Spatial Data Science MOOC

Exercise

Train and use an object detection model

Section 5 Exercise 2

November 17, 2020

esri
THE SCIENCE OF WHERE

Train and use an object detection model

Time to complete
60 minutes

Introduction

ArcGIS provides various ways in which you can complete a deep learning analysis. You can use the geoprocessing tools in ArcGIS Pro or use ArcGIS API for Python (Python API) to complete the analysis using code. The method that you choose depends on the analysis environment that you prefer. In this exercise, you will train the model using geoprocessing tools in ArcGIS Pro.

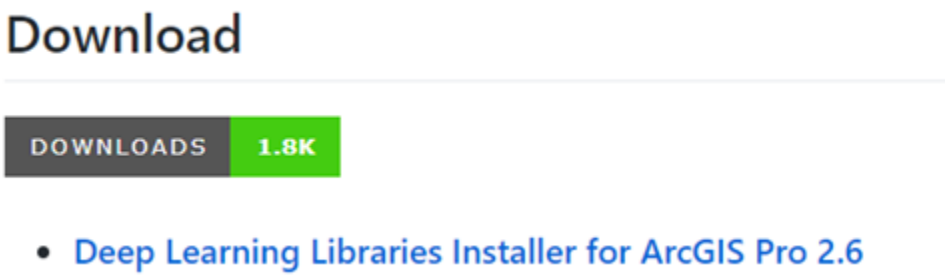
Step 1: Install deep learning libraries

To use the deep learning tools in ArcGIS, you must install specific Python packages. In this step, you will use the Deep Learning Libraries Installer for ArcGIS Pro 2.6 to install the Python packages required to train your deep learning object detection model.

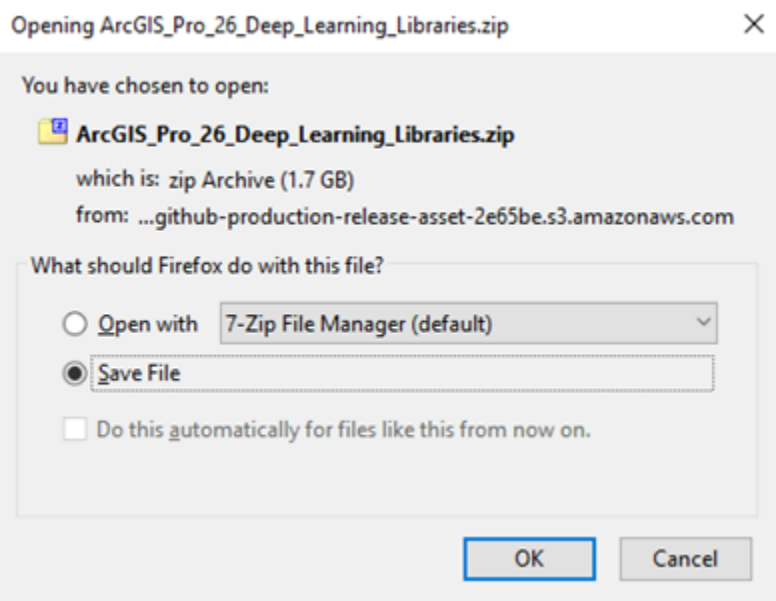
> The Deep Learning Libraries Installer is a 1.7 GB zip file. This is a larger file size than the other download files used in previous sections of this MOOC. It may take several minutes to download the file and complete the installation process.

a. Open a new web browser tab or window.

b. Go to Esri's GitHub deep-learning-frameworks page, scroll down, and then download the Deep Learning Libraries Installer for ArcGIS Pro 2.6.



c. Extract the files to a folder on your local computer, saving them in a location that you will remember.



Note: The window on your screen may look different from the image above.

This zip file contains a Windows Installer Package (.msi) and a Cabinet file (.cab). The Windows Installer Package includes a broad collection of components for performing deep learning and machine learning tasks (a total collection of 95 packages). These packages can be used with the Deep Learning Training tools by using the arcgis.learn module within ArcGIS API for Python and directly imported into your own scripts and tools.

d. Browse to the folder on your local computer where you saved the files.

e. Double-click the ProDeepLearning Windows Installer Package (.msi file).

f. If an Open File - Security Warning window appears, click Run to begin the installation process.

g. In the Deep Learning Libraries For ArcGIS Pro Setup window, follow the steps to complete the installation.

h. When installation is complete, continue to the next step.

