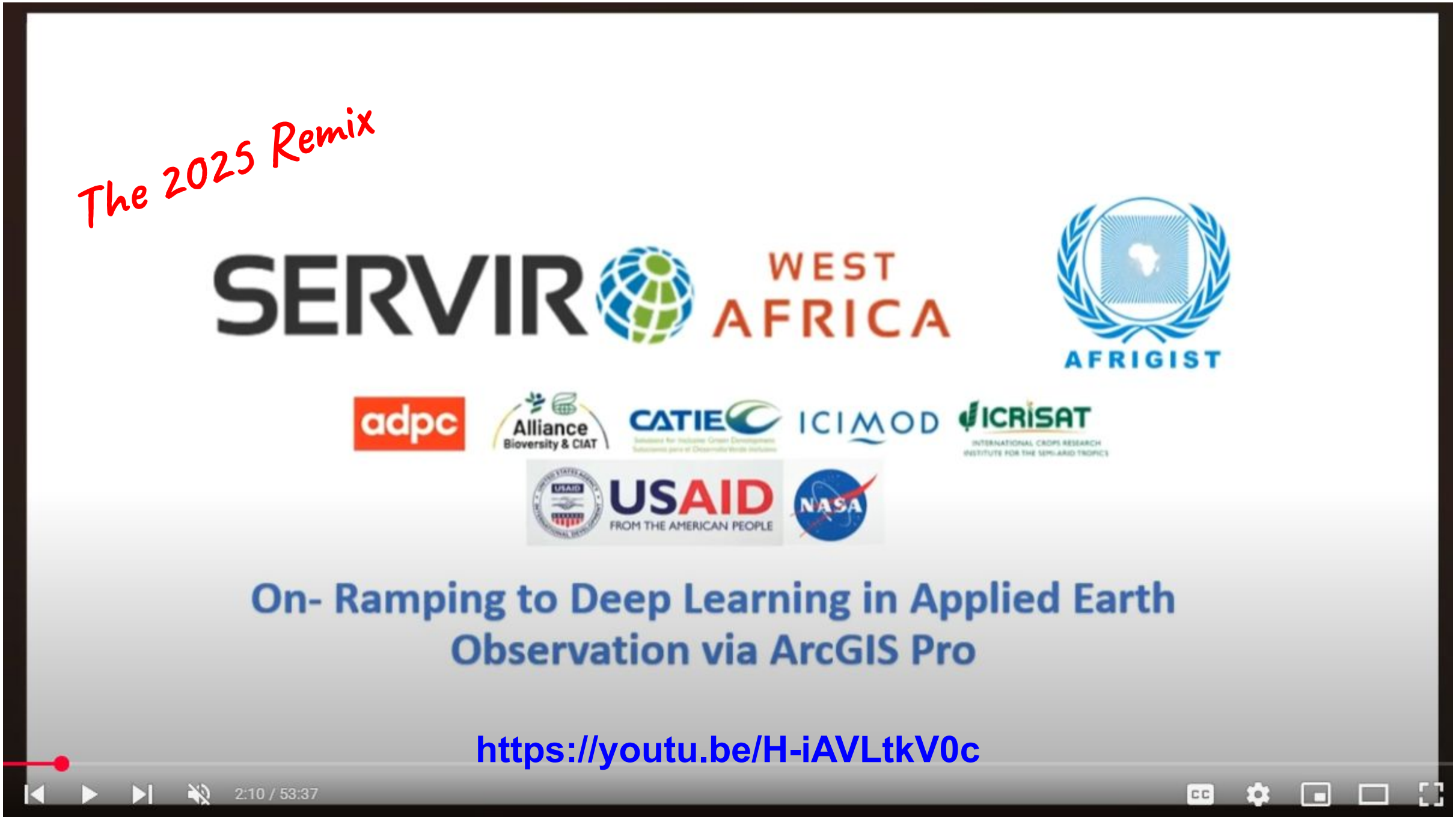
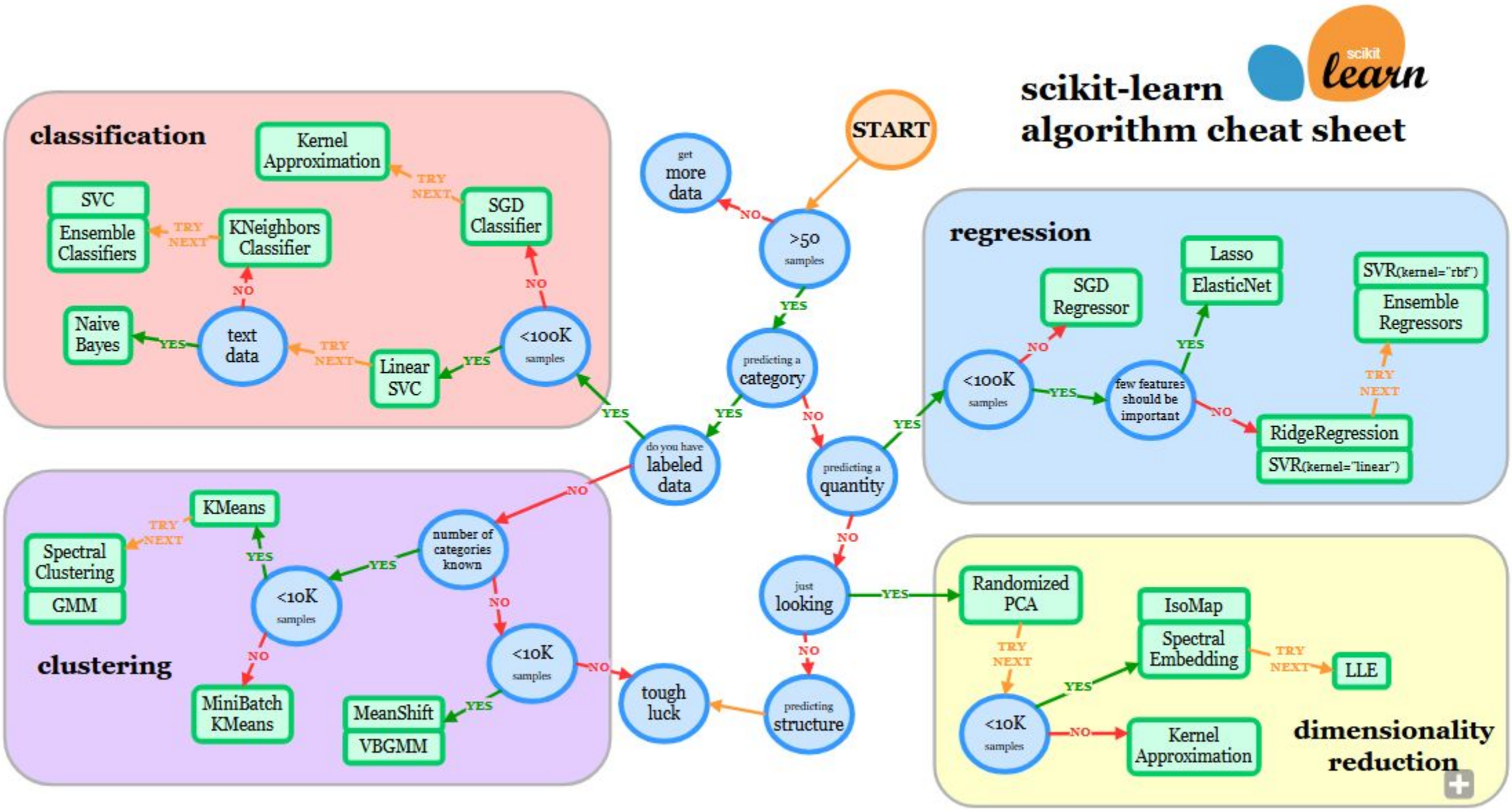
Using ArcGIS Pro as an On-Ramp to Deep Learning [in Earth Observation]

Emil Cherrington, Ph.D.
Lab for Applied Science / Earth System Science Center
University of Alabama in Huntsville

06.08.2025


The 2025 Remix
SERVIR
WEST AFRICA
AFRIGIST
adpc
Alliance Bioversity & CIAT
CATIE
ICIMOD
ICRISAT
USAID
FROM THE AMERICAN PEOPLE
NASA
On- Ramping to Deep Learning in Applied Earth Observation via ArcGIS Pro
https://youtu.be/H-iAVLtkV0c
2:10 / 53:37

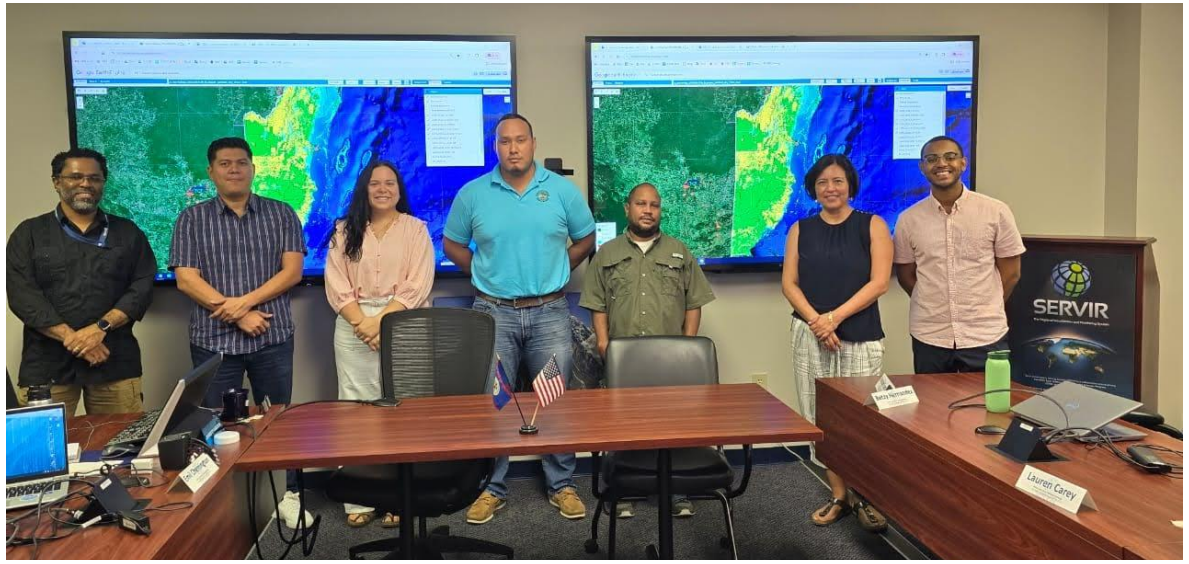


https://scikit-learn.org/stable/machine_learning_map.html

Prologue

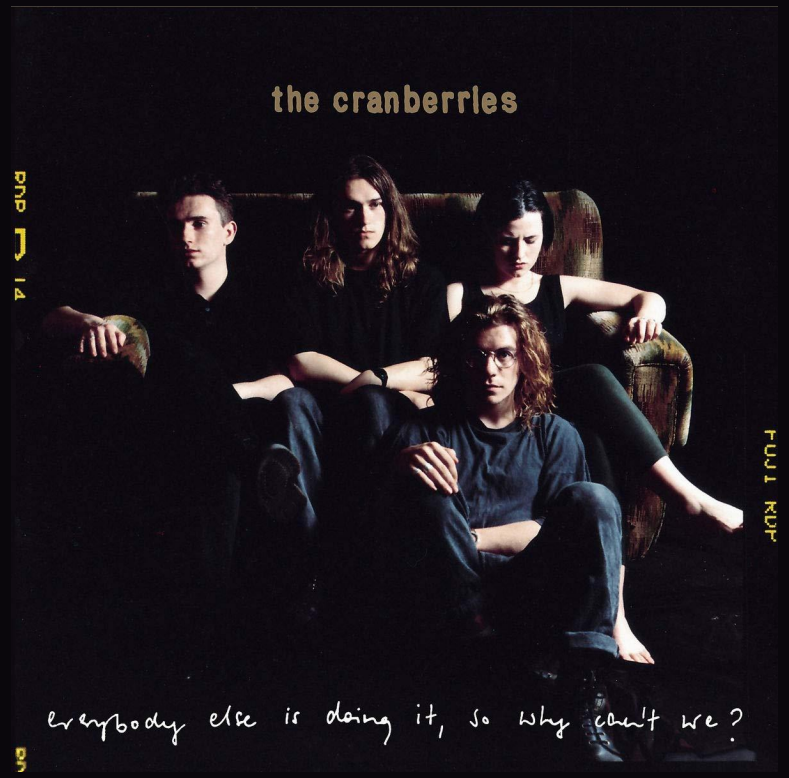
2019

2024



https://youtu.be/AWQs4Lkb-5M

The problem:


the cranberries
Everybody else is doing it, so why can't we?


EVERYBODY ELSE IS DOING
DEEP LEARNING,
SO WHY CAN'T WE?
NEURAL NETWORKS
NETWORKS ILLUSTRATED
PYTHON for DUMMIES
ai

The solution:

…But how?

Outline

- Objectives
- Software required
- Process overview
- Step 1: Exporting training data
- Step 2: Training the DL model
- Step 3: Pixel classification (model inferencing)
- Step 4: Reviewing model performance
- Step 5: Reviewing model outputs
- Summary
- Future work

Objectives

- Explain how to execute a deep learning-based land cover classification [using Esri's ArcGIS Pro 3.x]
  - *Explore the potential challenges one might have in trying to execute land cover classification due to the nuances of the software*
  - *Document best practices needed to get your classification to work*

Software Required

+

README

Deep Learning Libraries Installers for ArcGIS

PyTorch
scikit-image
scikit learn
Keras
cuDNN
plotly
ONNX
DASK
fast.ai
TensorFlow spaCy

ArcGIS Pro, Server and the ArcGIS API for Python all include tools to use AI and Deep Learning to solve geospatial problems, such as feature extraction, pixel classification, and feature categorization. This installer includes a broad collection of components, such as PyTorch, TensorFlow, Fast.ai and scikit-learn, for performing deep learning and machine learning tasks, a total collection of 254 packages. These packages can be used with the Deep Learning Training tools, interactive object detection, by using the `arcgis.learn` module within the ArcGIS API for Python, and directly imported into your own scripts and tools. Most of the tools in this collection will work on any machine, but common deep learning workflows require a recent NVIDIA graphics processing unit (GPU), and problem sizes are bound by available GPU memory, see the requirements section.

This installer adds all the included packages to the default `arcgispro-py3` environment that Pro and Server both ship with, and no additional environments are necessary in order to get started using the tools. If you do create custom environments, these packages will also be included so you can use the same tools in your own custom environments as well.

For an example of the kinds of workflows this installer and ArcGIS enables, see the AI & Deep Learning in the UC 2020 Plenary video

120 forks
Report repository

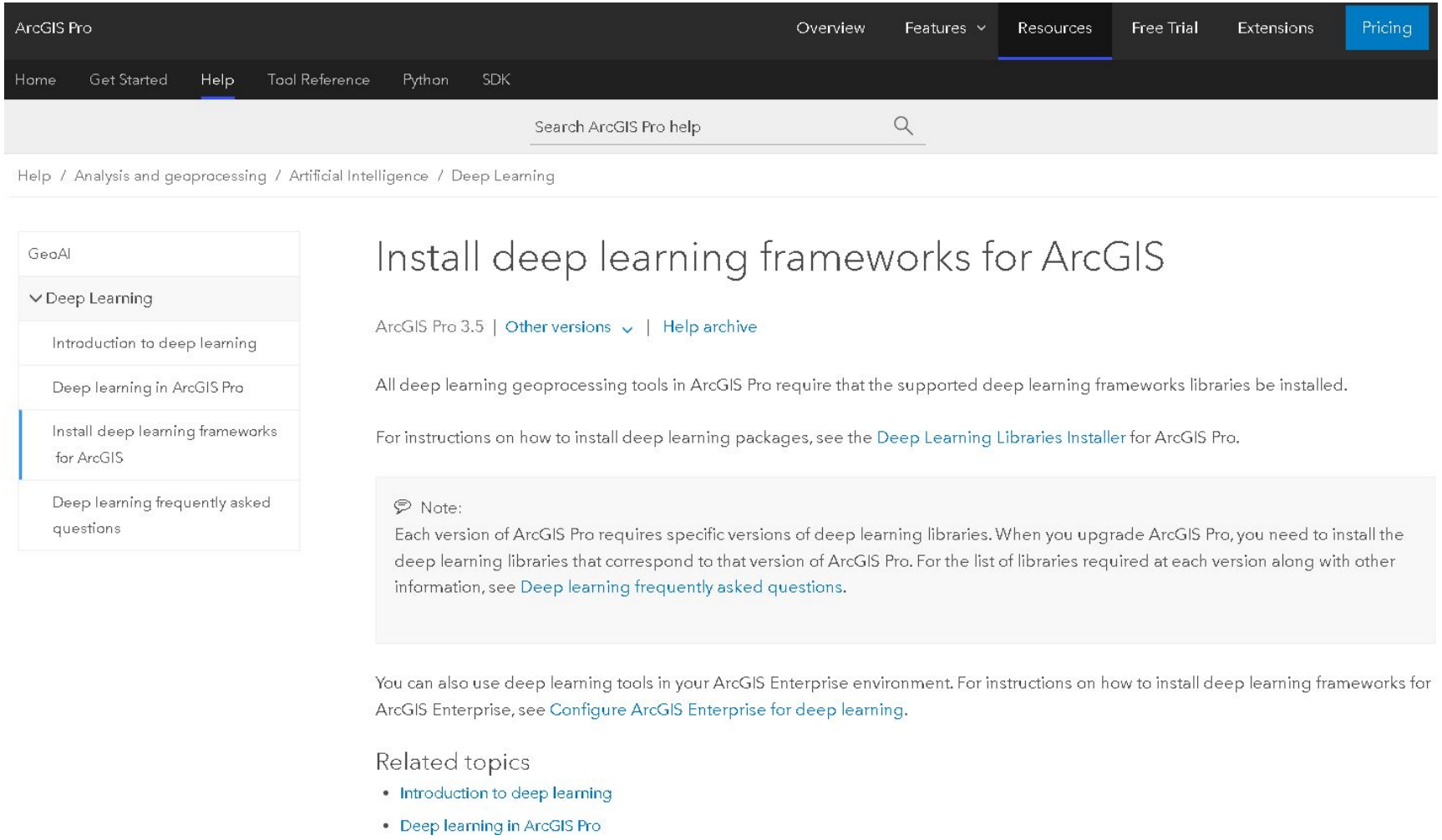
Releases 7
Deep Learning Libraries installer... Latest
on Jun 17, 2021
+ 6 releases

Packages
No packages published

Contributors 10

https://github.com/Esri/deep-learning-frameworks?tab=readme-ov-file

Installing DL frameworks



https://pro.arcgis.com/en/pro-app/latest/help/analysis/deep-learning/install-deep-learning-frameworks.htm

https://links.esri.com/pro-deep-learning-libraries/35

Process Overview: Deep Learning-based Classification

0. Collect training data ~a few hours

1. Export training data ~1-5 min.

2. Train deep learning (DL) model ~2-3 hours [laptop w/o GPU] ~5-30 min. [laptop w/ GPU]

~30 min.** 3. Classify pixels using DL model (perform inferencing)

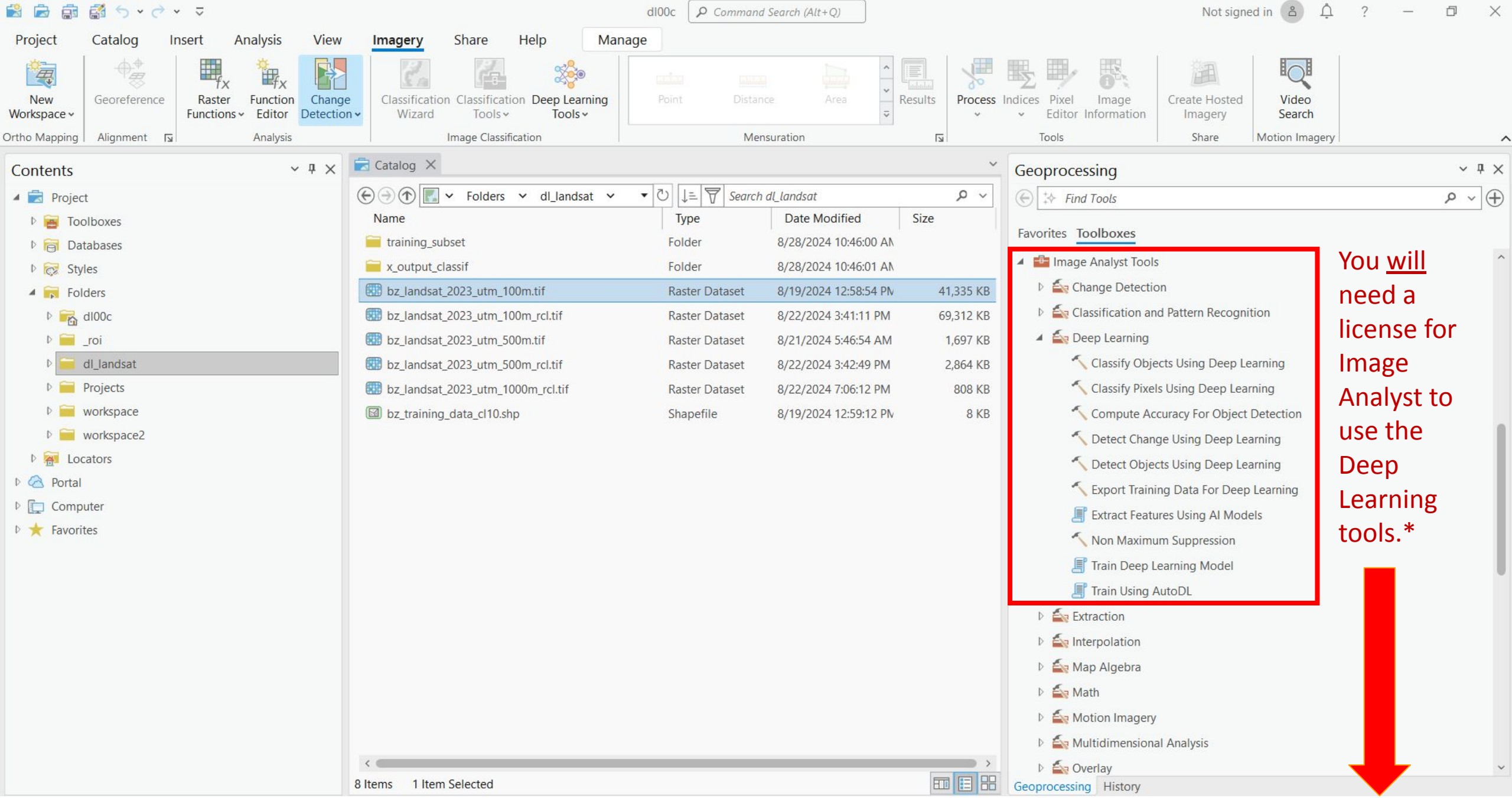
Please note:

