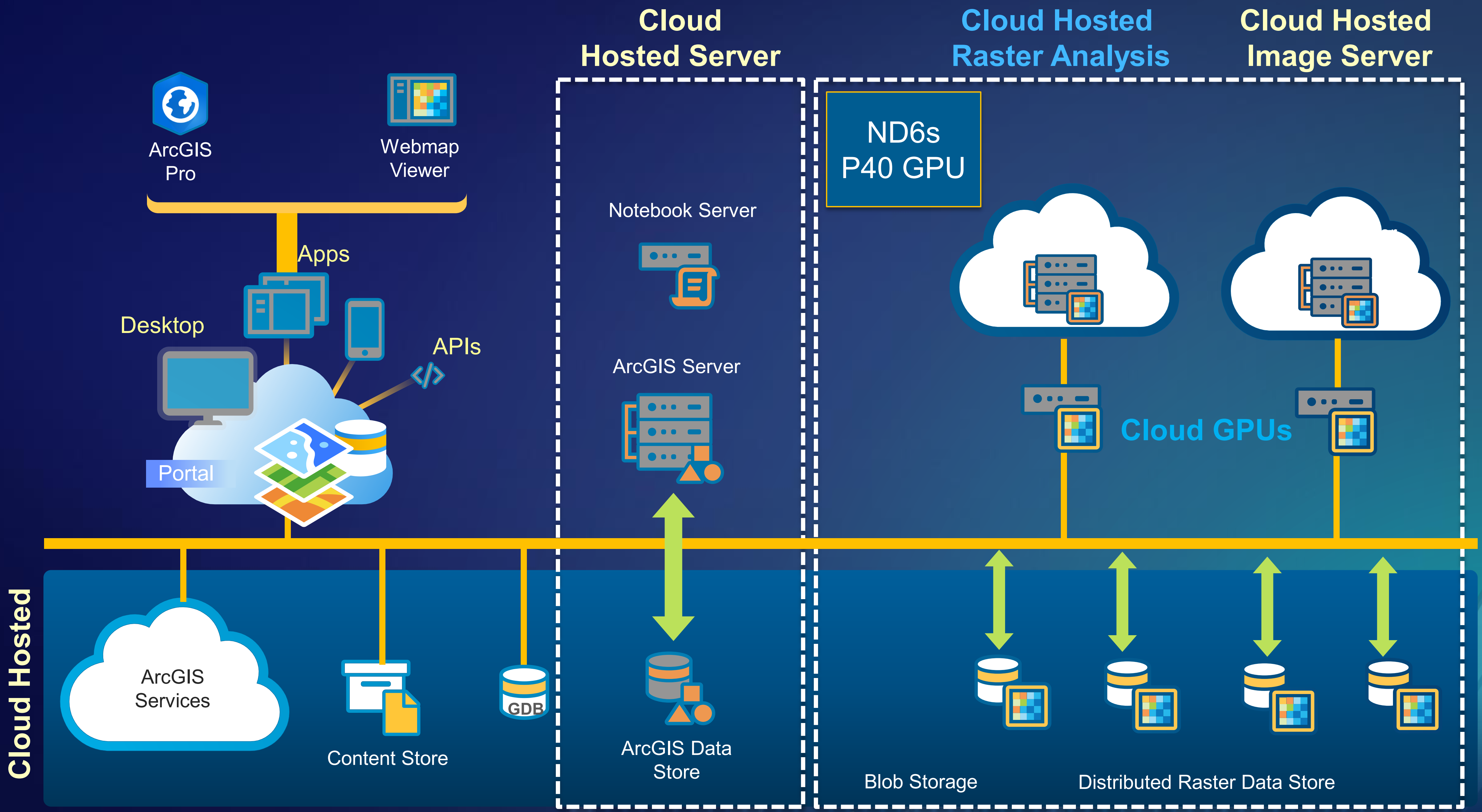
Tasks:

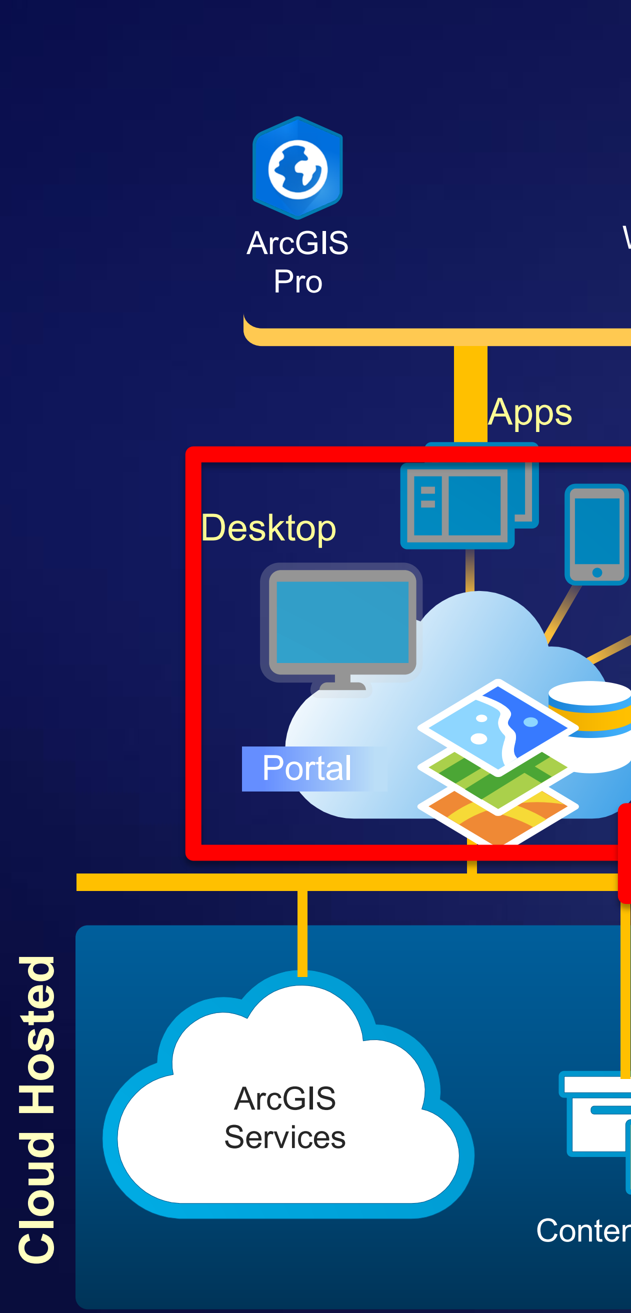
- DetectObjectsUsingDeepLearning
- ClassifyPixelsUsingDeepLearning
- ExportTrainingDataforDeepLearning
- QueryDeepLearningModelInfo
- InstallDeepLearningModel
- UninstallDeepLearningModel
- ListDeepLearningModels
- TrainDeepLearningModel
- ClassifyObjectsUsingDeepLearning