Step 2: Confirm the Python environment

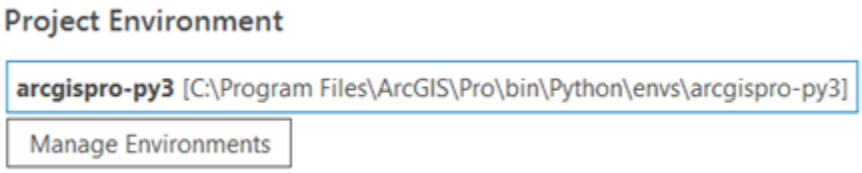
To begin, you will open the ArcGIS Pro project package that you downloaded for the first exercise and confirm that the arcgispro-py3 Python environment with the Deep Learning packages was successfully set up in ArcGIS Pro on your machine.

a. If necessary, start ArcGIS Pro and open the ObjectDetection.aprx project.

b. On the ribbon, click the blue Project tab.

c. In the left pane, click Python.

d. Under Project Environment, verify that arcgispro-py3 is listed at the beginning of the file path.



Multiple installations of Python can coexist on a single workstation, independent of one another. Each of these installations is referred to as a Python environment. By default, ArcGIS Pro has a single Python environment, arcgispro-py3, which includes all Python libraries used by ArcGIS Pro as well as several others, such as scipy and pandas.

The Deep Learning Libraries Installer adds all the included packages to the default arcgispro-py3 environment. Therefore, no custom environments need to be created and you can begin using the Deep Learning Training tools. If you do create custom environments, these packages will be included so that you can use the same tools in your own custom environments, as well.

Note: If you are working in a cloned Python environment, you will need to activate the default arcgispro-py3 environment to complete this exercise. Open the ArcGIS Pro Help: Python Package Manager documentation to learn how to activate the arcgispro-py3 environment.

e. In the top-left corner of ArcGIS Pro, click the Back button to return to your project.

> You must complete the first exercise in this section before proceeding to the next step.

Step 3: Train the model using a geoprocessing tool

Next, you will train the model using the Train Deep Learning Model geoprocessing tool in ArcGIS Pro.

a. In the Geoprocessing pane, search for and open the **Train Deep Learning Model (Image Analyst Tools)** tool.

Note: This tool may take a moment to open.

b. For Input Training Data, browse to and select your **..\ObjectDetection\ImageChips** folder.

c. For Output Model, type **PoolsModel_25_SSD**.

d. For Max Epochs, type **25**.

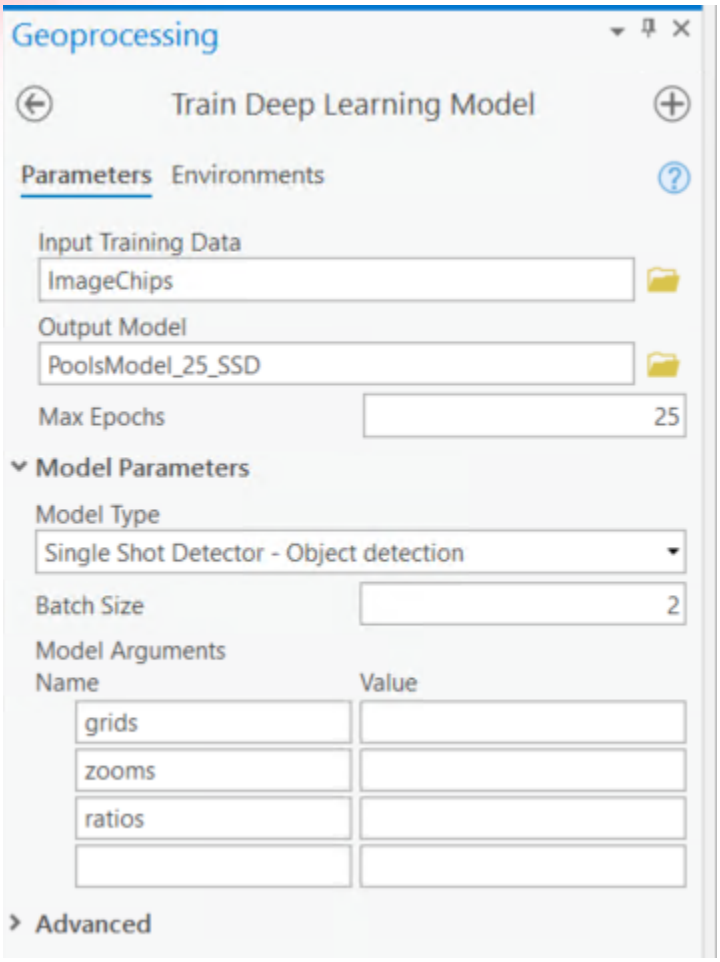
The number of epochs defines the number of times that the image chips will be processed by the neural network. The default number of 20 is a baseline to which you can adjust based on the results of your model.