1. For this example, I focus on the ~23,000 km^2 country of Belize, because it's relatively small.

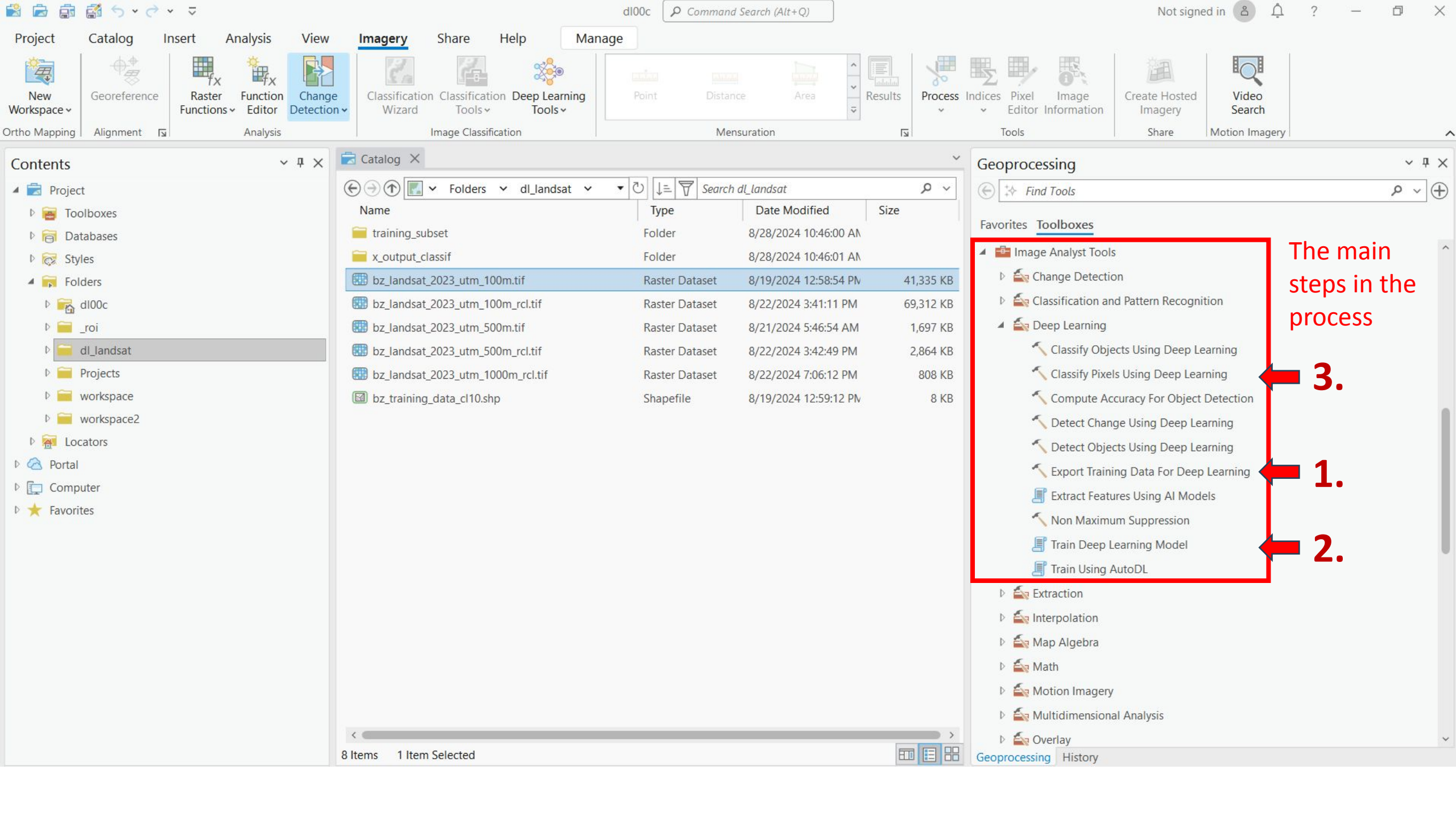
2. I'm using a 2023 Landsat-9 image mosaic as the input raster bands, w/ high quality training data I've collected.

3. I have resampled the data from 30m to 100m resolution to *try* to speed up the processing.

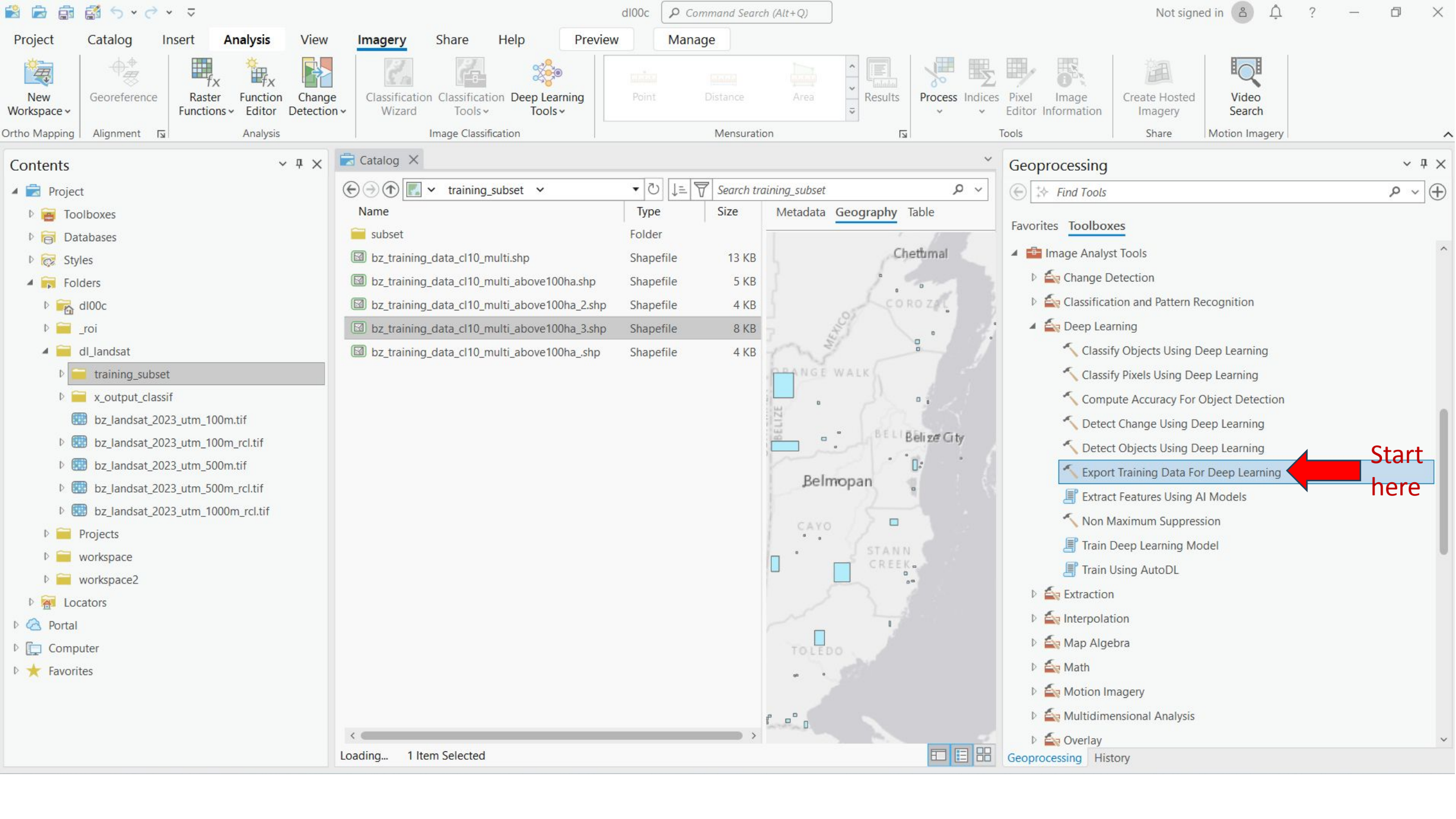
4. You should definitely test DL-based land cover classification on *your* own area of interest.

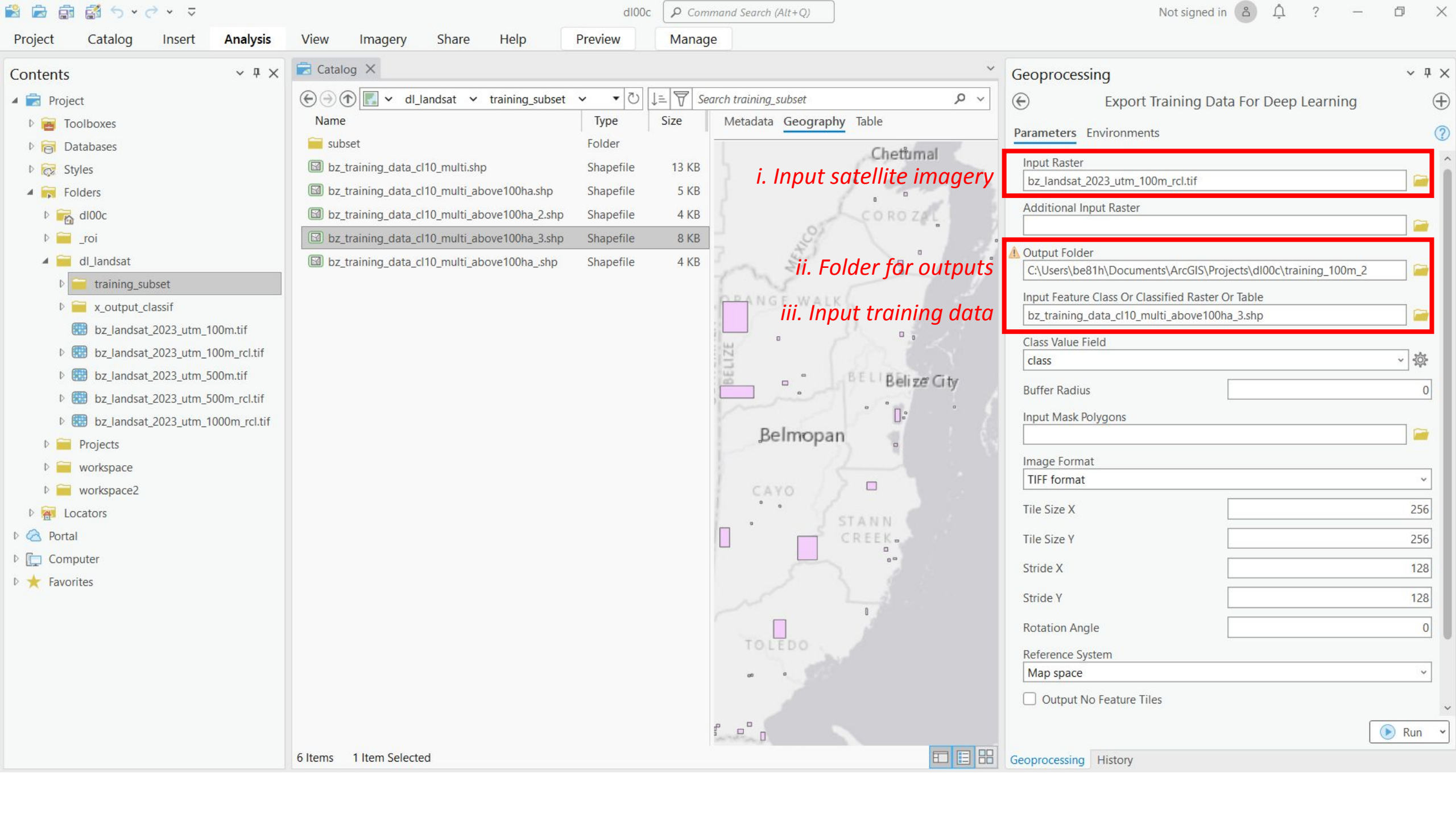


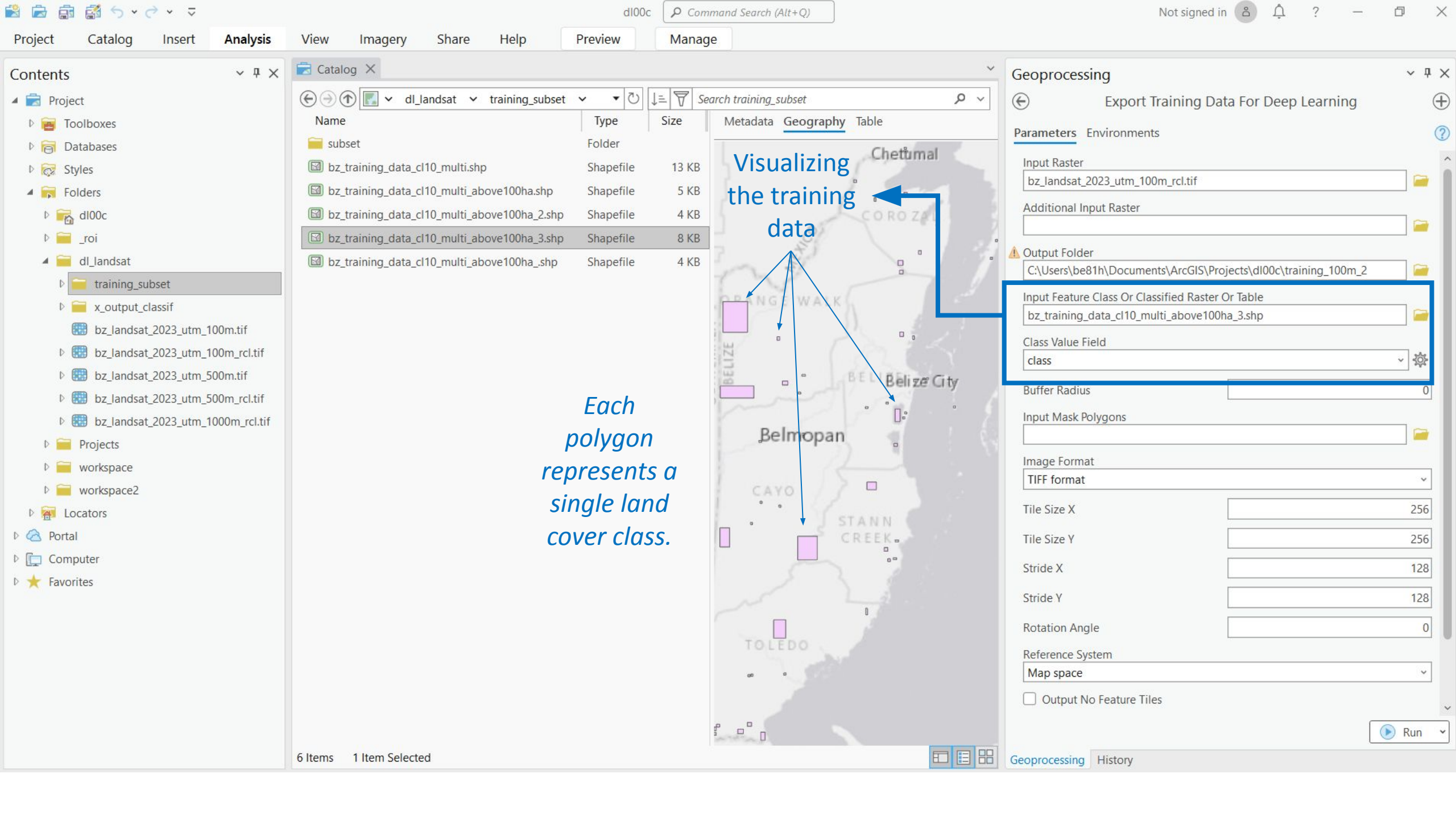
https://www.esri.com/en-us/arcgis/products/arcgis-pro/trial

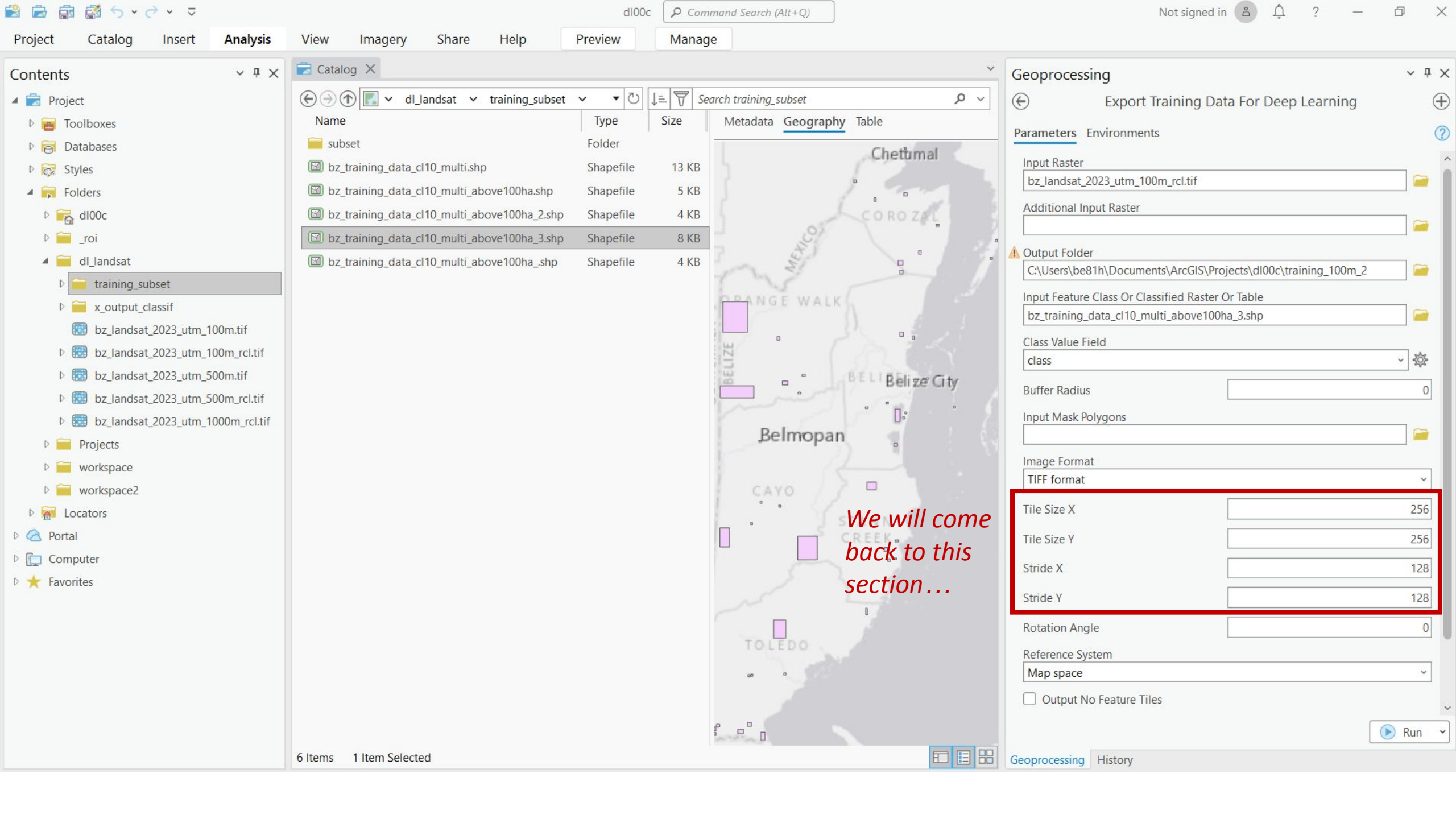

dl00c
Command Search (Alt+Q)
Not signed in
Project
Catalog
Insert
Analysis
View
Imagery
Share
Help
Manage
New Workspace
Georeference
Raster Functions
Function Editor
Change Detection
Classification Wizard
Classification Tools
Deep Learning Tools
Point
Distance
Area
Results
Process
Indices
Pixel Editor
Image Information
Create Hosted Imagery
Video Search
Ortho Mapping
Alignment
Analysis
Image Classification
Mensuration
Tools
Share
Motion Imagery
Contents
Project
Toolboxes
Databases
Styles
Folders
dl00c
_roi
dl_landsat
Projects
workspace
workspace2
Locators
Portal
Computer
Favorites
Catalog
Folders
dl_landsat
Search dl_landsat
Name
Type
Date Modified
Size
training_subset
Folder
8/28/2024 10:46:00 AM
x_output_classif
Folder
8/28/2024 10:46:01 AM
bz_landsat_2023_utm_100m.tif
Raster Dataset
8/19/2024 12:58:54 PM
41,335 KB
bz_landsat_2023_utm_100m_rcl.tif
Raster Dataset
8/22/2024 3:41:11 PM
69,312 KB
bz_landsat_2023_utm_500m.tif
Raster Dataset
8/21/2024 5:46:54 AM
1,697 KB
bz_landsat_2023_utm_500m_rcl.tif
Raster Dataset
8/22/2024 3:42:49 PM
2,864 KB
bz_landsat_2023_utm_1000m_rcl.tif
Raster Dataset
8/22/2024 7:06:12 PM
808 KB
bz_training_data_cl10.shp
Shapefile
8/19/2024 12:59:12 PM
8 KB
8 Items
1 Item Selected
Geoprocessing
Find Tools
Favorites
Toolboxes
Image Analyst Tools
Change Detection
Classification and Pattern Recognition
Deep Learning
Classify Objects Using Deep Learning
Classify Pixels Using Deep Learning
Compute Accuracy For Object Detection
Detect Change Using Deep Learning
Detect Objects Using Deep Learning
Export Training Data For Deep Learning
Extract Features Using AI Models
Non Maximum Suppression
Train Deep Learning Model
Train Using AutoDL
Extraction
Interpolation
Map Algebra
Math
Motion Imagery
Multidimensional Analysis
Overlay
Geoprocessing
History
The main steps in the process
3.
1.
2.