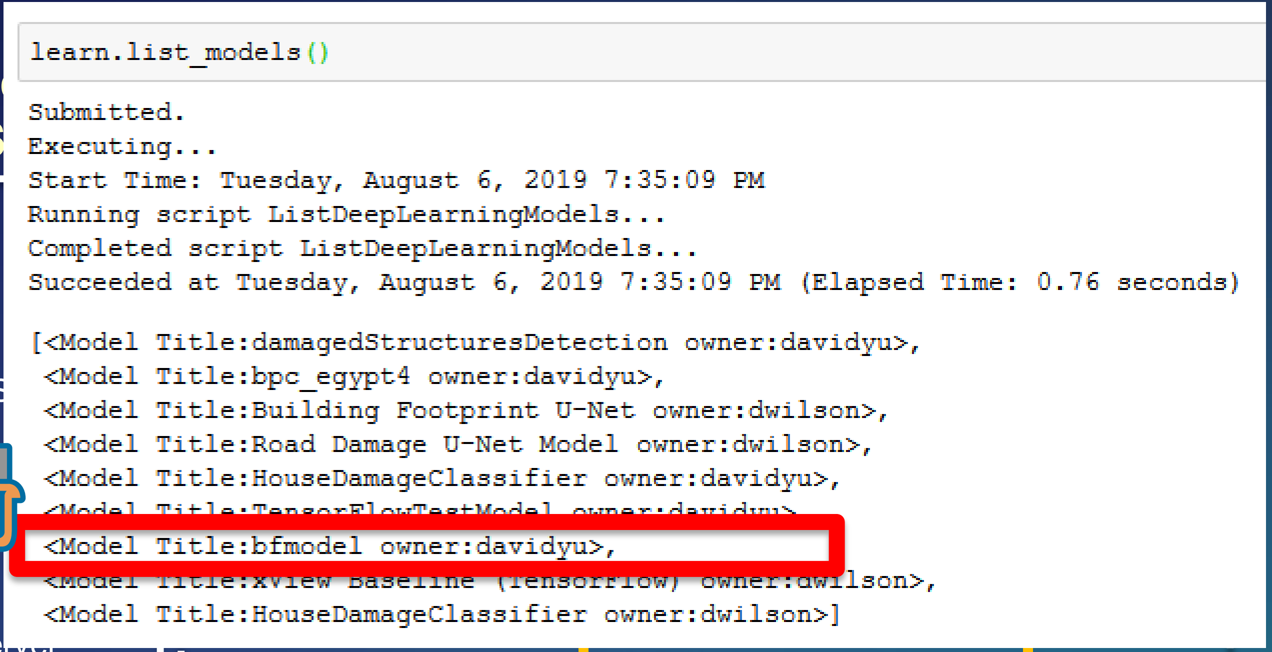
ArcGIS Enterprise Deep Learning – System Architecture

Scenario – Detecting Building Footprints at Scale

- Aim: Determine benchmark on a fixed dataset with a fixed inference method
  - Denver World Imagery
    - 4,727 blobs
    - 13,375,880,336 bytes
    - 1,669,001,963 m^2 (0.6m)
    - 2048x2048 cells per blob (TIF)
    - Custom Mask R-CNN for detecting building footprints
  - Florida aerial imagery from NOAA
  - xView baseline model for detecting building footprints
- Images stored in Azure Blob Storage
- Speedup achievable from scaling up # of Raster Analysis Servers
  - Observe relationship