e. Expand Model Parameters.

f. For Model Type, verify that Single Shot Detector - Object Detection is selected.

The model type will determine the deep learning algorithm and neural network that you will use to train your model. The models available to you depend on the metadata format chosen for the image chips. You chose a metadata format that is associated with object detection, so only the object detection model types are available. For more information about the other model types, see ArcGIS Pro Help: *Train Deep Learning Model (Image Analyst)*.

g. Leave the remaining defaults.



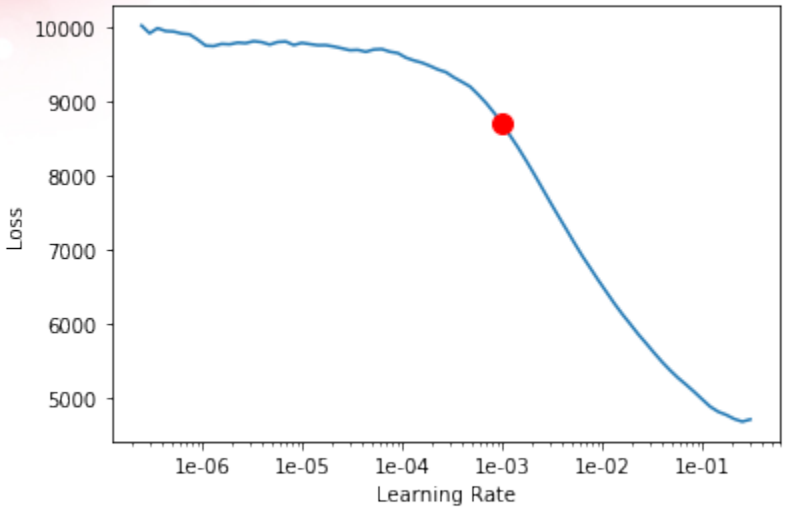
Model arguments refer to specific parameter values used to train the model. The model arguments will vary based on the model type that you choose. For the Single Shot Detector - Object Detection model type, you can specify the grid cell size, zoom level, and aspect ratio. These values define how the model examines the image to detect objects. For more information about the Single-Shot Detector model, see ArcGIS API for Python Help: *How single-shot detector (SSD) works?*

h. To learn more about the individual Model Parameters, point to the Geoprocessing Input Information icons.

i. Expand Advanced.

j. Leave the default settings for all fields.

The following graphic is used to explain one of the advanced parameters.



In this example, the loss (or model error) decreases as the learning rate is increased from 1e-7 to 1e-1. After this point, the loss begins to increase again. The part of the graph between 1e-3 and 1e-1 shows a steeper loss curve and you can choose a learning rate in this range to train the model. By default, the tool chooses a conservative (lower) learning rate (close to 1e-3) to ensure that it does not overshoot the minimum error loss function. However, for faster training, you may pick a higher learning rate (up to 1e-1).

The learning rate controls the weighting adjustment of the neural network. A low learning rate trains the model slowly, while a high learning rate can jump to conclusions and learn the incorrect information. You can specify a learning rate or leave the default. The default will choose the rate in which loss, or model error, is lowest before it starts to increase again, indicating that the learning rate is too high and introducing error into the model.

k. To learn more about the other Advanced parameters, point to the Geoprocessing Input Information icons.