Step 1: Exporting the Training Data

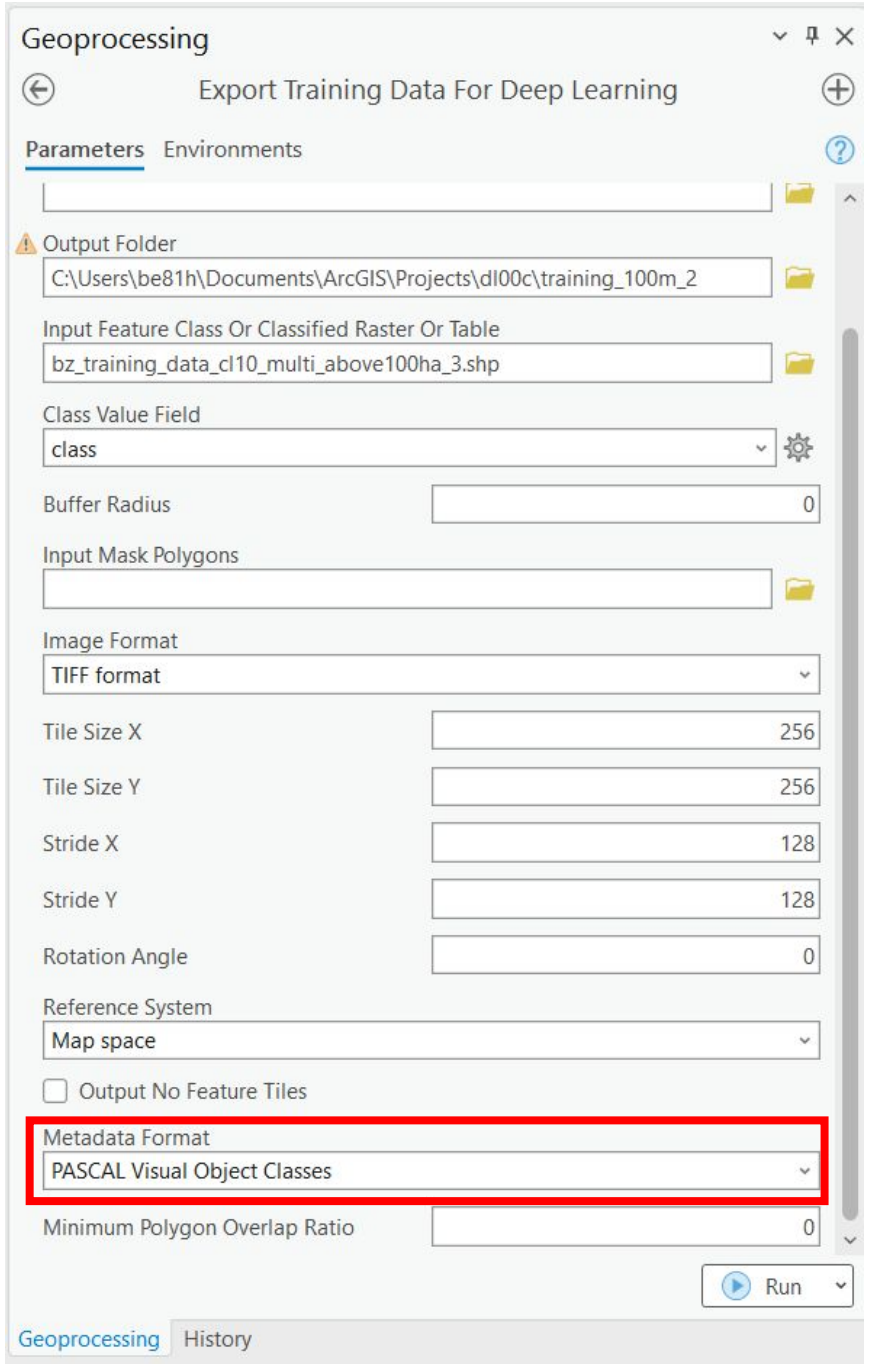

dl00c
Command Search (Alt+Q)
Not signed in
Project
Catalog
Insert
Analysis
View
Imagery
Share
Help
Preview
Manage
New Workspace
Georeference
Raster Functions
Function Editor
Change Detection
Classification Wizard
Classification Tools
Deep Learning Tools
Point
Distance
Area
Results
Process
Indices
Pixel Editor
Image Information
Create Hosted Imagery
Video Search
Ortho Mapping
Alignment
Analysis
Image Classification
Mensuration
Tools
Share
Motion Imagery
Contents
Project
Toolboxes
Databases
Styles
Folders
dl00c
_roi
dl_landsat
training_subset
x_output_classif
bz_landsat_2023_utm_100m.tif
bz_landsat_2023_utm_100m_rcl.tif
bz_landsat_2023_utm_500m.tif
bz_landsat_2023_utm_500m_rcl.tif
bz_landsat_2023_utm_1000m_rcl.tif
Projects
workspace
workspace2
Locators
Portal
Computer
Favorites
Catalog
training_subset
Search training_subset
Name
Type
Size
subset
Folder
bz_training_data_cl10_multi.shp
Shapefile
13 KB
bz_training_data_cl10_multi_above100ha.shp
Shapefile
5 KB
bz_training_data_cl10_multi_above100ha_2.shp
Shapefile
4 KB
bz_training_data_cl10_multi_above100ha_3.shp
Shapefile
8 KB
bz_training_data_cl10_multi_above100ha_.shp
Shapefile
4 KB
Metadata
Geography
Table
Chetumal
CORozal
MEXICO
ORANGE WALK
BELIZE
Belize City
Belmopan
CAYO
STANN CREEK
TOLEDO
Loading...
1 Item Selected
Geoprocessing
Find Tools
Favorites
Toolboxes
Image Analyst Tools
Change Detection
Classification and Pattern Recognition
Deep Learning
Classify Objects Using Deep Learning
Classify Pixels Using Deep Learning
Compute Accuracy For Object Detection
Detect Change Using Deep Learning
Detect Objects Using Deep Learning
Export Training Data For Deep Learning
Extract Features Using AI Models
Non Maximum Suppression
Train Deep Learning Model
Train Using AutoDL
Extraction
Interpolation
Map Algebra
Math
Motion Imagery
Multidimensional Analysis
Overlay
Start here
Geoprocessing
History


dl00c
Command Search (Alt+Q)
Not signed in
Project
Catalog
Insert
Analysis
View
Imagery
Share
Help
Preview
Manage
Contents
Project
Toolboxes
Databases
Styles
Folders
dl00c
_roi
dl_landsat
training_subset
x_output_classif
bz_landsat_2023_utm_100m.tif
bz_landsat_2023_utm_100m_rcl.tif
bz_landsat_2023_utm_500m.tif
bz_landsat_2023_utm_500m_rcl.tif
bz_landsat_2023_utm_1000m_rcl.tif
Projects
workspace
workspace2
Locators
Portal
Computer
Favorites
Catalog
dl_landsat
training_subset
Search training_subset
Name
Type
Size
subset
Folder
bz_training_data_cl10_multi.shp
Shapefile
13 KB
bz_training_data_cl10_multi_above100ha.shp
Shapefile
5 KB
bz_training_data_cl10_multi_above100ha_2.shp
Shapefile
4 KB
bz_training_data_cl10_multi_above100ha_3.shp
Shapefile
8 KB
bz_training_data_cl10_multi_above100ha_.shp
Shapefile
4 KB
Metadata
Geography
Table
Chetumal
COROZAL
MEXICO
ORANGE WALK
BELIZE
Belize City
Belmopan
CAYO
STANN CREEK
TOLEDO
i. Input satellite imagery
ii. Folder for outputs
iii. Input training data
6 Items
1 Item Selected
Geoprocessing
Export Training Data For Deep Learning
Parameters
Environments
Input Raster
bz_landsat_2023_utm_100m_rcl.tif
Additional Input Raster
Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m_2
Input Feature Class Or Classified Raster Or Table
bz_training_data_cl10_multi_above100ha_3.shp
Class Value Field
class
Buffer Radius
0
Input Mask Polygons
Image Format
TIFF format
Tile Size X
256
Tile Size Y
256
Stride X
128
Stride Y
128
Rotation Angle
0
Reference System
Map space
Output No Feature Tiles
Run
Geoprocessing
History


dl00c
Command Search (Alt+Q)
Not signed in
Project
Catalog
Insert
Analysis
View
Imagery
Share
Help
Preview
Manage
Contents
Project
Toolboxes
Databases
Styles
Folders
dl00c
_roi
dl_landsat
training_subset
x_output_classif
bz_landsat_2023_utm_100m.tif
bz_landsat_2023_utm_100m_rcl.tif
bz_landsat_2023_utm_500m.tif
bz_landsat_2023_utm_500m_rcl.tif
bz_landsat_2023_utm_1000m_rcl.tif
Projects
workspace
workspace2
Locators
Portal
Computer
Favorites
Catalog
dl_landsat
training_subset
Search training_subset
Name
Type
Size
subset
Folder
bz_training_data_cl10_multi.shp
Shapefile
13 KB
bz_training_data_cl10_multi_above100ha.shp
Shapefile
5 KB
bz_training_data_cl10_multi_above100ha_2.shp
Shapefile
4 KB
bz_training_data_cl10_multi_above100ha_3.shp
Shapefile
8 KB
bz_training_data_cl10_multi_above100ha_.shp
Shapefile
4 KB
6 Items
1 Item Selected
Visualizing the training data
Each polygon represents a single land cover class.
Metadata
Geography
Table
Chetumal
Belize City
Belmopan
BELIZE
CAYO
STANN CREEK
TOLEDO
Geoprocessing
Export Training Data For Deep Learning
Parameters
Environments
Input Raster
bz_landsat_2023_utm_100m_rcl.tif
Additional Input Raster
Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m_2
Input Feature Class Or Classified Raster Or Table
bz_training_data_cl10_multi_above100ha_3.shp
Class Value Field
class
Buffer Radius
0
Input Mask Polygons
Image Format
TIFF format
Tile Size X
256
Tile Size Y
256
Stride X
128
Stride Y
128
Rotation Angle
0
Reference System
Map space
Output No Feature Tiles
Run
Geoprocessing
History


dl00c
Command Search (Alt+Q)
Not signed in
Project
Catalog
Insert
Analysis
View
Imagery
Share
Help
Preview
Manage
Contents
Project
Toolboxes
Databases
Styles
Folders
dl00c
_roi
dl_landsat
training_subset
x_output_classif
bz_landsat_2023_utm_100m.tif
bz_landsat_2023_utm_100m_rcl.tif
bz_landsat_2023_utm_500m.tif
bz_landsat_2023_utm_500m_rcl.tif
bz_landsat_2023_utm_1000m_rcl.tif
Projects
workspace
workspace2
Locators
Portal
Computer
Favorites
Catalog
dl_landsat
training_subset
Search training_subset
Name
Type
Size
subset
Folder
bz_training_data_cl10_multi.shp
Shapefile
13 KB
bz_training_data_cl10_multi_above100ha.shp
Shapefile
5 KB
bz_training_data_cl10_multi_above100ha_2.shp
Shapefile
4 KB
bz_training_data_cl10_multi_above100ha_3.shp
Shapefile
8 KB
bz_training_data_cl10_multi_above100ha_.shp
Shapefile
4 KB
Metadata
Geography
Table
Chetumal
COROZAL
MEXICO
ORANGE WALK
BELIZE
Belize City
Belmopan
CAYO
TOLEDO
We will come back to this section…
6 Items
1 Item Selected
Geoprocessing
Export Training Data For Deep Learning
Parameters
Environments
Input Raster
bz_landsat_2023_utm_100m_rcl.tif
Additional Input Raster
Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m_2
Input Feature Class Or Classified Raster Or Table
bz_training_data_cl10_multi_above100ha_3.shp
Class Value Field
class
Buffer Radius
0
Input Mask Polygons
Image Format
TIFF format
Tile Size X
256
Tile Size Y
256
Stride X
128
Stride Y
128
Rotation Angle
0
Reference System
Map space
Output No Feature Tiles
Run
Geoprocessing
History

Issue 1: Making sure we select the correct metadata type



The default on this tool is "PASCAL Visual Object Classes," but we need to change this to perform classifications.

i. This has been noted as an issue by Esri.

ii. The solution is to use the 'Classified Tiles' option during the training step.

Technical Support | Overview | Product Support

Description

When attempting to Train a Deep Learning Model it is possible to add image chips but not be presented with the desired model type.
If the desired model type is selected first, and the image chips are then added, Error 00800 may be returned.

Cause

The cause of the model type not being found and Error 00800 is an incompatible Meta Data Format selected when exporting training data with the Label Objects for the Deep Learning tool. The Meta Data Format dictates what model types the user can use and/or see when Training Deep Learning Models.

Solution or Workaround

To resolve this issue it is necessary to change the Meta Data Format to one compatible with the required model type. The below tables show which model types are compatible with Meta Data Format types.

Classify Pixels Using Deep Learning									
	Meta Data Format Type								
Model Type		DeepLabV3	U-Net	The Pyramid Scene Parsing Network	BDCN Edge Detector	HED Edge Detector	Multi Task Road Extractor	ConnectNet	Change Detector
	KITTI Labels	No	No	No	No	No	No	No	No
	PASCAL Visual Object Classes	No	No	No	No	No	No	No	No
	Classified Tiles	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
	RCNN Masks	No	No	No	No	No	No	No	No
	Labeled Tiles	No	No	No	No	No	No	No	No
	Export Tiles	No	No	No	No	No	No	No	No
	CycleGAN	No	No	No	No	No	No	No	No
	Multi-labeled Tiles	No	No	No	No	No	No	No	No

esri Products | Industries | Support & Services | Stories | About

Technical Support | Overview | Product Support

PROBLEM

Train Deep Learning Model - Model Type not found or Error 00800

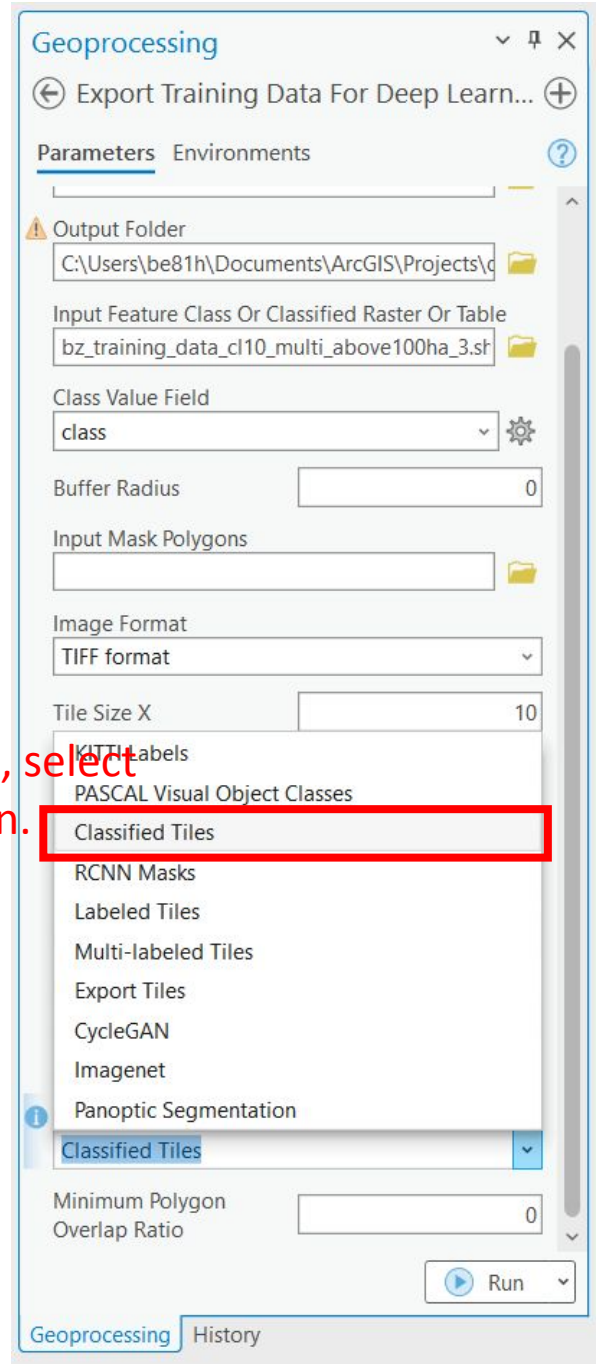
Last Published: May 6, 2024

1. This is the default

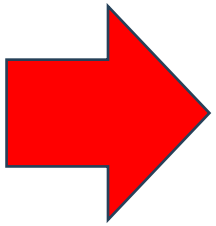
This format is useful for "detecting objects using DL," but not standard land cover classification.

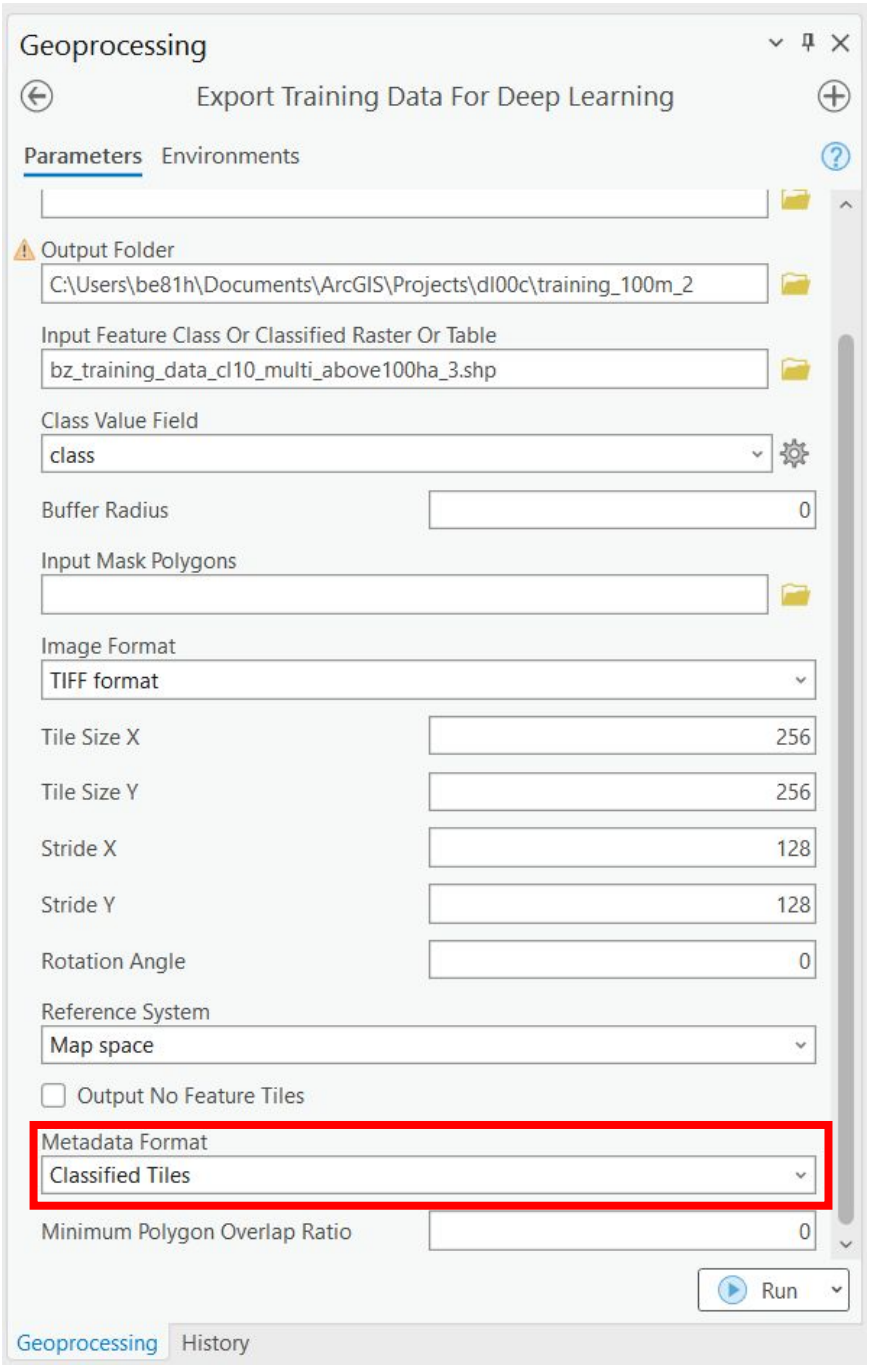
2. But <u>this</u> is what you need to run a land cover classification.

https://support.esri.com/en-us/knowledge-base/problem-train-deep-learning-model-model-type-not-found--000025249



Therefore, select this option.