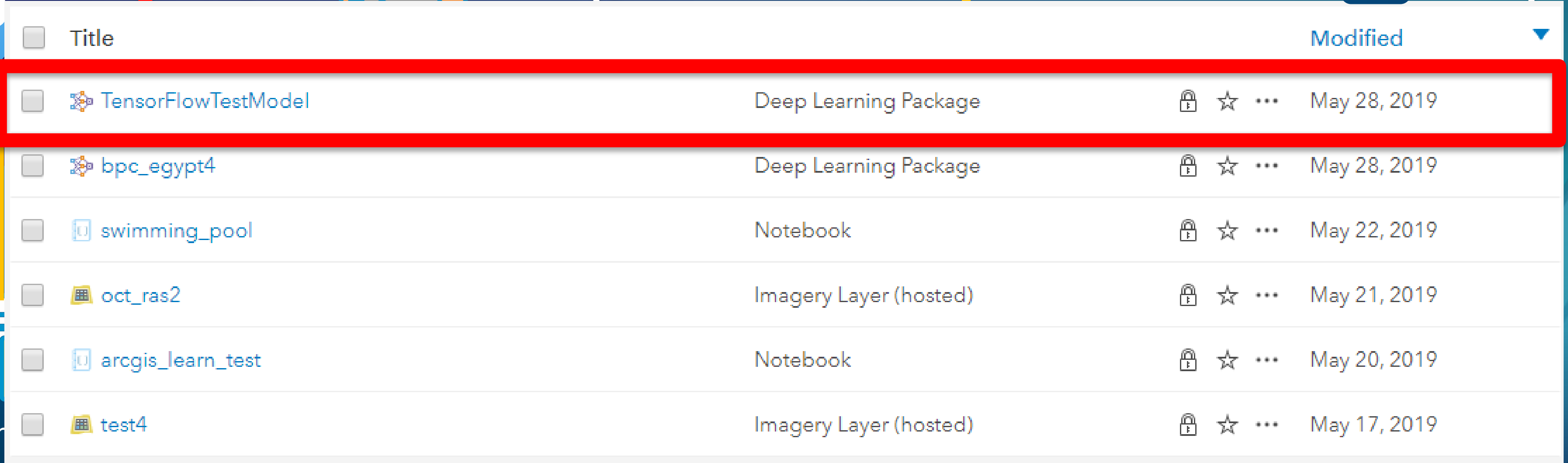

Cloud
Hosted Server
Cloud Hosted
Raster Analysis
Cloud Hosted
Image Server
ArcGIS
Pro
Webmap
Viewer
ND6s
P40 GPU
Notebook Server
Apps
Desktop
APIs
ArcGIS Server
Portal
Cloud GPUs
Cloud Hosted
ArcGIS
Services
Content Store
GDB
ArcGIS Data
Store
Blob Storage
Distributed Raster Data Store