> It can take between 45 minutes and 4 hours to train this model, depending on your computer's processing power. You have been provided the trained model file to use if you do not want to train the model.

l. If you would like to run the model, click Run.

Step 4: Review the model

The Train Deep Learning Model (Image Analyst Tools) tool trains a deep learning model and updates the model definition file (.emd) with this information. You can use this model definition file to detect, or infer, the location of the remaining swimming pools. By reviewing

the results, you can assess the model accuracy to determine if you should modify the model or proceed with your analysis.

In the remaining exercise steps, you will use the trained model file that is provided for you.

a. Open File Explorer and browse to **..\ObjectDetection\Results\PoolsModel_25_SSD**.

Note: You are browsing to the file location that you used to unzip your data.

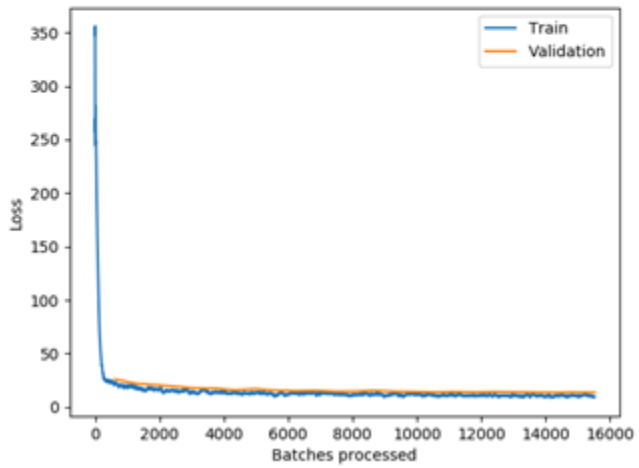
b. Double-click model_metrics.html.

SingleShotDetector

Backbone: resnet34

Learning Rate: slice(0.0005248074602497722, 0.005248074602497722, None)

Training and Validation loss



A web browser opens with model metrics that describe the following information:

- Learning rate: controls the speed at which the model is trained (how quickly the model parameters are updated). In this graphic, the learning rate shows a range of values, where the smaller number on the left is the learning rate applied to the first few layers of the network, and the larger number on the right is applied to the last few layers. The idea here is that the low learning rate trains the first few layers of the network slowly, while the higher learning rate trains the final layers of the network more quickly. The end goal is to find the highest learning rate where the loss is still improving, and that becomes your optimal learning rate.
- Training and validation loss graph: compares training and validation losses over the training epochs. A good-performing model typically shows a continual decrease in both training and validation loss over the training epochs. If the validation loss begins to increase, then you may have overfitting, where the model is recognizing a particular set of data too closely and therefore may not generalize well to other data.
- Average precision score: assesses the performance of object detection models. It measures the average precision on the validation set for each class. An average precision score ranges from 0 to 1, where values closer to 1 indicate better model performance.

c Scroll down to Ground Truth/Predictions.

Comparing the ground truth images with the predicted images will also help you determine the accuracy of your model. This model provides a good baseline, predicting most of the pools identified in the ground truth.

These metrics provided can help you determine if you should modify the parameters of this tool (learning rate, number of epochs, grid cell size, and so on) to improve the results of the models. Because modifications would require more processing time, you will proceed with this model.

d. Close the results and File Explorer, and then return to ArcGIS Pro.

Step 5: Perform inferencing using the model

After you train the model, you will perform inferencing. Inferencing uses the trained model to extract information from your imagery. In this case, you will extract, or detect, swimming pools for the specified area of interest.

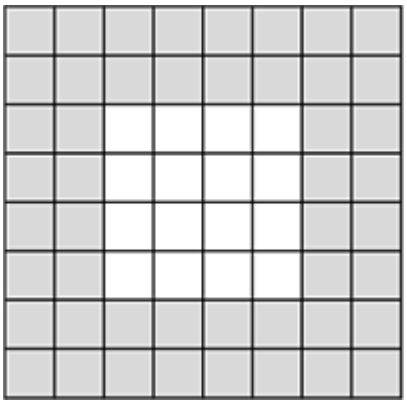
a. In the Geoprocessing pane, search for and open the **Detect Objects Using Deep Learning (Image Analyst Tools)** tool.

b Set the following tool parameters:

- Input Raster: NAIP_AOI.tif
- Output Detected Objects: **SwimmingPoolsAll**
- Model Definition: **..\ObjectDetection\Results\PoolsModel_25_SSD\PoolsModel_25_SSD.emd** *(Note: Click the Browse button.)*

You will see some items listed in the Arguments section. These arguments will be used on your image as it passes through the layers of the model. The default arguments use the values defined when training the model. You can use these values as a baseline that can be adjusted to refine the inferencing results. The following information provides explanations of each argument.