Issue 2: Making sure we select an appropriate patch size.

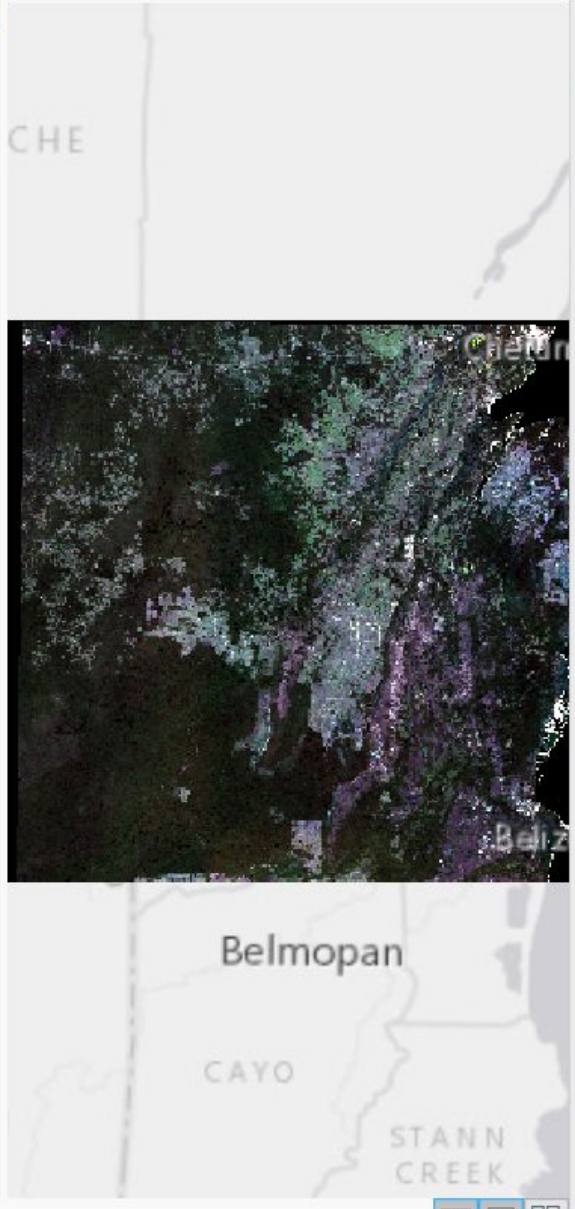
Catalog

training_500m > images

Search images

Name	Type	Size
000000000000.tif	Raster Data	1,279 KB
000000000001.tif	Raster Data	802 KB
000000000002.tif	Raster Data	1,273 KB
000000000003.tif	Raster Data	684 KB
000000000004.tif	Raster Data	155 KB
000000000005.tif	Raster Data	61 KB
000000000006.tif	Raster Data	1,193 KB
000000000007.tif	Raster Data	562 KB
000000000008.tif	Raster Data	777 KB
000000000009.tif	Raster Data	267 KB
000000000010.tif	Raster Data	221 KB

Metadata | Geography | Table



11 Items 1 Item Selected

Geoprocessing

Export Training Data For Deep Learning

Parameters | Environments

Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m_2

Input Feature Class Or Classified Raster Or Table
bz_training_data_cl10_multi_above100ha_3.shp

Class Value Field
class

Buffer Radius 0

Input Mask Polygons

Image Format
TIFF format

Tile Size X 256

Tile Size Y 256

Stride X 128

Stride Y 128

Rotation Angle 0

Reference System
Map space

Output No Feature Tiles

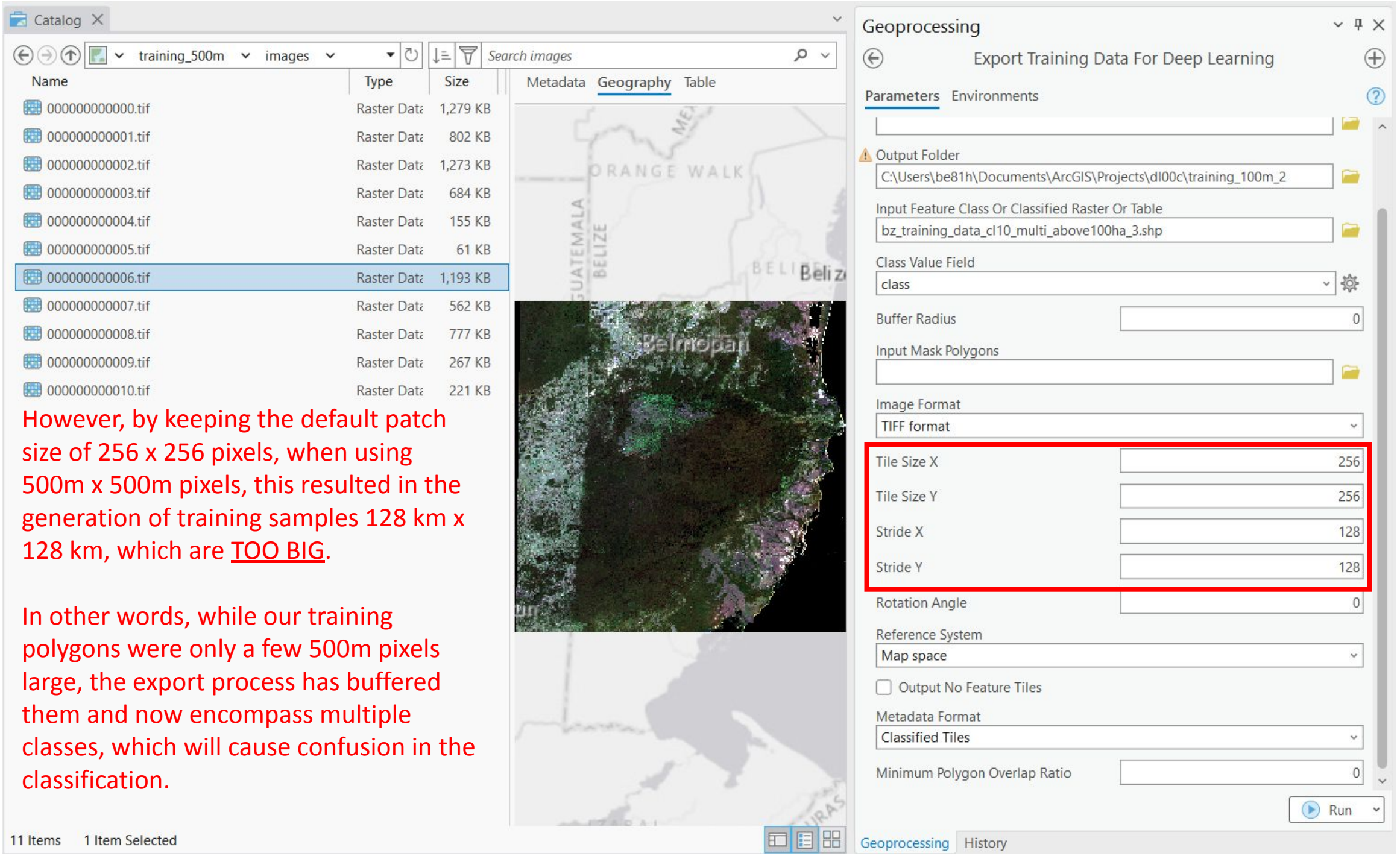
Metadata Format
Classified Tiles

Minimum Polygon Overlap Ratio 0

The default patch size is 256 x 256 pixels.

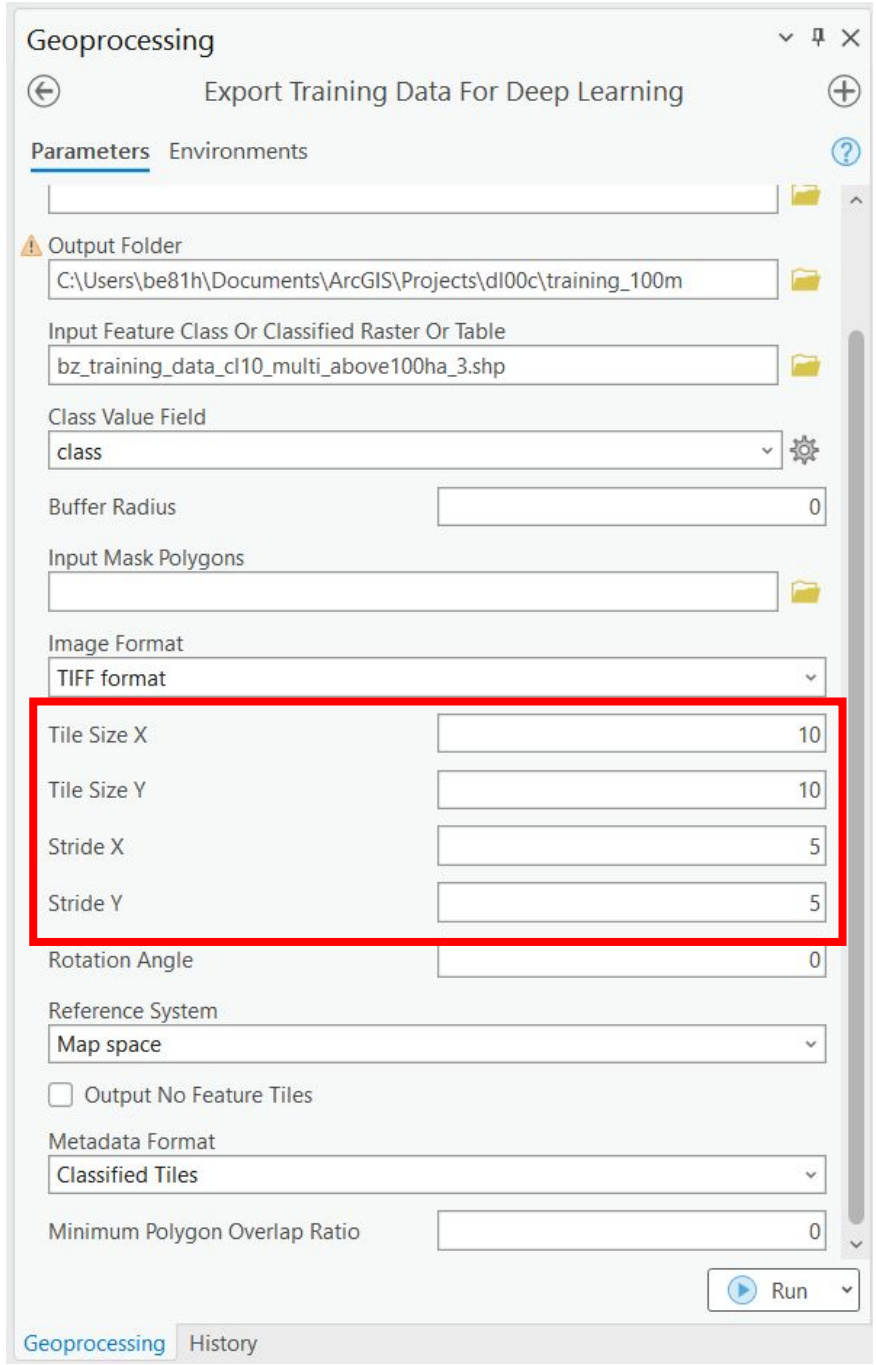
Run

Geoprocessing | History



However, by keeping the default patch size of 256 x 256 pixels, when using 500m x 500m pixels, this resulted in the generation of training samples 128 km x 128 km, which are TOO BIG.

In other words, while our training polygons were only a few 500m pixels large, the export process has buffered them and now encompass multiple classes, which will cause confusion in the classification.



i. We readjusted our strategy to use 100m x 100m pixel data instead of 500m x 500m pixel data.

ii. We adjusted down the patch size to 10 x 10 pixels, so that the patches would be 1 km x 1 km instead of 128 km x 128 km.

iii. We filtered our training data to ensure that we didn't have training samples that were smaller than 1km x 1km.

Catalog

training_100m > images

Search images

Name	Type	Size
000000018511.tif	Raster Data	5 KB
000000018512.tif	Raster Data	5 KB
000000018513.tif	Raster Data	5 KB
000000018514.tif	Raster Data	5 KB
000000018515.tif	Raster Data	5 KB
000000018516.tif	Raster Data	5 KB
000000018517.tif	Raster Data	5 KB
000000018518.tif	Raster Data	5 KB
000000018519.tif	Raster Data	5 KB
000000018520.tif	Raster Data	5 KB
000000018521.tif	Raster Data	5 KB
000000018522.tif	Raster Data	5 KB
000000018523.tif	Raster Data	5 KB
000000018524.tif	Raster Data	5 KB
000000018525.tif	Raster Data	5 KB
000000018526.tif	Raster Data	5 KB
000000018527.tif	Raster Data	5 KB
000000018528.tif	Raster Data	5 KB
000000018529.tif	Raster Data	5 KB
000000018530.tif	Raster Data	5 KB
000000018531.tif	Raster Data	5 KB
000000018532.tif	Raster Data	5 KB
000000018533.tif	Raster Data	5 KB
000000018534.tif	Raster Data	5 KB
000000018535.tif	Raster Data	5 KB
000000018536.tif	Raster Data	5 KB

3655 Items | 1 Item Selected

Metadata | Geography | Table

Here's what the final training samples looked like after exporting.

10 pixels x 10 pixels

(1km x 1km patch)

Geoprocessing

Export Training Data For Deep Learning

Parameters | Environments

Output Folder: C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m

Input Feature Class Or Classified Raster Or Table: bz_training_data_cl10_multi_above100ha_3.shp

Class Value Field: class

Buffer Radius: 0

Input Mask Polygons:

Image Format: TIFF format

Tile Size X: 10

Tile Size Y: 10

Stride X: 5

Stride Y: 5

Rotation Angle: 0

Reference System: Map space

Output No Feature Tiles

Metadata Format: Classified Tiles

Minimum Polygon Overlap Ratio: 0

Run

Geoprocessing | History

Final options selected:

Geoprocessing

Export Training Data For Deep Learning

Parameters | Environments

Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m

Input Feature Class Or Classified Raster Or Table
bz_training_data_cl10_multi_above100ha_3.shp

Class Value Field
class

Buffer Radius: 0

Input Mask Polygons

Image Format
TIFF format

Tile Size X: 10

Tile Size Y: 10

Stride X: 5

Stride Y: 5

Rotation Angle: 0

Reference System
Map space

Output No Feature Tiles

Metadata Format
Classified Tiles

Minimum Polygon Overlap Ratio: 0

Run

Geoprocessing | History

Export Training Data For Deep Learning (Image Analyst Tools)

Started: Friday, August 23, 2024 at 11:37:49 PM

Completed: Friday, August 23, 2024 at 11:39:03 PM

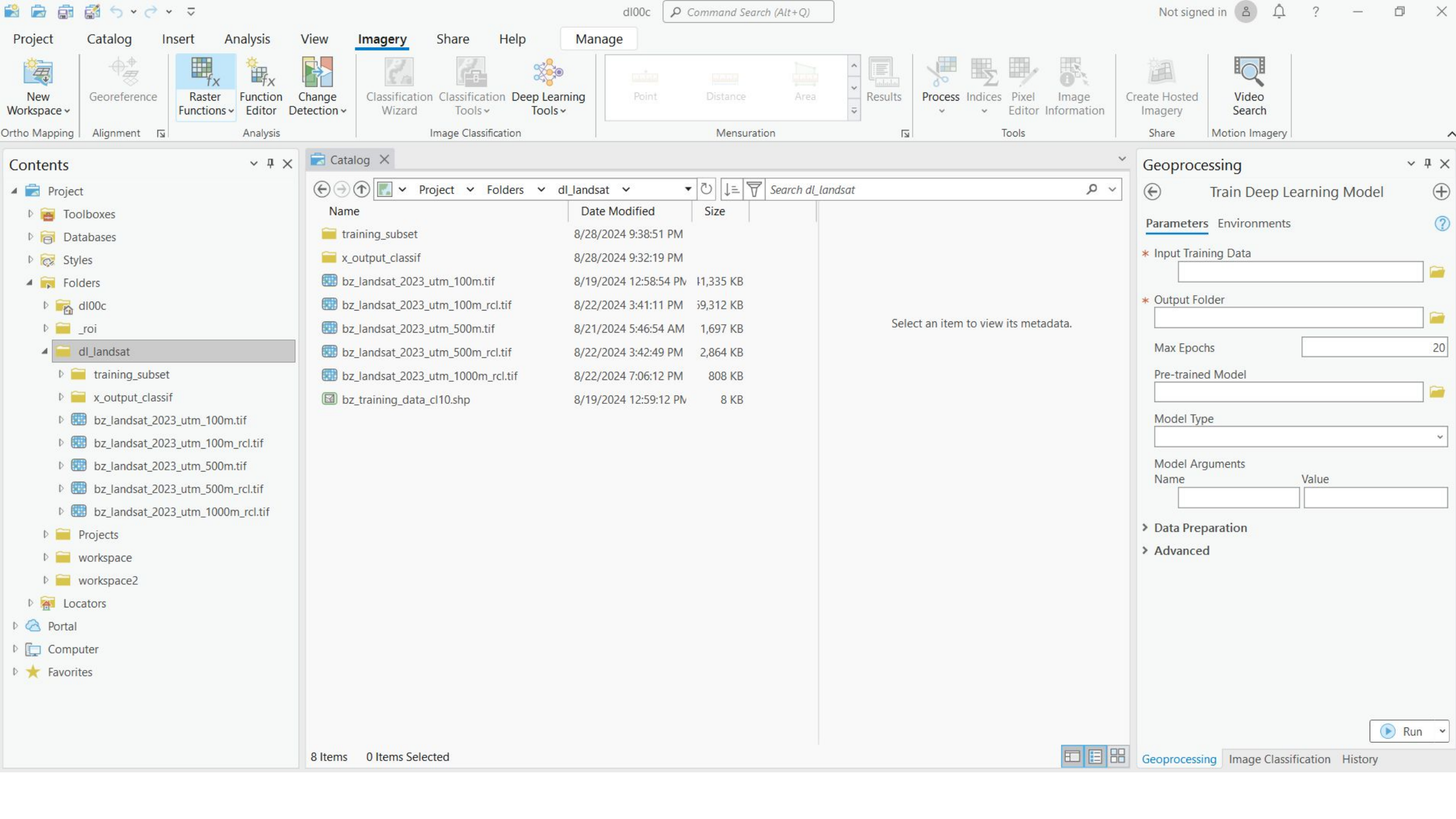
Elapsed Time: 1 Minute 14 Seconds

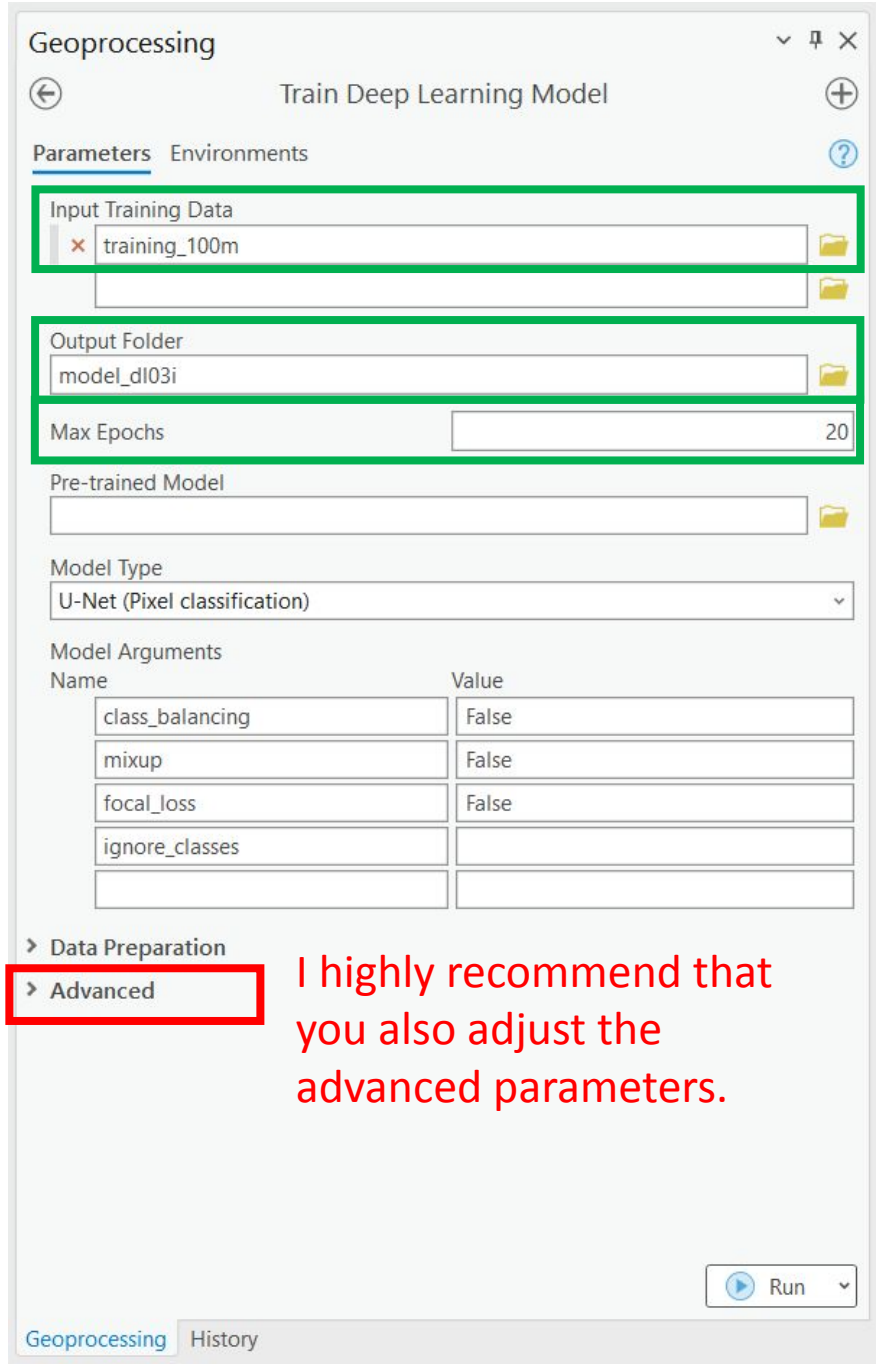
Parameters | Environments | Messages (2)

Input Raster	C:\gis\dl_landsat\bz_landsat_2023_utm_100m_rcl.tif
Output Folder	C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m
Input Feature Class Or Classified Raster Or Table	C:\gis\dl_landsat\training_subset\bz_training_data_cl10_multi_above100ha_3.shp
Image Format	TIFF
Tile Size X	10
Tile Size Y	10
Stride X	5
Stride Y	5
Output No Feature Tiles	ONLY_TILES_WITH_FEATURES
Metadata Format	Classified_Tiles
Start Index	0
Class Value Field	class
Buffer Radius	0
Input Mask Polygons	
Rotation Angle	0
Reference System	MAP_SPACE
Processing Mode	PROCESS_AS_MOSAICKED_IMAGE
Blacken Around Feature	NO_BLACKEN
Crop Mode	FIXED_SIZE
Additional Input Raster	
Instance Feature Class	
Instance Class Value Field	
Minimum Polygon Overlap Ratio	0

Step 2: Training the Deep Learning Model

This is arguably the most crucial, as well as most time consuming step.