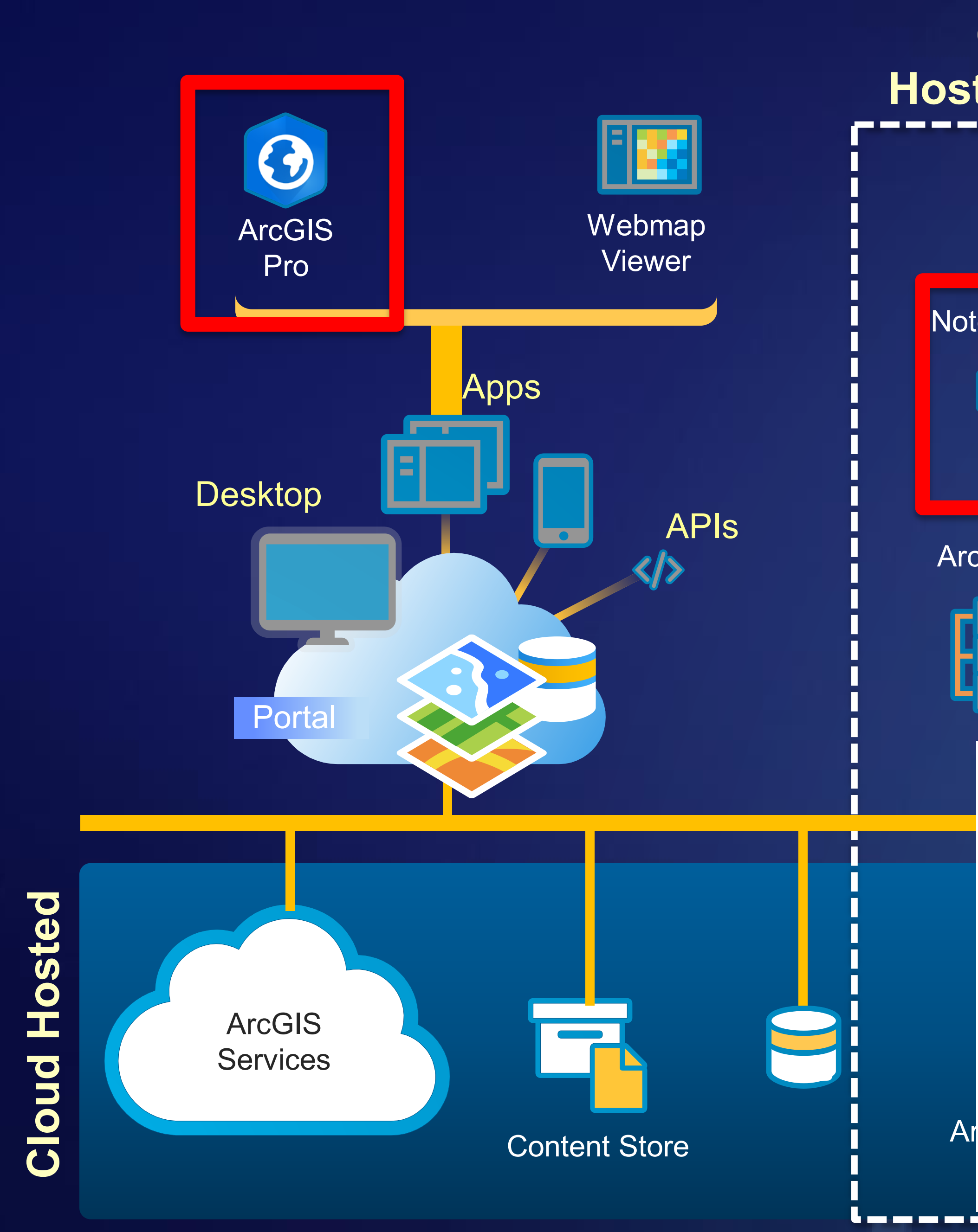


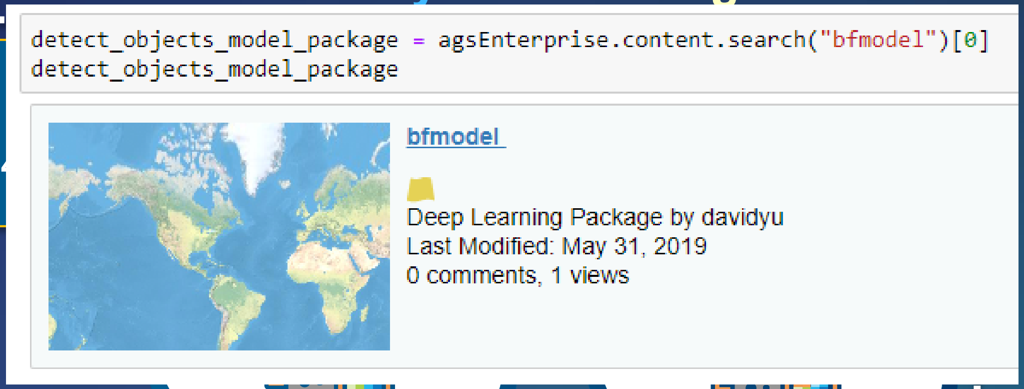
```
learn.list_models()
```

```
Submitted.
Executing...
Start Time: Tuesday, August 6, 2019 7:35:09 PM
Running script ListDeepLearningModels...
Completed script ListDeepLearningModels...
Succeeded at Tuesday, August 6, 2019 7:35:09 PM (Elapsed Time: 0.76 seconds)

[<Model Title:damagedStructuresDetection owner:davidyu>,
 <Model Title:bpc_egypt4 owner:davidyu>,
 <Model Title:Building Footprint U-Net owner:dwilson>,
 <Model Title:Road Damage U-Net Model owner:dwilson>,
 <Model Title:HouseDamageClassifier owner:davidyu>,
 <Model Title:TensorFlowTestModel owner:davidyu>,
 <Model Title:bfmodel owner:davidyu>,
 <Model Title:xView Baseline (TensorFlow) owner:dwilson>,
 <Model Title:HouseDamageClassifier owner:dwilson>]
```



□	Title			Modified ▼
□	TensorFlowTestModel	Deep Learning Package	🔒 ☆ ···	May 28, 2019
□	bpc_egypt4	Deep Learning Package	🔒 ☆ ···	May 28, 2019
□	swimming_pool	Notebook	🔒 ☆ ···	May 22, 2019
□	oct_ras2	Imagery Layer (hosted)	🔒 ☆ ···	May 21, 2019
□	arcgis_learn_test	Notebook	🔒 ☆ ···	May 20, 2019
□	test4	Imagery Layer (hosted)	🔒 ☆ ···	May 17, 2019

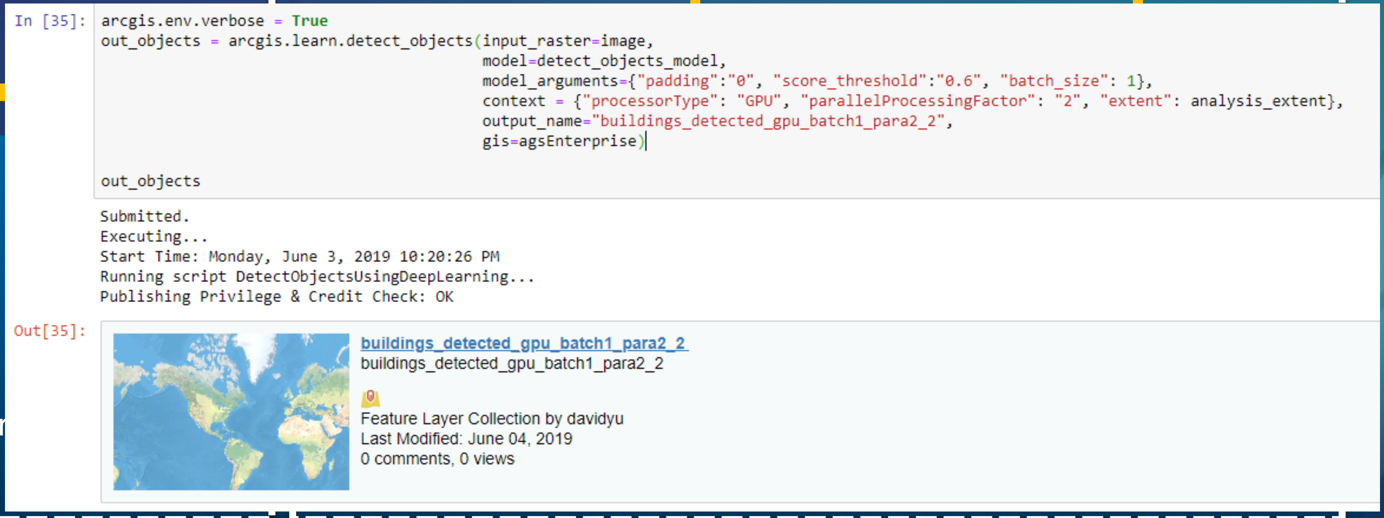




```
detect_objects_model_package = agsEnterprise.content.search("bfmodel")[0]
detect_objects_model_package
```