- Padding adds a border of cells around the image. This border is used to ensure that the image maintains its original size as it passes through the model. Padding is most relevant if you are detecting objects that are around the edge of your image.



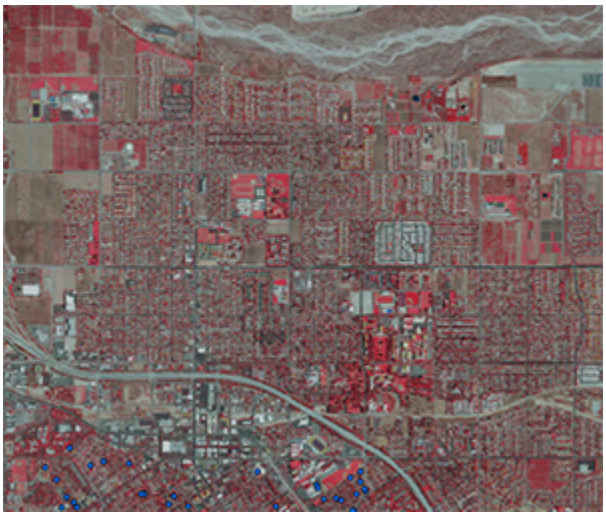
The padding in this image is 2 pixels in size, indicated in gray.

- Threshold defines the required confidence level for the object detection. In this analysis, the threshold is 0.5, meaning that the model has to be at least 50 percent confident that the object is a swimming pool.
- NMS_Overlap is the percentage of allowable overlap between features. In this analysis, features that overlap more than 10 percent will be removed.
- Exclude_Pad_Detections allows you to exclude items in the padded areas. In this analysis, you will exclude the padded areas from inferencing.

c Check the Non Maximum Suppression box.

Non Maximum Suppression will identify duplicate features. The feature with the lower confidence level will be removed. The default values for this parameter will use the confidence field to determine which feature has a lower confidence level. The Max Overlap Ratio will define how much overlap between features is allowable. The default does not allow any overlap.

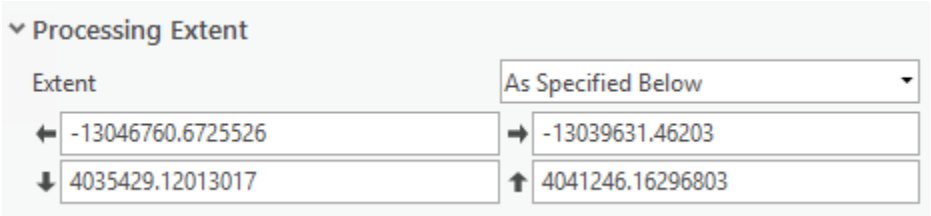
d On the ribbon, from the Map tab, zoom to the Inference Pools bookmark.



Inferencing can be a time-consuming process based on your computer's processing power and the scale of your analysis. To minimize processing time, you will analyze a smaller area of interest.

e At the top of the Geoprocessing pane, click the Environments tab.

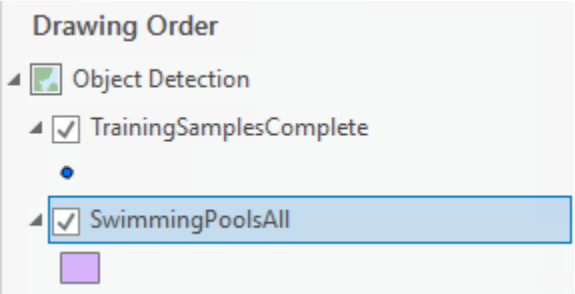
f Under Processing Extent, update Extent to Current Display Extent.