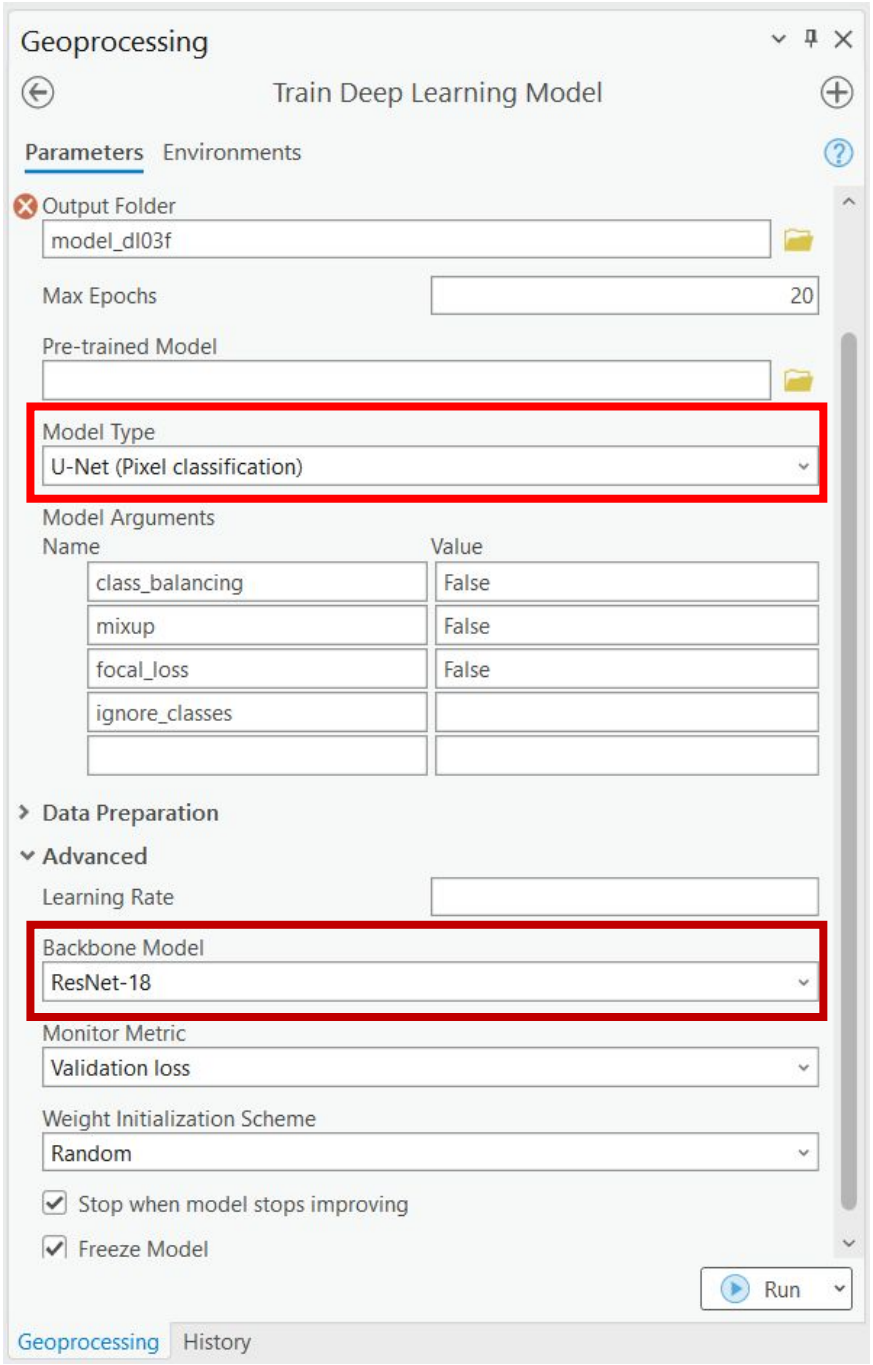

dl00c
Command Search (Alt+Q)
Not signed in
Project
Catalog
Insert
Analysis
View
Imagery
Share
Help
Manage
New Workspace
Georeference
Raster Functions
Function Editor
Change Detection
Classification Wizard
Classification Tools
Deep Learning Tools
Point
Distance
Area
Results
Process
Indices
Pixel Editor
Image Information
Create Hosted Imagery
Video Search
Ortho Mapping
Alignment
Analysis
Image Classification
Mensuration
Tools
Share
Motion Imagery
Contents
Project
Toolboxes
Databases
Styles
Folders
dl00c
_roi
dl_landsat
training_subset
x_output_classif
bz_landsat_2023_utm_100m.tif
bz_landsat_2023_utm_100m_rcl.tif
bz_landsat_2023_utm_500m.tif
bz_landsat_2023_utm_500m_rcl.tif
bz_landsat_2023_utm_1000m_rcl.tif
Projects
workspace
workspace2
Locators
Portal
Computer
Favorites
Catalog
Project
Folders
dl_landsat
Search dl_landsat
Name
Date Modified
Size
training_subset 8/28/2024 9:38:51 PM
x_output_classif 8/28/2024 9:32:19 PM
bz_landsat_2023_utm_100m.tif 8/19/2024 12:58:54 PM 1,335 KB
bz_landsat_2023_utm_100m_rcl.tif 8/22/2024 3:41:11 PM 9,312 KB
bz_landsat_2023_utm_500m.tif 8/21/2024 5:46:54 AM 1,697 KB
bz_landsat_2023_utm_500m_rcl.tif 8/22/2024 3:42:49 PM 2,864 KB
bz_landsat_2023_utm_1000m_rcl.tif 8/22/2024 7:06:12 PM 808 KB
bz_training_data_cl10.shp 8/19/2024 12:59:12 PM 8 KB
Select an item to view its metadata.
8 Items
0 Items Selected
Geoprocessing
Train Deep Learning Model
Parameters
Environments
Input Training Data
Output Folder
Max Epochs
20
Pre-trained Model
Model Type
Model Arguments
Name
Value
Data Preparation
Advanced
Run
Geoprocessing
Image Classification
History



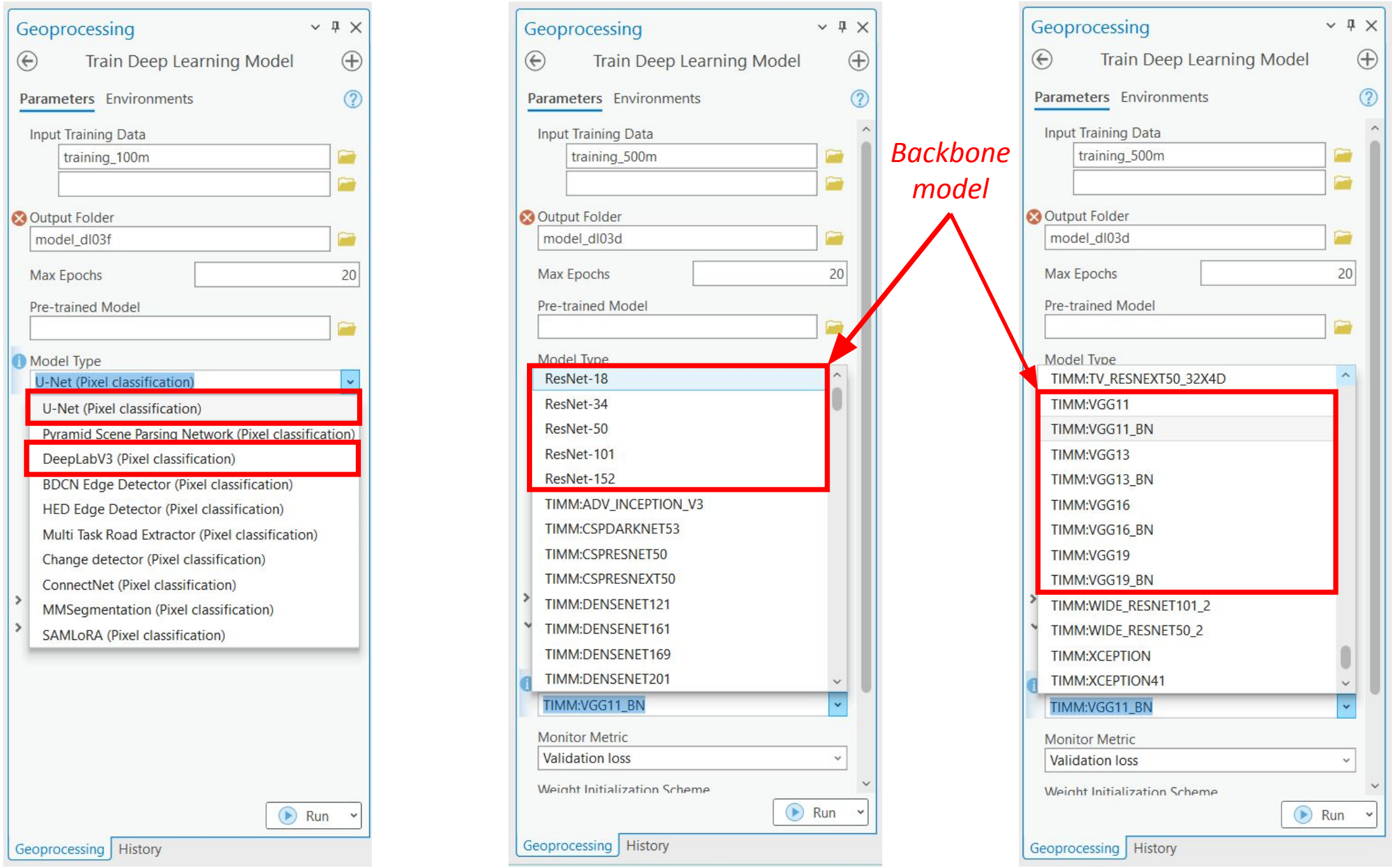
This input is from the previous step.

You just need to specify where the DL model will be saved.

From Esri: "The maximum number of epochs for which the model will be trained. A maximum epoch of one means the dataset will be passed forward and backward through the neural network one time." I recommend you test increasing this value (e.g., to 50), as this can theoretically help increase your model accuracy.



~~We will also~~ want to experiment with both of these.


Geoprocessing
Train Deep Learning Model
Parameters
Environments
Input Training Data
training_100m
Output Folder
model_dl03f
Max Epochs
20
Pre-trained Model
Model Type
U-Net (Pixel classification)
U-Net (Pixel classification)
Pyramid Scene Parsing Network (Pixel classification)
DeepLabV3 (Pixel classification)
BDCN Edge Detector (Pixel classification)
HED Edge Detector (Pixel classification)
Multi Task Road Extractor (Pixel classification)
Change detector (Pixel classification)
ConnectNet (Pixel classification)
MMSegmentation (Pixel classification)
SAMLoRA (Pixel classification)
Run
Geoprocessing
History
Geoprocessing
Train Deep Learning Model
Parameters
Environments
Input Training Data
training_500m
Output Folder
model_dl03d
Max Epochs
20
Pre-trained Model
Model Type
ResNet-18
ResNet-34
ResNet-50
ResNet-101
ResNet-152
TIMM:ADV_INCEPTION_V3
TIMM:CSPDARKNET53
TIMM:CSPRESNET50
TIMM:CSPRESNEXT50
TIMM:DENSENET121
TIMM:DENSENET161
TIMM:DENSENET169
TIMM:DENSENET201
TIMM:VGG11_BN
Monitor Metric
Validation loss
Weight Initialization Scheme
Run
Geoprocessing
History
Backbone model
Geoprocessing
Train Deep Learning Model
Parameters
Environments
Input Training Data
training_500m
Output Folder
model_dl03d
Max Epochs
20
Pre-trained Model
Model Type
TIMM:TV_RESNEXT50_32X4D
TIMM:VGG11
TIMM:VGG11_BN
TIMM:VGG13
TIMM:VGG13_BN
TIMM:VGG16
TIMM:VGG16_BN
TIMM:VGG19
TIMM:VGG19_BN
TIMM:WIDE_RESNET101_2
TIMM:WIDE_RESNET50_2
TIMM:XCEPTION
TIMM:XCEPTION41
TIMM:VGG11_BN
Monitor Metric
Validation loss
Weight Initialization Scheme
Run
Geoprocessing
History

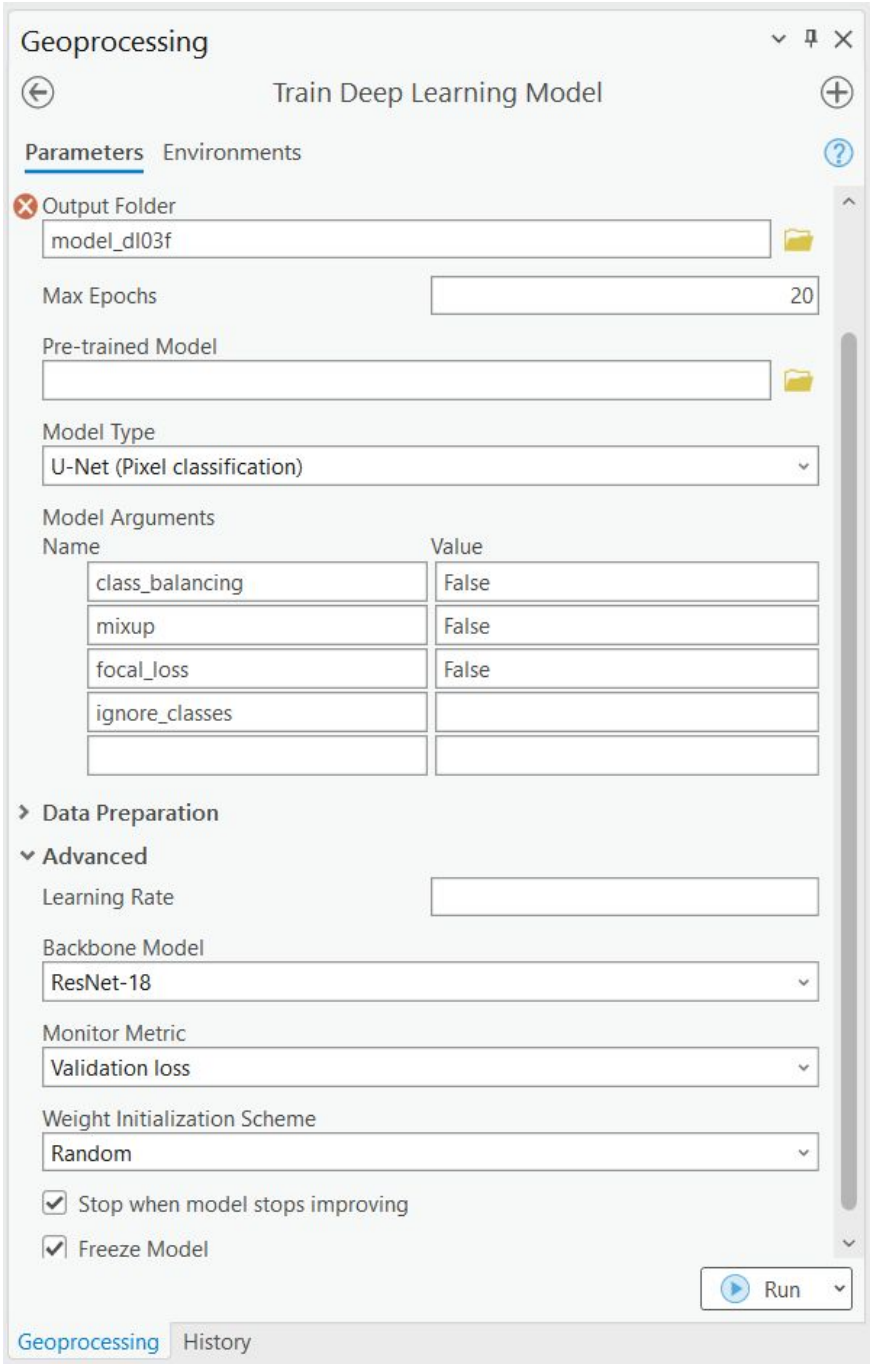
More info on backbone models, from Esri

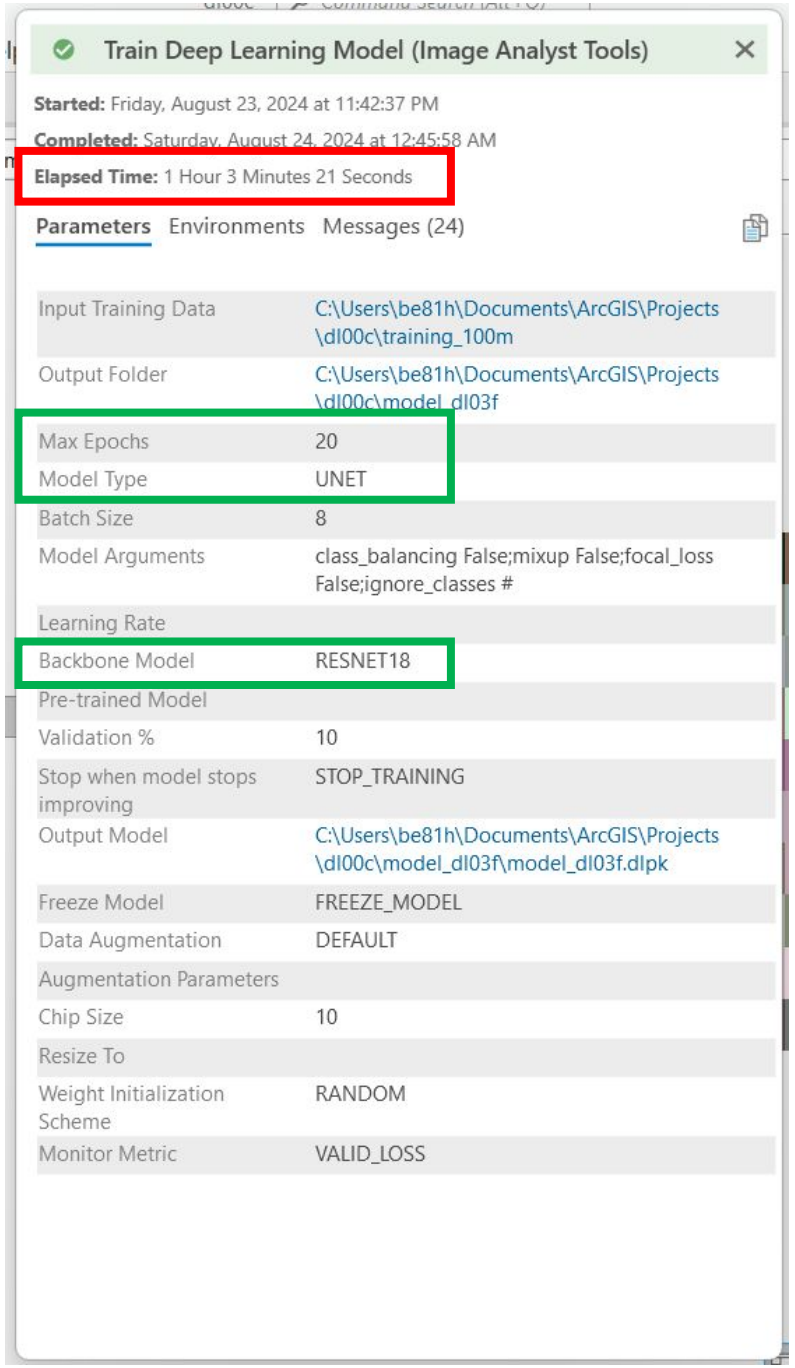
Specifies the preconfigured neural network that will be used as the architecture for training the new model. This method is known as Transfer Learning.