bfmodel

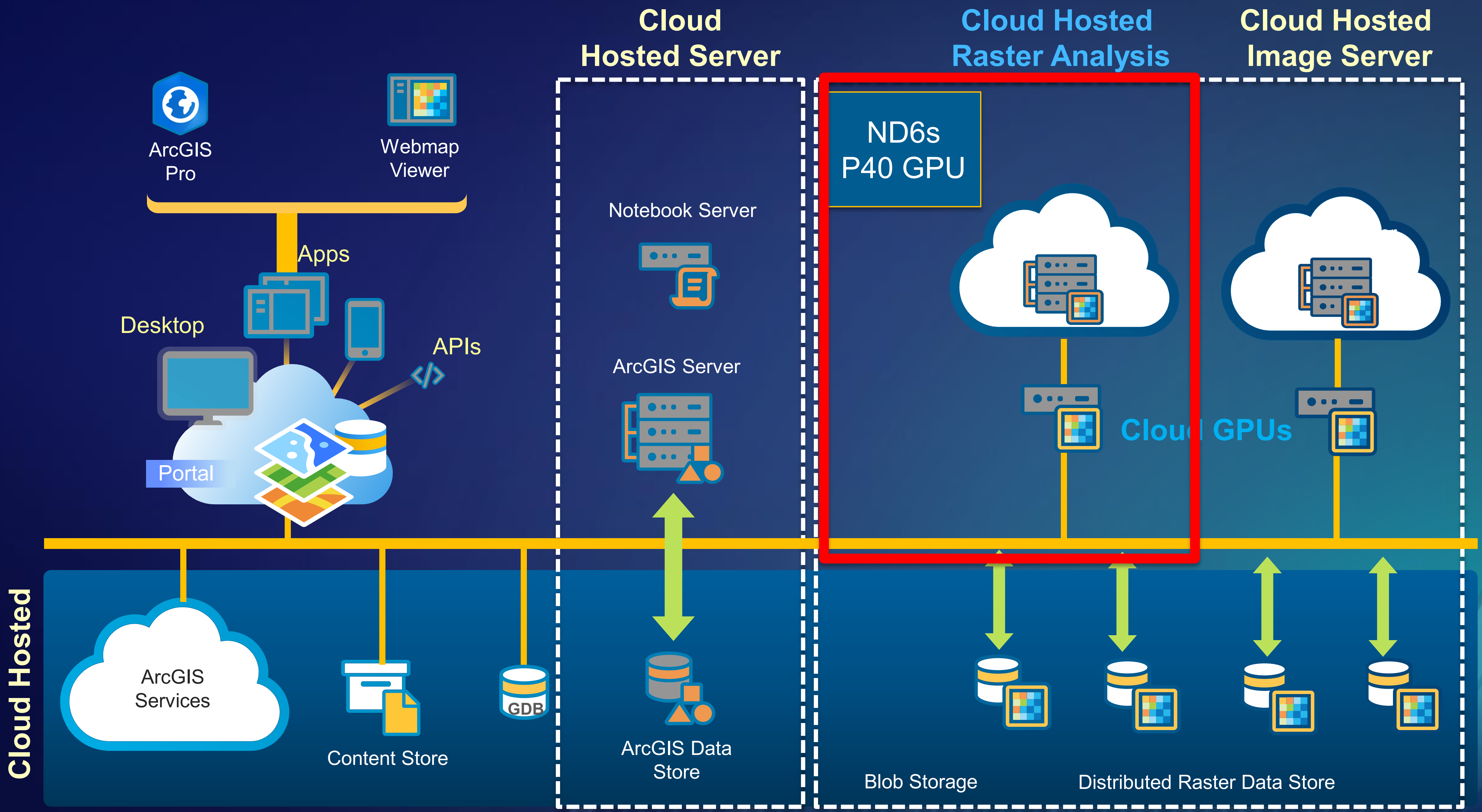
Deep Learning Package by davidyu
Last Modified: May 31, 2019
0 comments, 1 views



```
In [35]: arcgis.env.verbose = True
         out_objects = arcgis.learn.detect_objects(input_raster=image,
                                                   model=detect_objects_model,
                                                   model_arguments={"padding":"0", "score_threshold":"0.6", "batch_size": 1},
                                                   context = {"processorType": "GPU", "parallelProcessingFactor": "2", "extent": analysis_extent},
                                                   output_name="buildings_detected_gpu_batch1_para2_2",
                                                   gis=agsEnterprise)

         out_objects

         Submitted.
         Executing...
         Start Time: Monday, June 3, 2019 10:20:26 PM
         Running script DetectObjectsUsingDeepLearning...
         Publishing Privilege & Credit Check: OK
Out[35]:
```

buildings_detected_gpu_batch1_para2_2
buildings_detected_gpu_batch1_para2_2

Feature Layer Collection by davidyu
Last Modified: June 04, 2019
0 comments, 0 views