Coordinates are defined based on the current extent of the map. These coordinates will be used as the processing extent for this tool.

g Click Run.

Note: It may take several minutes for the tool to finish running.



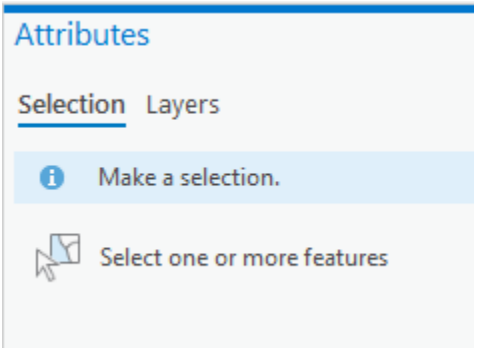
The model detected swimming pools for the specified area and created a new feature layer with the results. You will not be able to see the pools at this scale. You will zoom in to the map to review the inferencing results and assess their accuracy.

Step 6: Review inferencing results

a. In the Contents pane, click the SwimmingPoolsAll symbol.

b. In the Symbology pane, under ArcGIS 2D, click Black Outline (2 pts).

c. From the Map tab, in the Selection group, click Attributes.

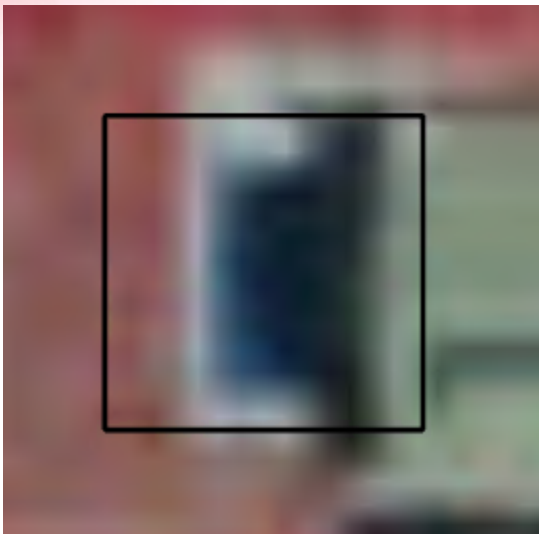


The Attributes pane opens and provides attribute information for features in a selected area. You can also use this pane to zoom to each of the features and determine if the model detected the appropriate object.

d. In the Attributes pane, click the Layers tab.

e. From the Choose A Layer drop-down list, choose SwimmingPoolsAll.

f. Under SwimmingPoolsAll, click the Step Forward button .



The map moves to a detected pool in the SwimmingPoolsAll layer. You can use this tool to review the detected pools, visually assessing the accuracy of the model results. Based on the accuracy of the model results, you would modify the model or continue running inferencing on the entire study area.

Using this model, you can quickly detect the remaining swimming pools in Southern California, providing tax assessors with the information that they need to identify more accurate property values and taxes.

g. Save the project.

h. If you would like to train the model using ArcGIS API for Python, proceed to the optional stretch goal; otherwise, exit ArcGIS Pro.

Stretch goal (Optional)

Previously, you trained the model using the Train Deep Learning Model geoprocessing tool. In this stretch goal, you will train the model using ArcGIS API for Python in a Jupyter Notebook.

The contents of the notebook that you will run describes the process of training a deep learning model using ArcGIS API for Python. It includes the ArcGIS API code and descriptions of each step in the process.

Use the following steps to complete the stretch goal:

1. From the Windows taskbar, search for and open the **Python Command Prompt**.
2. In the Python Command Prompt, type `cd`, add a space, and then add the file path where you saved the ArcGIS Pro project. For example, the file path may look like
```
cd C:\Users\Student\Documents\ObjectDetection
```
3. Press Enter.
4. In the Python Command Prompt window, type `jupyter-notebook` and press Enter.
5. In the web browser that opens, from the Files tab, click model_training.ipynb.
6. Click Run to run through each cell in the script.

> Do not run the `ssd.fit()` cell because it will begin to train the model. It can take between 45 minutes and 4 hours to train this model, depending on your computer's processing power.

7. When you are finished, close the web browser and python command prompt, and then save your project and exit ArcGIS Pro.