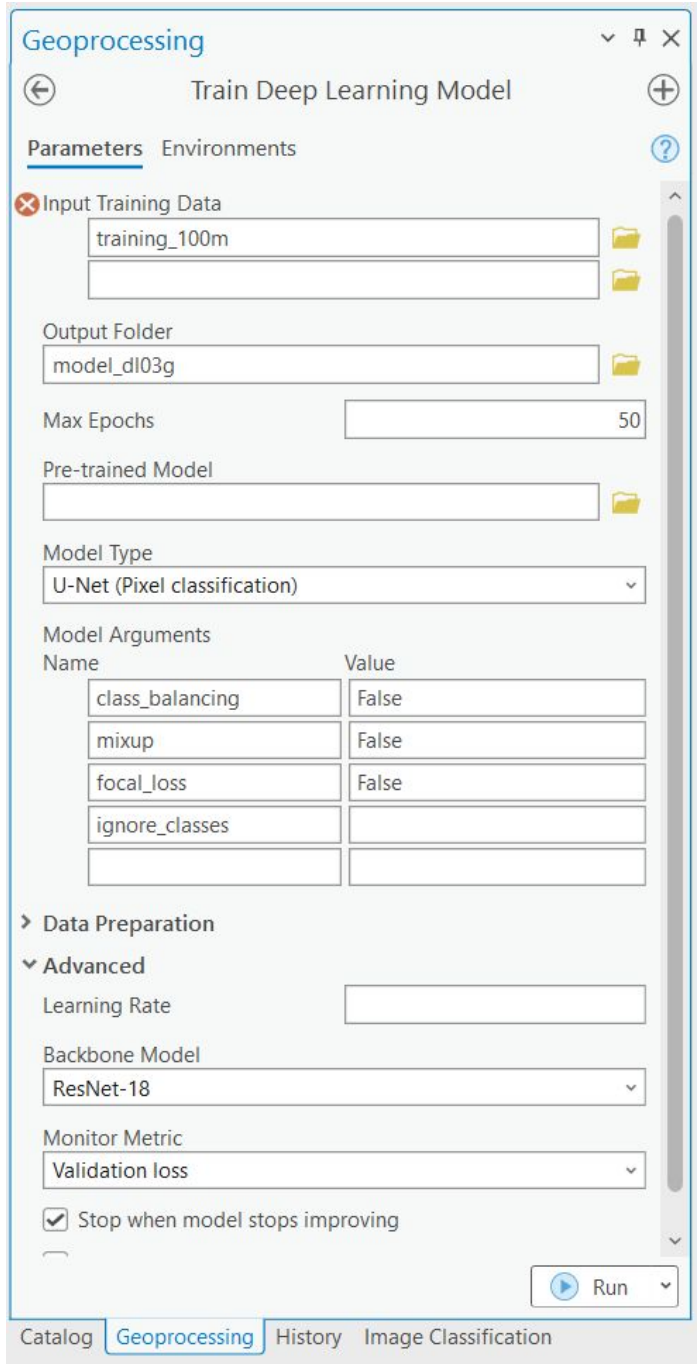
Additionally, supported convolution neural networks from the PyTorch Image Models (timm) can be specified using timm as a prefix, for example, timm:resnet31 , timm:inception_v4 , timm:efficientnet_b3, and so on.

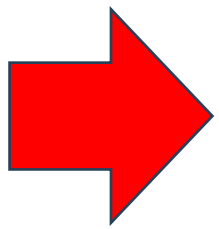
- DenseNet-121—The preconfigured model will be a dense network trained on the Imagenet Dataset that contains more than 1 million images and is 121 layers deep. Unlike ResNET, which combines the layer using summation, DenseNet combines the layers using concatenation.
- DenseNet-161—The preconfigured model will be a dense network trained on the Imagenet Dataset that contains more than 1 million images and is 161 layers deep. Unlike ResNET, which combines the layer using summation, DenseNet combines the layers using concatenation.
- DenseNet-169—The preconfigured model will be a dense network trained on the Imagenet Dataset that contains more than 1 million images and is 169 layers deep. Unlike ResNET, which combines the layer using summation, DenseNet combines the layers using concatenation.
- DenseNet-201—The preconfigured model will be a dense network trained on the Imagenet Dataset that contains more than 1 million images and is 201 layers deep. Unlike ResNET, which combines the layer using summation, DenseNet combines the layers using concatenation.
- MobileNet version 2—The preconfigured model will be trained on the Imagenet Database and is 54 layers deep and intended for Edge device computing, since it uses less memory.
- ResNet-18—The preconfigured model will be a residual network trained on the Imagenet Dataset that contains more than million images and is 18 layers deep.
- ResNet-34—The preconfigured model will be a residual network trained on the Imagenet Dataset that contains more than 1 million images and is 34 layers deep. This is the default.
- ResNet-50—The preconfigured model will be a residual network trained on the Imagenet Dataset that contains more than 1 million images and is 50 layers deep.
- ResNet-101—The preconfigured model will be a residual network trained on the Imagenet Dataset that contains more than 1 million images and is 101 layers deep.
- ResNet-152—The preconfigured model will be a residual network trained on the Imagenet Dataset that contains more than 1 million images and is 152 layers deep.
- VGG-11—The preconfigured model will be a convolution neural network trained on the Imagenet Dataset that contains more than 1 million images to classify images into 1,000 object categories and is 11 layers deep.
- VGG-11 with batch normalization—The preconfigured model will be based on the VGG network but with batch normalization, which means each layer in the network is normalized. It trained on the Imagenet dataset and has 11 layers.
- VGG-13—The preconfigured model will be a convolution neural network trained on the Imagenet Dataset that contains more than 1 million images to classify images into 1,000 object categories and is 13 layers deep.
- VGG-13 with batch normalization—The preconfigured model will be based on the VGG network but with batch normalization, which means each layer in the network is normalized. It trained on the Imagenet dataset and has 13 layers.
- VGG-16—The preconfigured model will be a convolution neural network trained on the Imagenet Dataset that contains more than 1 million images to classify images into 1,000 object categories and is 16 layers deep.
- VGG-16 with batch normalization—The preconfigured model will be based on the VGG network but with batch normalization, which means each layer in the network is normalized. It trained on the Imagenet dataset and has 16 layers.
- VGG-19—The preconfigured model will be a convolution neural network trained on the Imagenet Dataset that contains more than 1 million images to classify images into 1,000 object categories and is 19 layers deep.

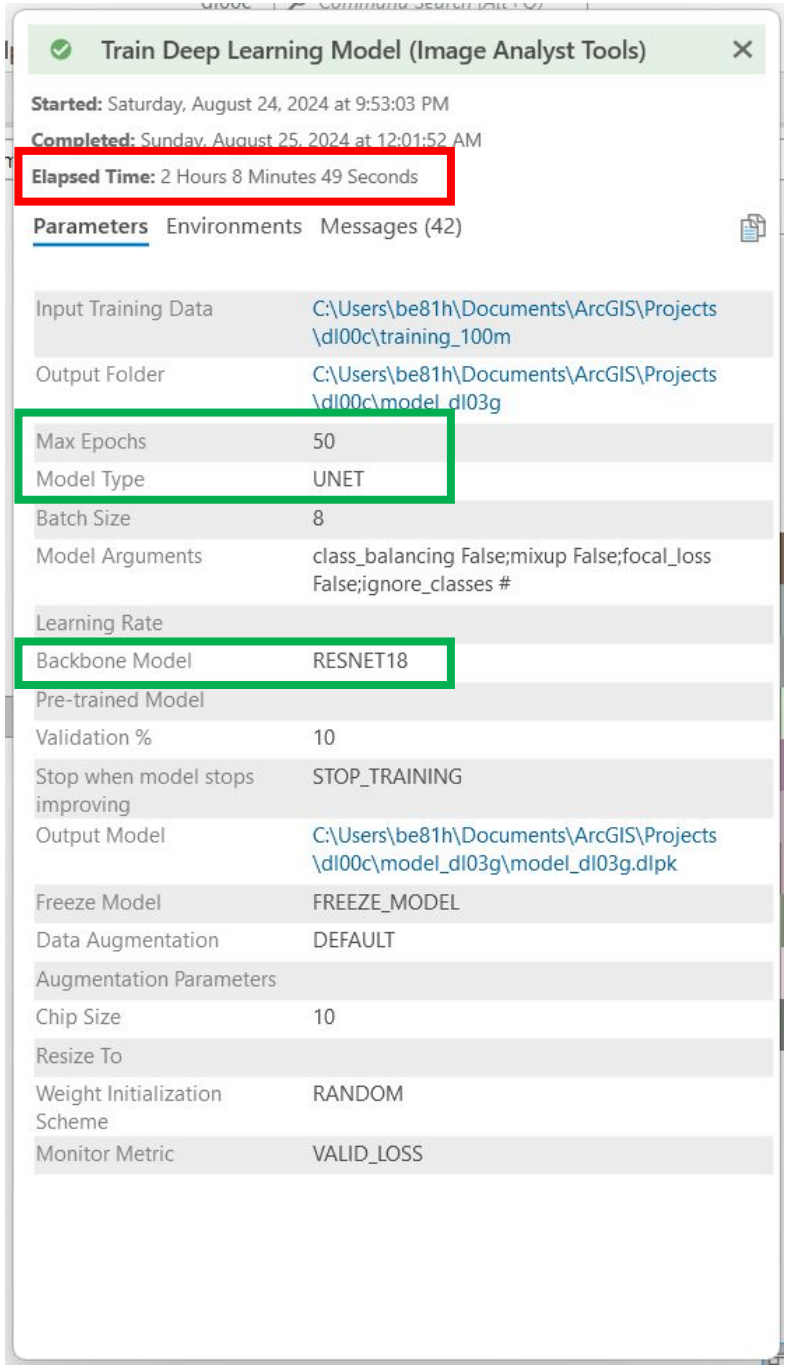
https://pro.arcgis.com/en/pro-app/latest/tool-reference/image-analyst/train-deep-learning-model.htm

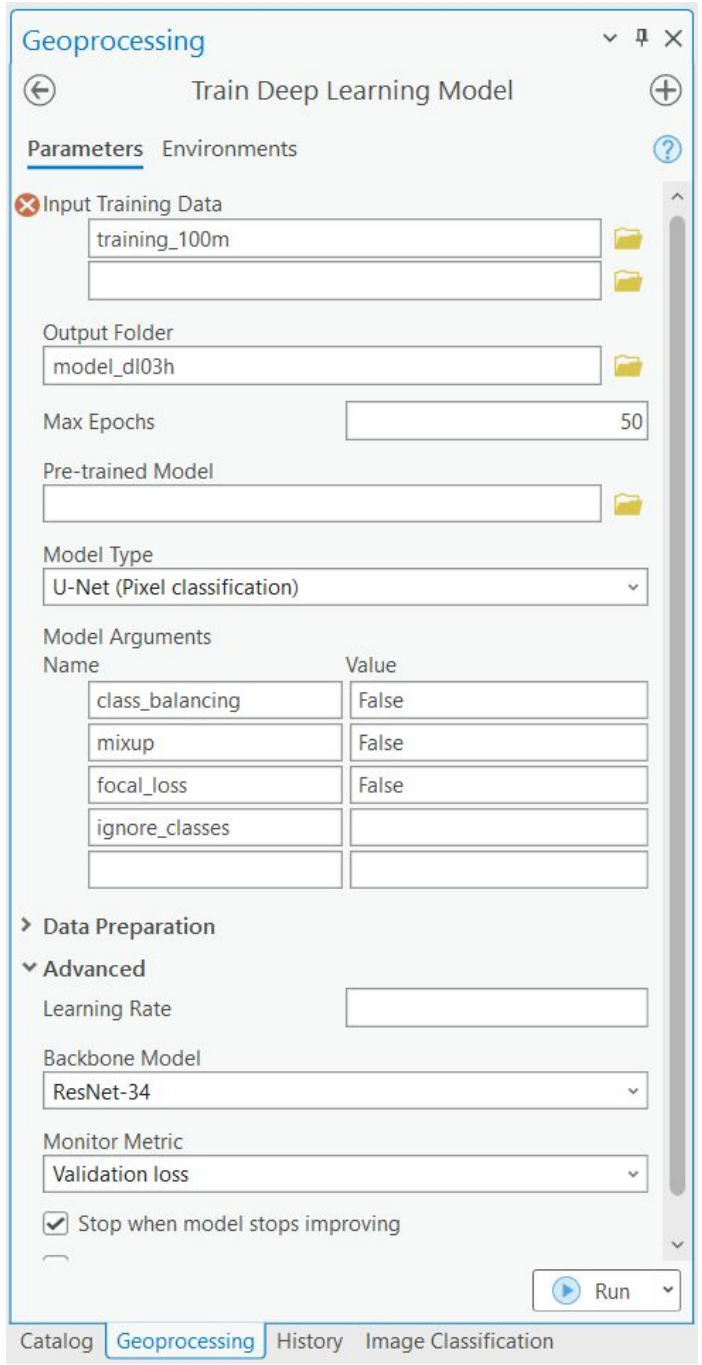

Geoprocessing
Train Deep Learning Model
Parameters
Environments
Output Folder
model_dl03f
Max Epochs
20
Pre-trained Model
Model Type
U-Net (Pixel classification)
Model Arguments
Name
Value
class_balancing
False
mixup
False
focal_loss
False
ignore_classes
Data Preparation
Advanced
Learning Rate
Backbone Model
ResNet-18
Monitor Metric
Validation loss
Weight Initialization Scheme
Random
Stop when model stops improving
Freeze Model
Run
Geoprocessing
History

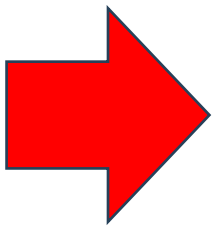

Train Deep Learning Model (Image Analyst Tools)
Started: Friday, August 23, 2024 at 11:42:37 PM
Completed: Saturday, August 24, 2024 at 12:45:58 AM
Elapsed Time: 1 Hour 3 Minutes 21 Seconds
Parameters
Environments
Messages (24)
Input Training Data
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m
Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03f
Max Epochs
20
Model Type
UNET
Batch Size
8
Model Arguments
class_balancing False;mixup False;focal_loss False;ignore_classes #
Learning Rate
Backbone Model
RESNET18
Pre-trained Model
Validation %
10
Stop when model stops improving
STOP_TRAINING
Output Model
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03f\model_dl03f.dlpk
Freeze Model
FREEZE_MODEL
Data Augmentation
DEFAULT
Augmentation Parameters
Chip Size
10
Resize To
Weight Initialization Scheme
RANDOM
Monitor Metric
VALID_LOSS

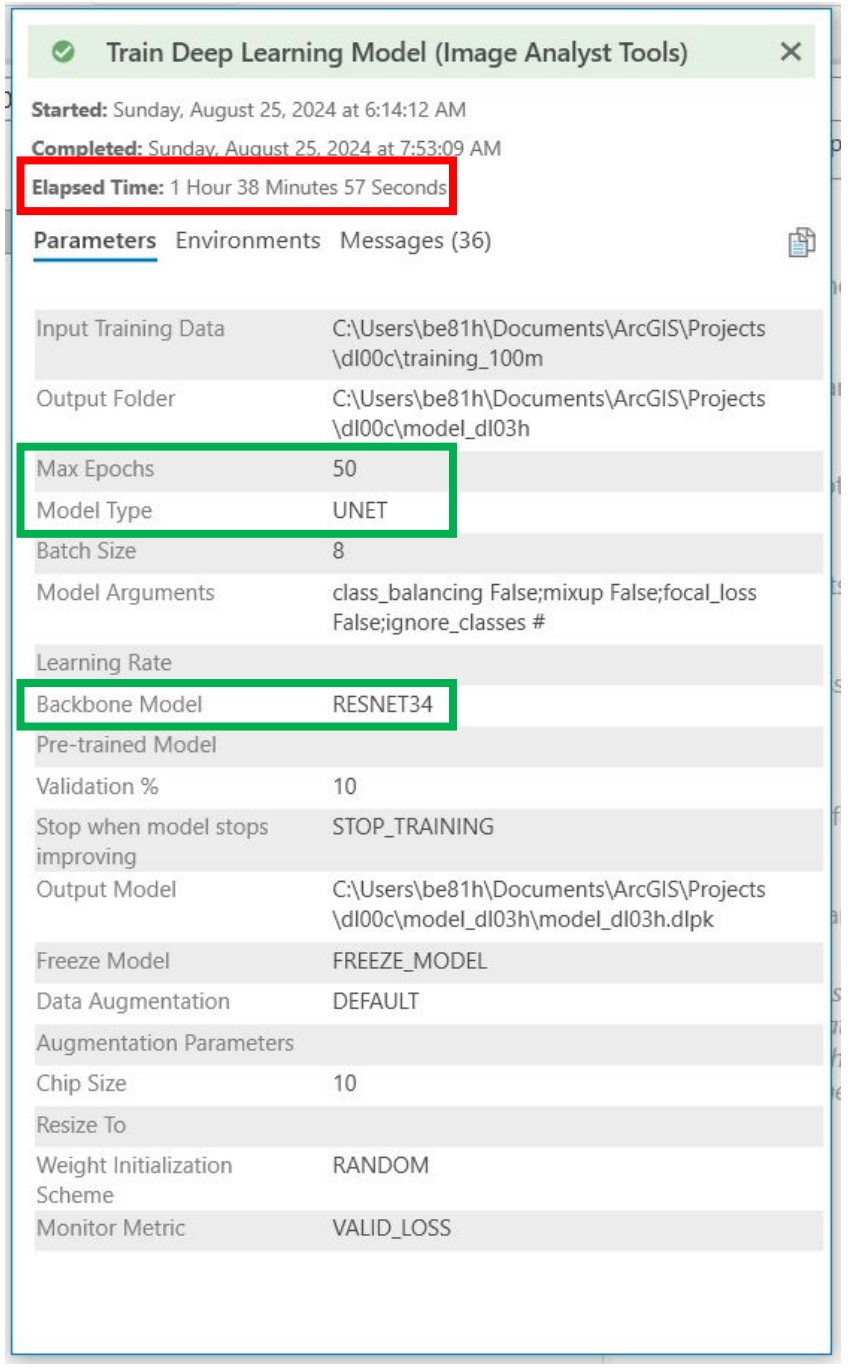

Geoprocessing
Train Deep Learning Model
Parameters
Environments
Input Training Data
training_100m
Output Folder
model_dl03g
Max Epochs
50
Pre-trained Model
Model Type
U-Net (Pixel classification)
Model Arguments
Name
Value
class_balancing
False
mixup
False
focal_loss
False
ignore_classes
Data Preparation
Advanced
Learning Rate
Backbone Model
ResNet-18
Monitor Metric
Validation loss
Stop when model stops improving
Run
Catalog
Geoprocessing
History
Image Classification

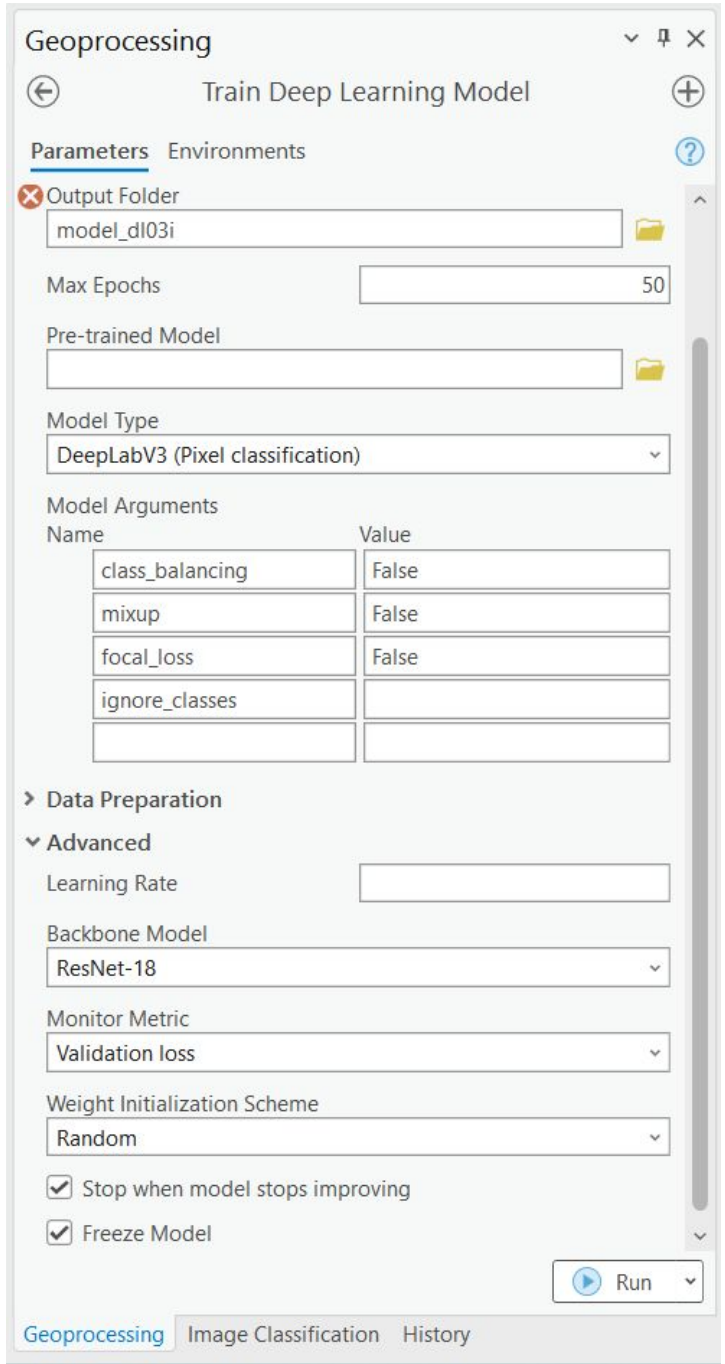


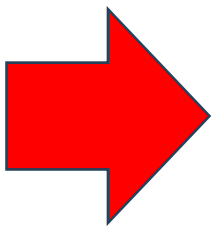

Train Deep Learning Model (Image Analyst Tools)
Started: Saturday, August 24, 2024 at 9:53:03 PM
Completed: Sunday, August 25, 2024 at 12:01:52 AM
Elapsed Time: 2 Hours 8 Minutes 49 Seconds
Parameters
Environments
Messages (42)
Input Training Data
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m
Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03g
Max Epochs
50
Model Type
UNET
Batch Size
8
Model Arguments
class_balancing False;mixup False;focal_loss False;ignore_classes #
Learning Rate
Backbone Model
RESNET18
Pre-trained Model
Validation %
10
Stop when model stops improving
STOP_TRAINING
Output Model
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03g\model_dl03g.dlpk
Freeze Model
FREEZE_MODEL
Data Augmentation
DEFAULT
Augmentation Parameters
Chip Size
10
Resize To
Weight Initialization Scheme
RANDOM
Monitor Metric
VALID_LOSS