Cloud Hosted Server

Cloud Hosted Raster Analysis

Cloud Hosted Image Server

ArcGIS Pro

Webmap Viewer

Apps

Desktop

APIs

Portal

Notebook Server

ArcGIS Server

ND6s P40 GPU

Cloud GPUs

Cloud Hosted

ArcGIS Services

Content Store

GDB

ArcGIS Data Store

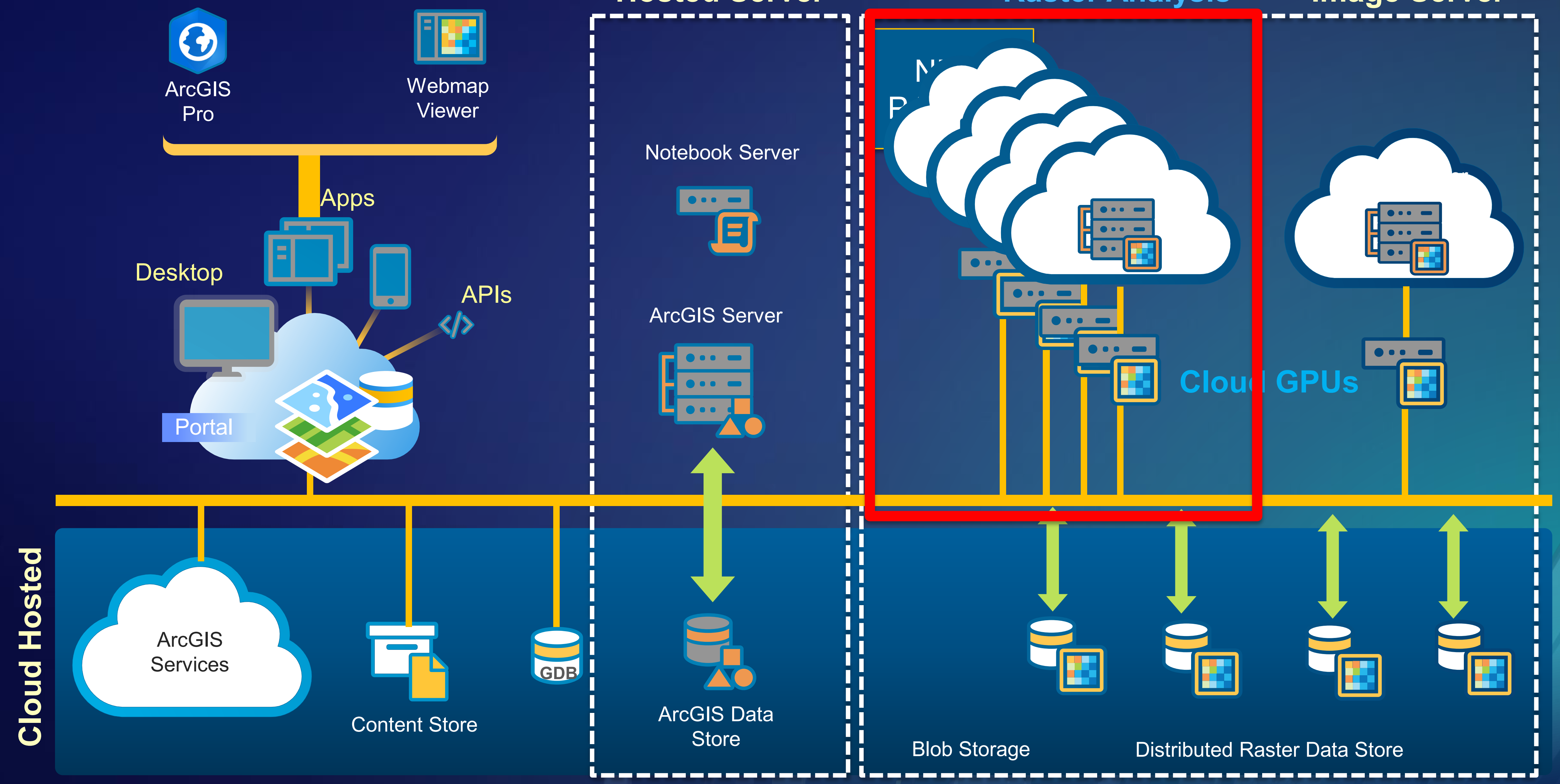
Blob Storage