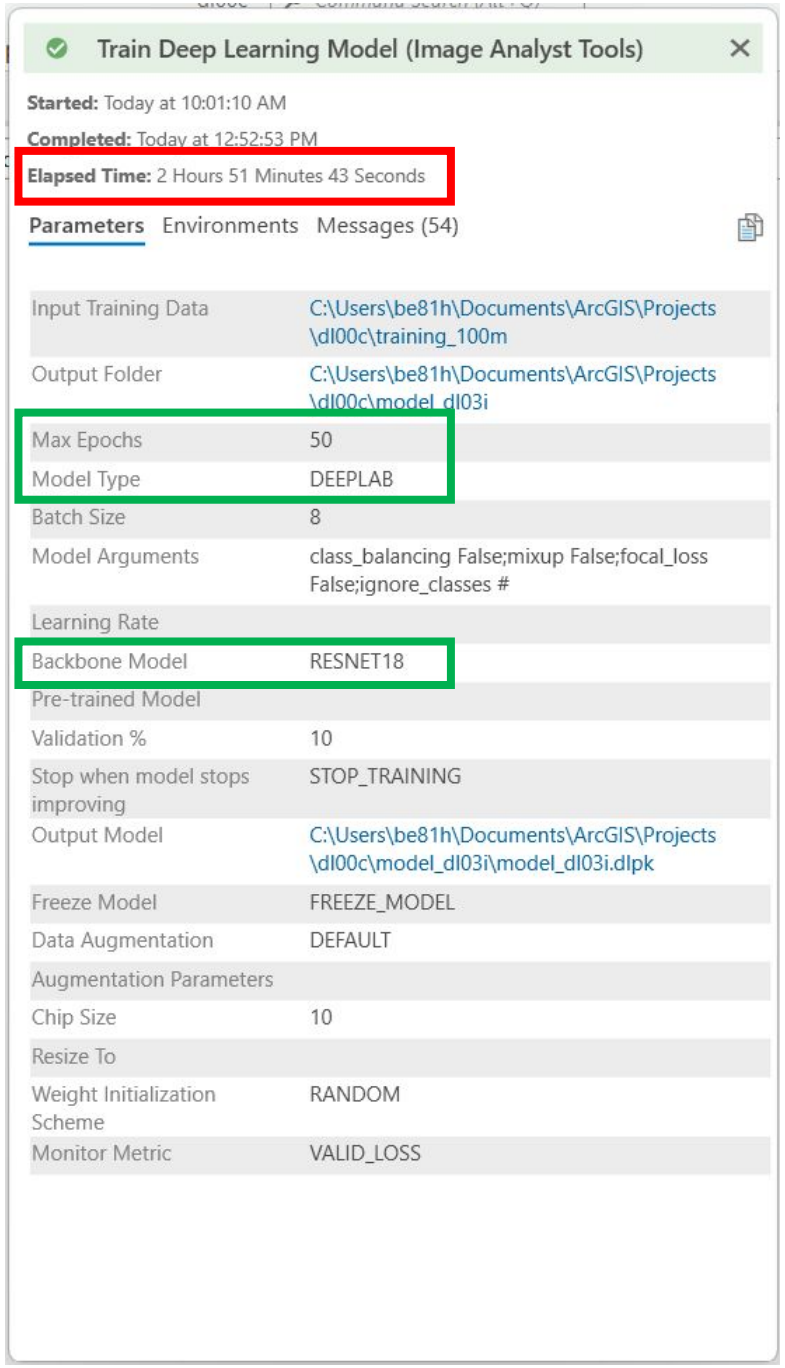

Geoprocessing
Train Deep Learning Model
Parameters
Environments
Input Training Data
training_100m
Output Folder
model_dl03h
Max Epochs
50
Pre-trained Model
Model Type
U-Net (Pixel classification)
Model Arguments
Name
Value
class_balancing
False
mixup
False
focal_loss
False
ignore_classes
Data Preparation
Advanced
Learning Rate
Backbone Model
ResNet-34
Monitor Metric
Validation loss
Stop when model stops improving
Run
Catalog
Geoprocessing
History
Image Classification

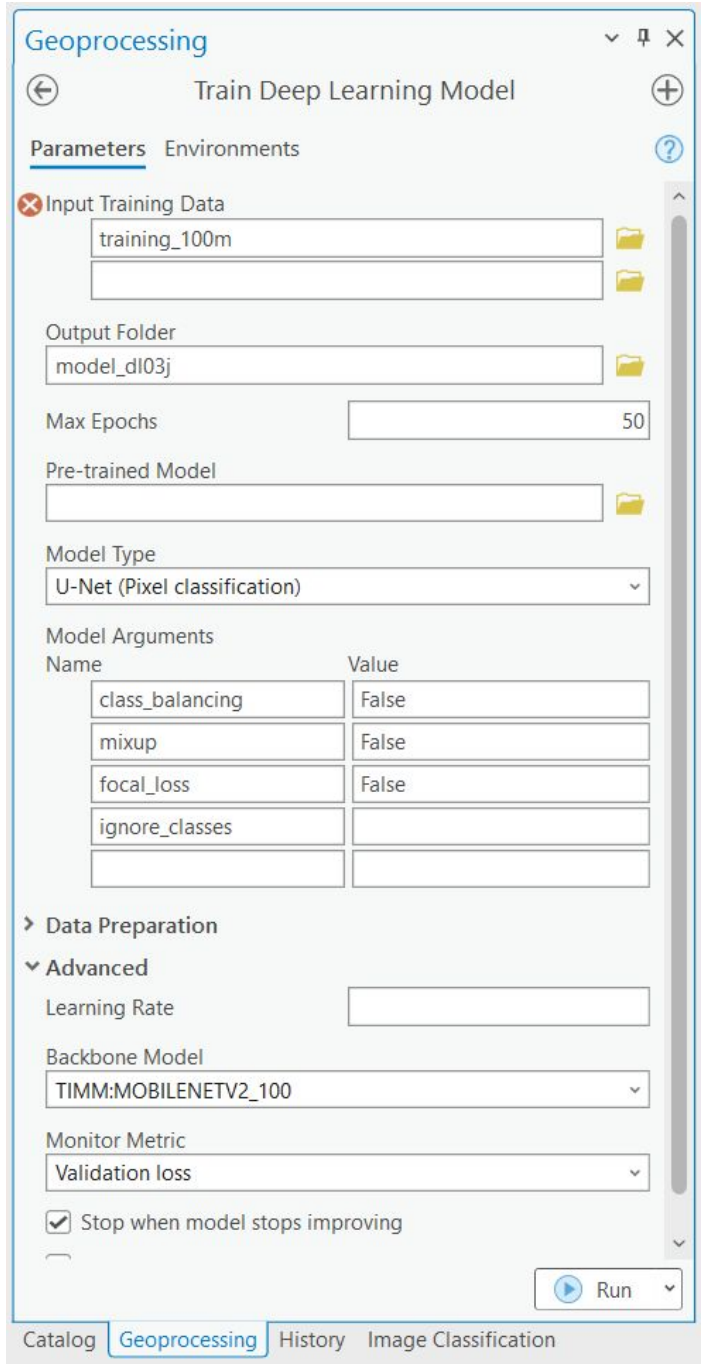


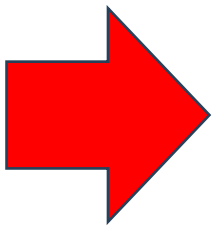

Train Deep Learning Model (Image Analyst Tools)
Started: Sunday, August 25, 2024 at 6:14:12 AM
Completed: Sunday, August 25, 2024 at 7:53:09 AM
Elapsed Time: 1 Hour 38 Minutes 57 Seconds
Parameters
Environments
Messages (36)
Input Training Data
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m
Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03h
Max Epochs
50
Model Type
UNET
Batch Size
8
Model Arguments
class_balancing False;mixup False;focal_loss False;ignore_classes #
Learning Rate
Backbone Model
RESNET34
Pre-trained Model
Validation %
10
Stop when model stops improving
STOP_TRAINING
Output Model
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03h\model_dl03h.dlpk
Freeze Model
FREEZE_MODEL
Data Augmentation
DEFAULT
Augmentation Parameters
Chip Size
10
Resize To
Weight Initialization Scheme
RANDOM
Monitor Metric
VALID_LOSS

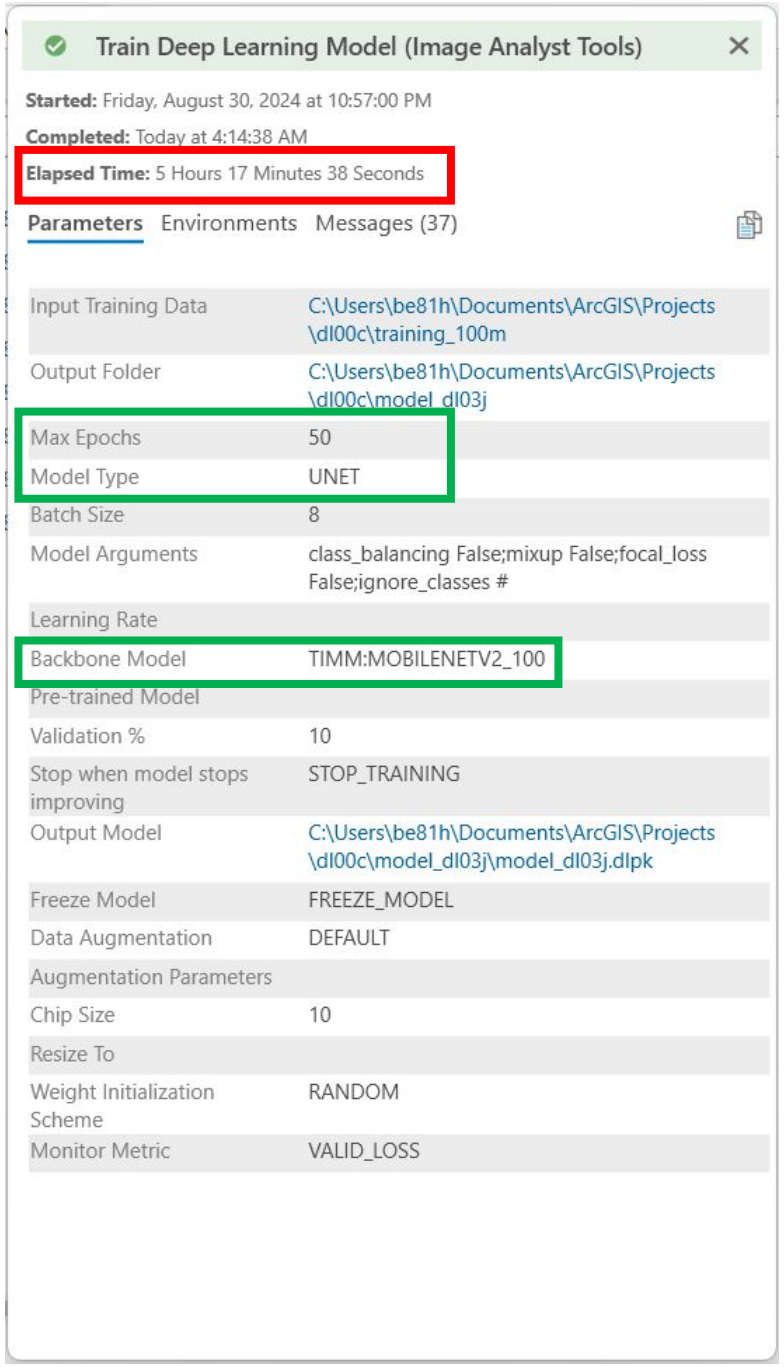

Geoprocessing
Train Deep Learning Model
Parameters
Environments
Output Folder
model_dl03i
Max Epochs
50
Pre-trained Model
Model Type
DeepLabV3 (Pixel classification)
Model Arguments
Name
Value
class_balancing
False
mixup
False
focal_loss
False
ignore_classes
Data Preparation
Advanced
Learning Rate
Backbone Model
ResNet-18
Monitor Metric
Validation loss
Weight Initialization Scheme
Random
Stop when model stops improving
Freeze Model
Run
Geoprocessing
Image Classification
History

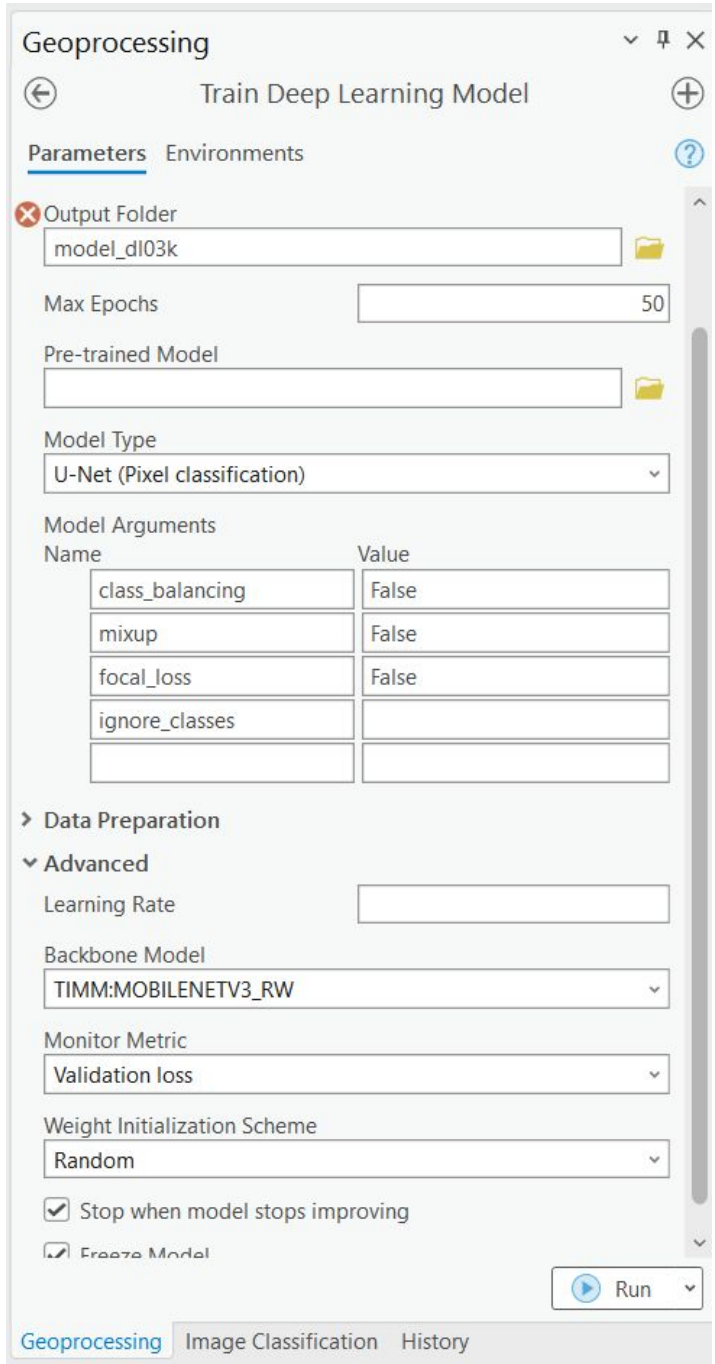


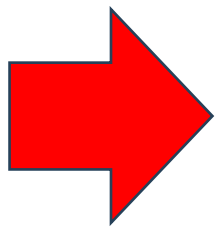

Train Deep Learning Model (Image Analyst Tools)
Started: Today at 10:01:10 AM
Completed: Today at 12:52:53 PM
Elapsed Time: 2 Hours 51 Minutes 43 Seconds
Parameters
Environments
Messages (54)
Input Training Data
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m
Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03i
Max Epochs
50
Model Type
DEEPLAB
Batch Size
8
Model Arguments
class_balancing False;mixup False;focal_loss False;ignore_classes #
Learning Rate
Backbone Model
RESNET18
Pre-trained Model
Validation %
10
Stop when model stops improving
STOP_TRAINING
Output Model
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03i\model_dl03i.dlpk
Freeze Model
FREEZE_MODEL
Data Augmentation
DEFAULT
Augmentation Parameters
Chip Size
10
Resize To
Weight Initialization Scheme
RANDOM
Monitor Metric
VALID_LOSS

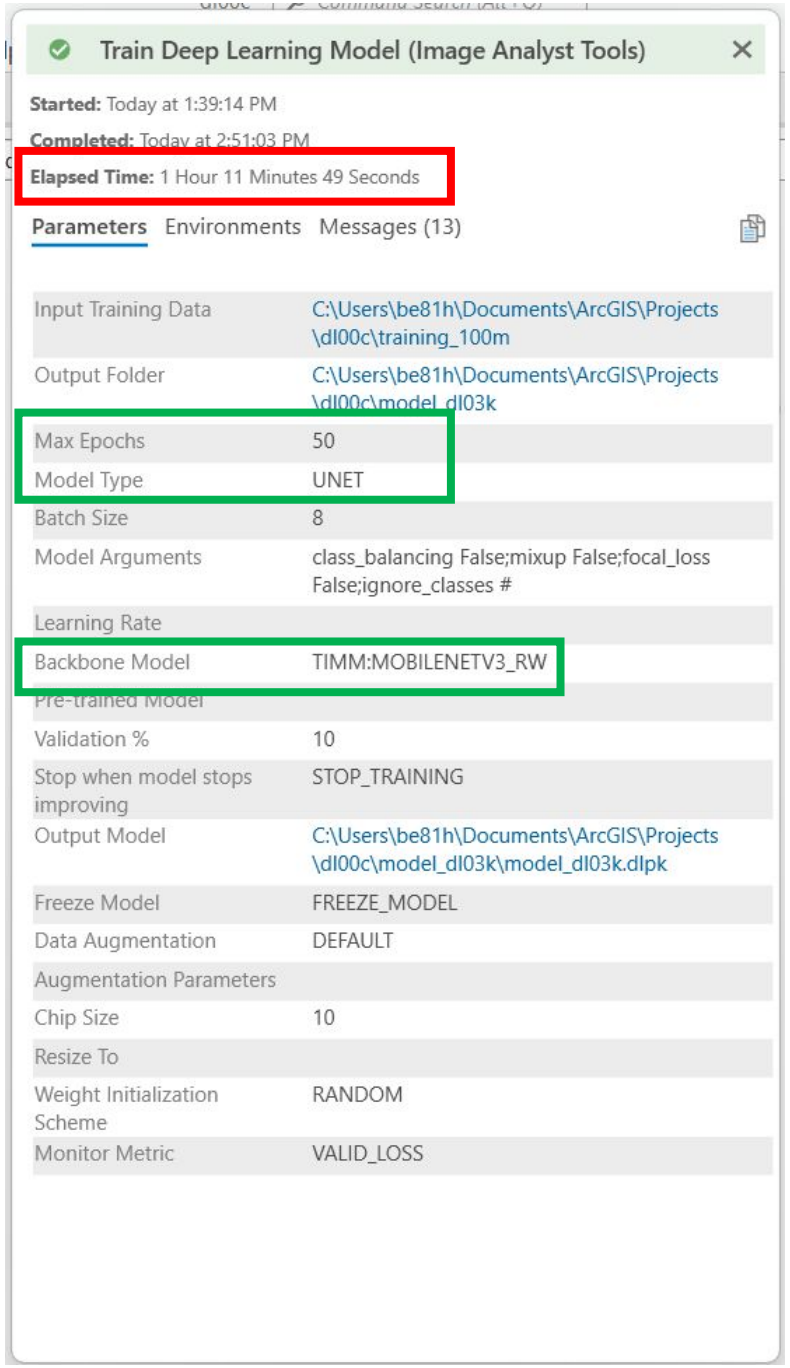

Geoprocessing
Train Deep Learning Model
Parameters
Environments
Input Training Data
training_100m
Output Folder
model_dl03j
Max Epochs
50
Pre-trained Model
Model Type
U-Net (Pixel classification)
Model Arguments
Name
Value
class_balancing
False
mixup
False
focal_loss
False
ignore_classes
Data Preparation
Advanced
Learning Rate
Backbone Model
TIMM:MOBILENETV2_100
Monitor Metric
Validation loss
Stop when model stops improving
Run
Catalog
Geoprocessing
History
Image Classification