Distributed Raster Data Store

Cloud Hosted Server

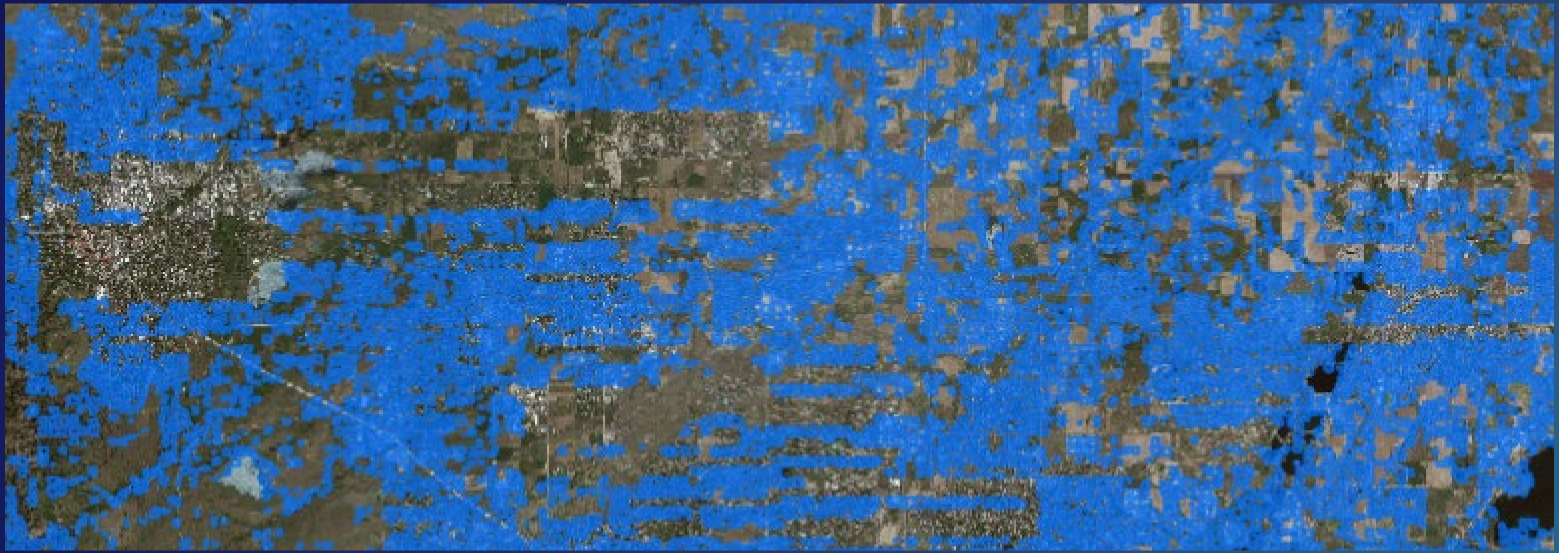
Cloud Hosted Raster Analysis

Cloud Hosted Image Server



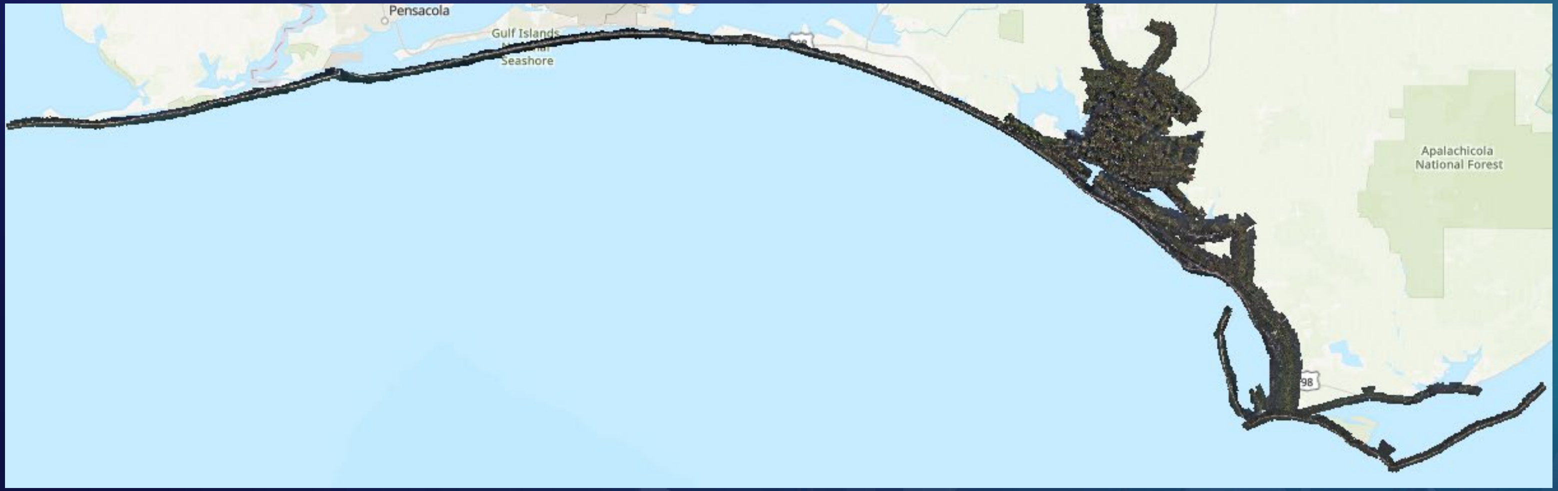
Benchmarks

Denver Imagery



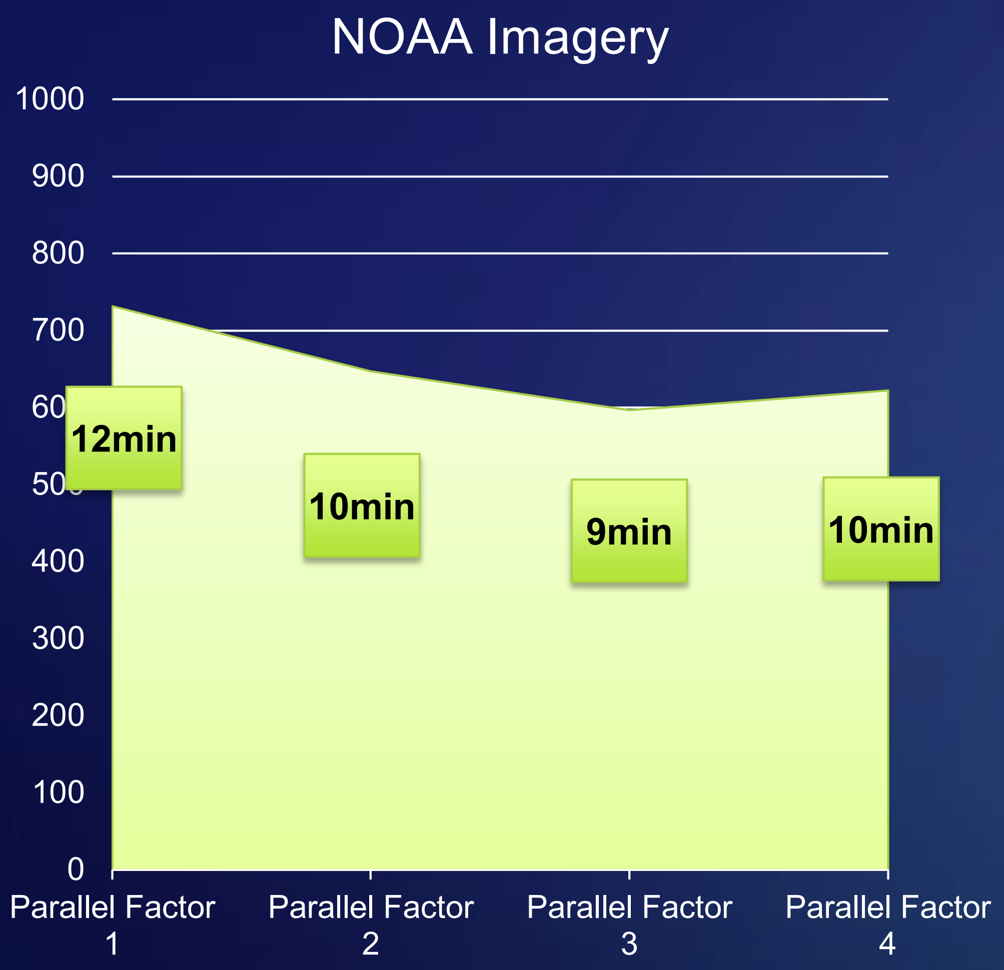
County R
Palomino St
County Road 37
Pinto St
Shetland
Nelson St

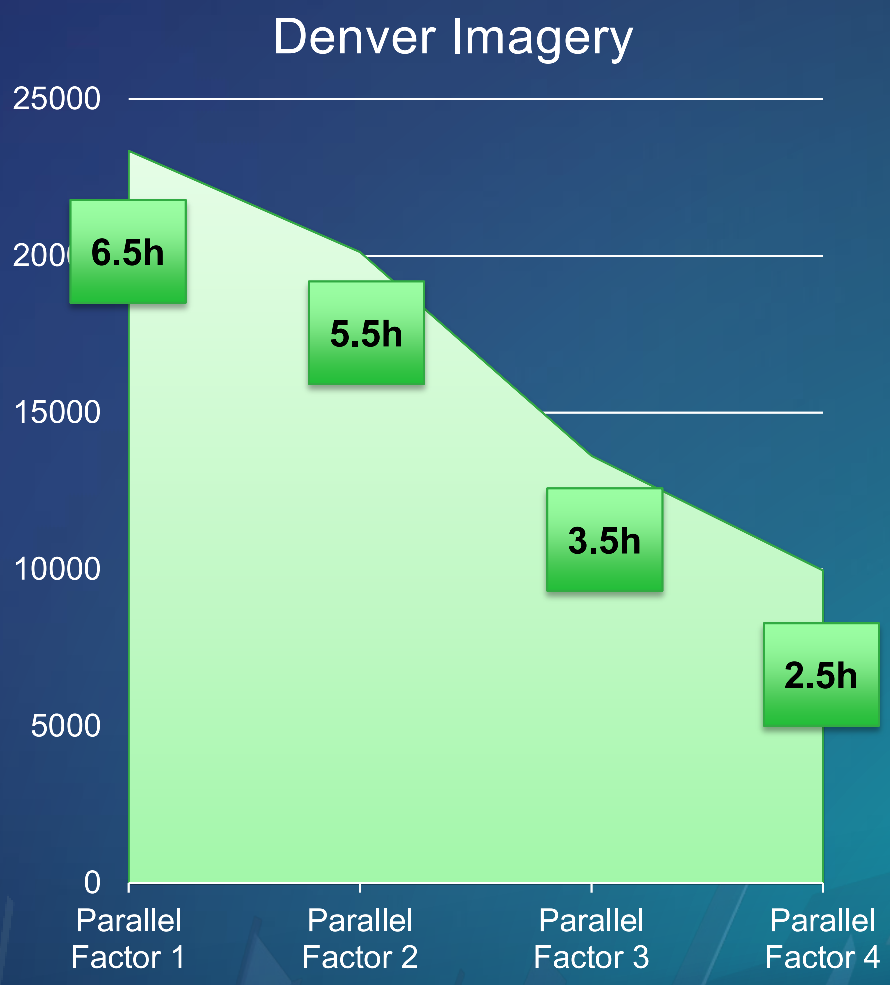
NOAA Imagery



E Baldwin Rd
W Orlando Rd
Syracuse Ave

Benchmarks


NOAA Imagery
1000
900
800
700
600
500
400
300
200
100
0
12min
10min
9min
10min
Parallel Factor 1
Parallel Factor 2
Parallel Factor 3
Parallel Factor 4


Denver Imagery
25000
20000
15000
10000
5000
0
6.5h
5.5h
3.5h
2.5h
Parallel Factor 1
Parallel Factor 2
Parallel Factor 3
Parallel Factor 4

Recap

- New Geoprocessing tools
  - model training in ArcGIS Pro and Enterprise
  - Inferencing in ArcGIS Pro and Enterprise
- arcgis.learn module enhancements
- Sample Notebooks and Learn lessons for deep learning workflows
- New model types (PSPNet, RetinaNet, MaskRCNN…)
- Multi GPU inference using enterprise

Resources

- Geonet Community https://community.esri.com/groups/arcgis-python-api/pages/overview
- ArcGIS API for Python https://developers.arcgis.com/python/
- API Reference https://developers.arcgis.com/python/api-reference/
- GitHub Repo https://github.com/Esri/arcgis-python-api
- Sample Notebooks https://developers.arcgis.com/python/sample-notebooks/
- GeoAI blogs https://medium.com/geoai

esri

THE SCIENCE OF WHERE