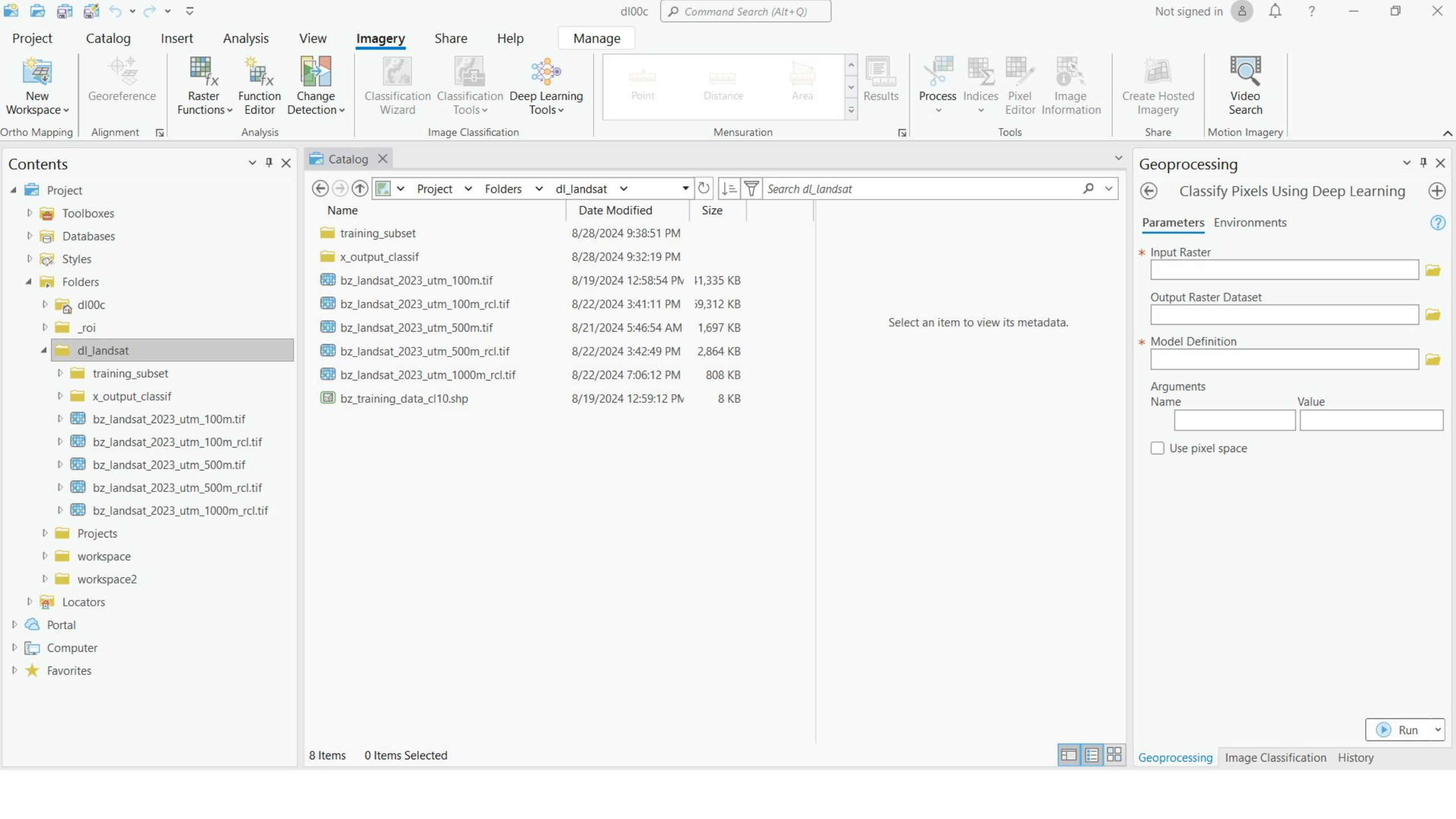

Train Deep Learning Model (Image Analyst Tools)
Started: Friday, August 30, 2024 at 10:57:00 PM
Completed: Today at 4:14:38 AM
Elapsed Time: 5 Hours 17 Minutes 38 Seconds
Parameters
Environments
Messages (37)
Input Training Data
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m
Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03j
Max Epochs
50
Model Type
UNET
Batch Size
8
Model Arguments
class_balancing False;mixup False;focal_loss False;ignore_classes #
Learning Rate
Backbone Model
TIMM:MOBILENETV2_100
Pre-trained Model
Validation %
10
Stop when model stops improving
STOP_TRAINING
Output Model
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03j\model_dl03j.dlpk
Freeze Model
FREEZE_MODEL
Data Augmentation
DEFAULT
Augmentation Parameters
Chip Size
10
Resize To
Weight Initialization Scheme
RANDOM
Monitor Metric
VALID_LOSS

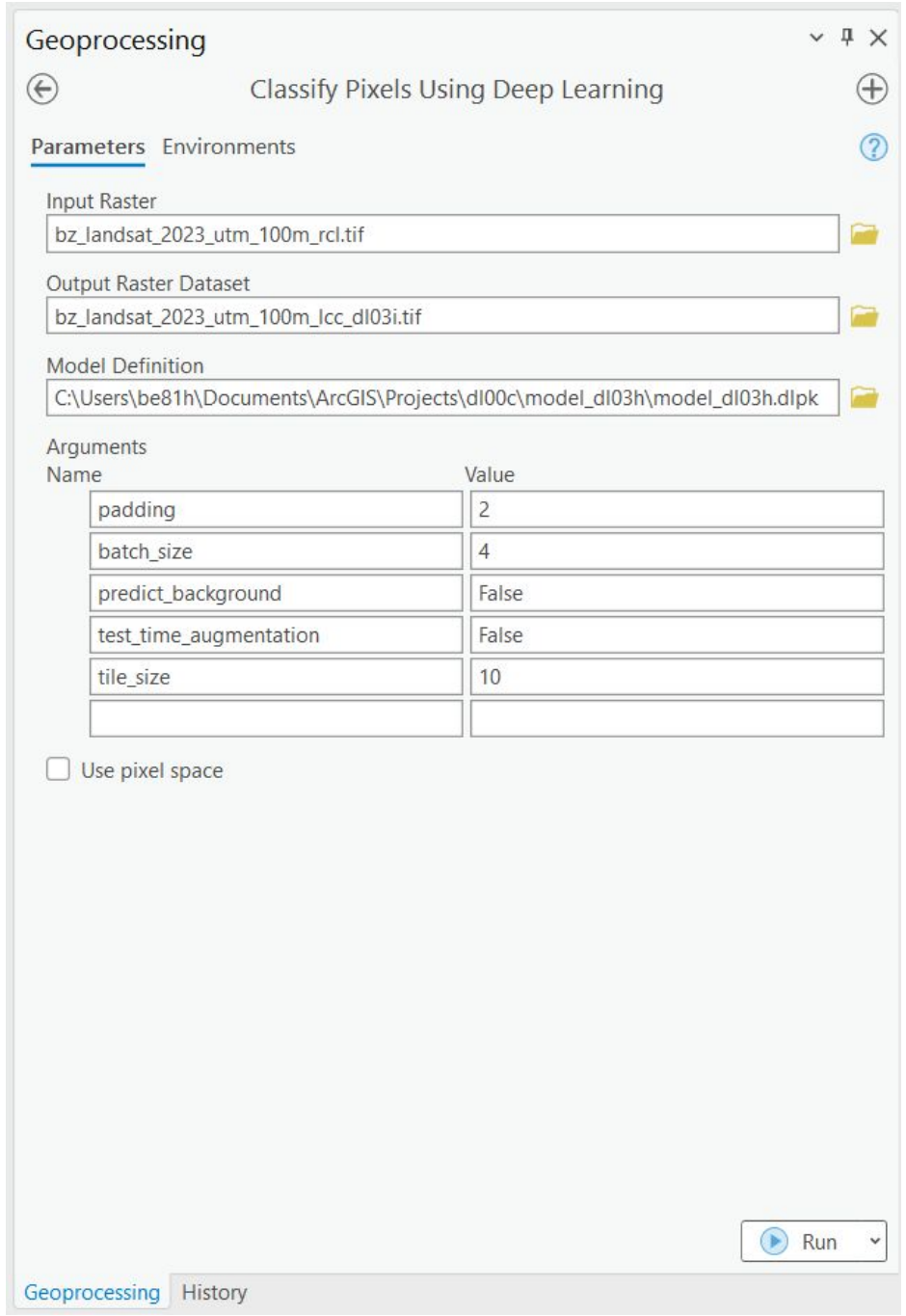

Geoprocessing
Train Deep Learning Model
Parameters
Environments
Output Folder
model_dl03k
Max Epochs
50
Pre-trained Model
Model Type
U-Net (Pixel classification)
Model Arguments
Name
Value
class_balancing
False
mixup
False
focal_loss
False
ignore_classes
Data Preparation
Advanced
Learning Rate
Backbone Model
TIMM:MOBILENETV3_RW
Monitor Metric
Validation loss
Weight Initialization Scheme
Random
Stop when model stops improving
Run
Geoprocessing
Image Classification
History




Train Deep Learning Model (Image Analyst Tools)
Started: Today at 1:39:14 PM
Completed: Today at 2:51:03 PM
Elapsed Time: 1 Hour 11 Minutes 49 Seconds
Parameters
Environments
Messages (13)
Input Training Data
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m
Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03k
Max Epochs
50
Model Type
UNET
Batch Size
8
Model Arguments
class_balancing False;mixup False;focal_loss False;ignore_classes #
Learning Rate
Backbone Model
TIMM:MOBILENETV3_RW
Pre-trained Model
Validation %
10
Stop when model stops improving
STOP_TRAINING
Output Model
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03k\model_dl03k.dlpk
Freeze Model
FREEZE_MODEL
Data Augmentation
DEFAULT
Augmentation Parameters
Chip Size
10
Resize To
Weight Initialization Scheme
RANDOM
Monitor Metric
VALID_LOSS

Step 3: Pixel Classification ('Model Inferencing')


dl00c
Command Search (Alt+Q)
Not signed in
Project
Catalog
Insert
Analysis
View
Imagery
Share
Help
Manage
New Workspace
Georeference
Raster Functions
Function Editor
Change Detection
Classification Wizard
Classification Tools
Deep Learning Tools
Point
Distance
Area
Results
Process
Indices
Pixel Editor
Image Information
Create Hosted Imagery
Video Search
Ortho Mapping
Alignment
Analysis
Image Classification
Mensuration
Tools
Share
Motion Imagery
Contents
Project
Toolboxes
Databases
Styles
Folders
dl00c
_roi
dl_landsat
training_subset
x_output_classif
bz_landsat_2023_utm_100m.tif
bz_landsat_2023_utm_100m_rcl.tif
bz_landsat_2023_utm_500m.tif
bz_landsat_2023_utm_500m_rcl.tif
bz_landsat_2023_utm_1000m_rcl.tif
Projects
workspace
workspace2
Locators
Portal
Computer
Favorites
Catalog
Project
Folders
dl_landsat
Search dl_landsat
Name
Date Modified
Size
training_subset 8/28/2024 9:38:51 PM
x_output_classif 8/28/2024 9:32:19 PM
bz_landsat_2023_utm_100m.tif 8/19/2024 12:58:54 PM 11,335 KB
bz_landsat_2023_utm_100m_rcl.tif 8/22/2024 3:41:11 PM 39,312 KB
bz_landsat_2023_utm_500m.tif 8/21/2024 5:46:54 AM 1,697 KB
bz_landsat_2023_utm_500m_rcl.tif 8/22/2024 3:42:49 PM 2,864 KB
bz_landsat_2023_utm_1000m_rcl.tif 8/22/2024 7:06:12 PM 808 KB
bz_training_data_cl10.shp 8/19/2024 12:59:12 PM 8 KB
Select an item to view its metadata.
8 Items
0 Items Selected
Geoprocessing
Classify Pixels Using Deep Learning
Parameters
Environments
Input Raster
Output Raster Dataset
Model Definition
Arguments
Name
Value
Use pixel space
Run
Geoprocessing
Image Classification
History


Geoprocessing
Classify Pixels Using Deep Learning
Parameters
Environments
Input Raster
bz_landsat_2023_utm_100m_rcl.tif
Output Raster Dataset
bz_landsat_2023_utm_100m_lcc_dl03i.tif
Model Definition
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03h\model_dl03h.dlpk
Arguments
Name
Value
padding
2
batch_size
4
predict_background
False
test_time_augmentation
False
tile_size
10
Use pixel space
Run
Geoprocessing
History


Classify Pixels Using Deep Learning
(Image Analyst Tools)
Started: Monday, August 26, 2024 at 1:44:21 PM
Completed: Monday, August 26, 2024 at 2:14:08 PM
Elapsed Time: 29 Minutes 47 Seconds
Parameters
Environments
Messages (2)
Input Raster
C:\gis\dl_landsat\bz_landsat_2023_utm_100m_rcl.tif
Output Raster Dataset
C:\gis\dl_landsat\x_output_classif\bz_landsat_2023_utm_100m_lcc_dl03g.tif
Model Definition
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\model_dl03g\model_dl03g.dlpk
Arguments
padding 2;batch_size 4;predict_background False;test_time_augmentation False;tile_size 10
Processing Mode
PROCESS_AS_MOSAICKED_IMAGE
Output Folder
Output Features
Overwrite attachments
NO_OVERWRITE
Use pixel space
NO_PIXELSPACE

Step 4: Reviewing the model performance

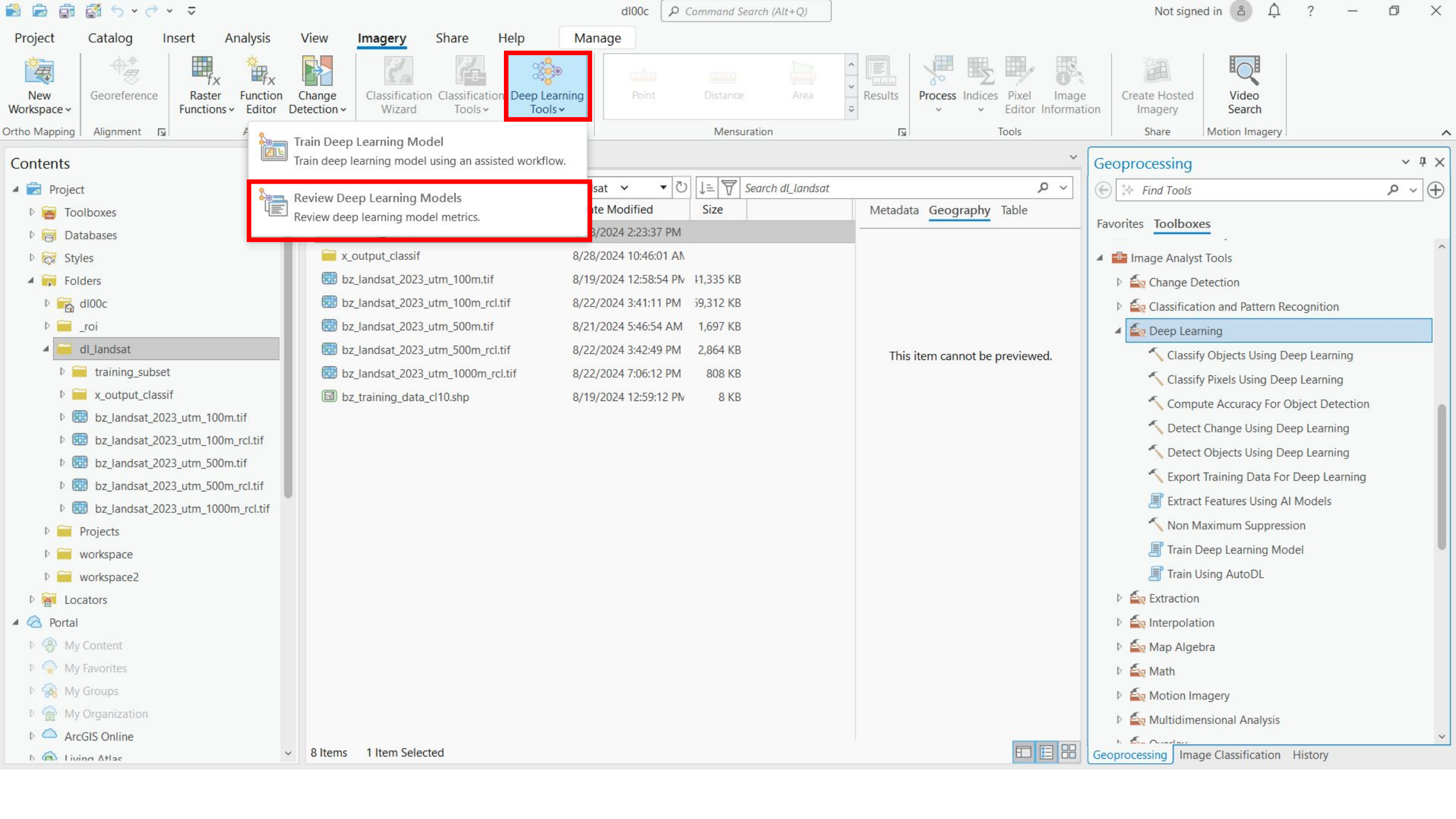

dl00c
Command Search (Alt+Q)
Not signed in
Project
Catalog
Insert
Analysis
View
Imagery
Share
Help
Manage
New Workspace
Georeference
Raster Functions
Function Editor
Change Detection
Classification Wizard
Classification Tools
Deep Learning Tools
Point
Distance
Area
Results
Process
Indices
Pixel Editor
Image Information
Create Hosted Imagery
Video Search
Ortho Mapping
Alignment
Mensuration
Tools
Share
Motion Imagery
Train Deep Learning Model
Train deep learning model using an assisted workflow.
Review Deep Learning Models
Review deep learning model metrics.
Contents
Project
Toolboxes
Databases
Styles
Folders
dl00c
_roi
dl_landsat
training_subset
x_output_classif
bz_landsat_2023_utm_100m.tif
bz_landsat_2023_utm_100m_rcl.tif
bz_landsat_2023_utm_500m.tif
bz_landsat_2023_utm_500m_rcl.tif
bz_landsat_2023_utm_1000m_rcl.tif
Projects
workspace
workspace2
Locators
Portal
My Content
My Favorites
My Groups
My Organization
ArcGIS Online
Search dl_landsat
Size
8/28/2024 10:46:01 AM
x_output_classif
8/19/2024 12:58:54 PM 1,335 KB
bz_landsat_2023_utm_100m.tif
8/22/2024 3:41:11 PM 9,312 KB
bz_landsat_2023_utm_100m_rcl.tif
8/21/2024 5:46:54 AM 1,697 KB
bz_landsat_2023_utm_500m.tif
8/22/2024 3:42:49 PM 2,864 KB
bz_landsat_2023_utm_500m_rcl.tif
8/22/2024 7:06:12 PM 808 KB
bz_landsat_2023_utm_1000m_rcl.tif
8/19/2024 12:59:12 PM 8 KB
bz_training_data_cl10.shp
Metadata
Geography
Table
This item cannot be previewed.
8 Items
1 Item Selected
Geoprocessing
Find Tools
Favorites
Toolboxes
Image Analyst Tools
Change Detection
Classification and Pattern Recognition
Deep Learning
Classify Objects Using Deep Learning
Classify Pixels Using Deep Learning
Compute Accuracy For Object Detection
Detect Change Using Deep Learning
Detect Objects Using Deep Learning
Export Training Data For Deep Learning
Extract Features Using AI Models
Non Maximum Suppression
Train Deep Learning Model
Train Using AutoDL
Extraction
Interpolation
Map Algebra
Math
Motion Imagery
Multidimensional Analysis
Geoprocessing
Image Classification
History

Image Classification

Deep Learning Model Reviewer

Model

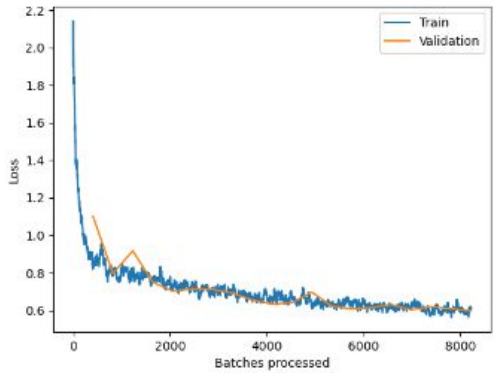
model_dl03f

Model Type: UnetClassifier

Backbone: resnet18

Learning Rate: slice('5.7544e-05', '5.7544e-04', None)

Training and Validation Loss



Analysis of the Model

Accuracy: 7.4499e-01

Epochs Details

Epoch	Traini	Valid	Time	Other Metrics
1	0.82	1.10	00:0	"accuracy": 0.686164 "Dice": 0.632022
2	0.79	0.79	00:0	"accuracy": 0.696137

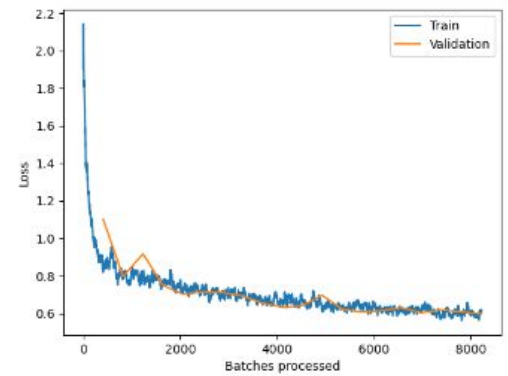
Geoprocessing | Image Classification | History

Starting accuracy: 68.6%

Final accuracy: 74.5%

Image Classification

Deep Learning Model Reviewer



Analysis of the Model

Accuracy: 7.4499e-01

Epochs Details

Epoch	Traini	Valid	Time	Other Metrics
17	0.60	0.60	00:0	"accuracy": 0.743095 "Dice": 0.661550
18	0.57	0.61	00:0	"accuracy": 0.741232 "Dice": 0.660554
19	0.57	0.60	00:0	"accuracy": 0.744109 "Dice": 0.661157
20	0.61	0.59	00:0	"accuracy": 0.744986 "Dice": 0.663018

Geoprocessing | Image Classification | History

Image Classification

Deep Learning Model Reviewer

Model

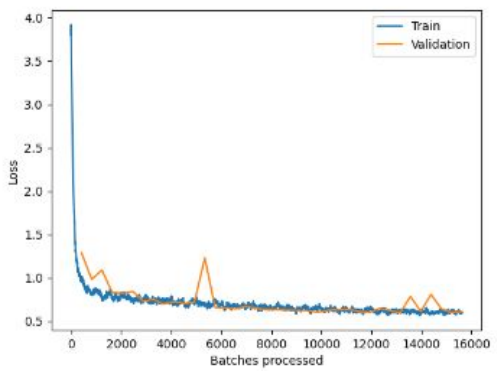
model_dl03g

Model Type: UnetClassifier

Backbone: resnet18

Learning Rate: slice('3.3113e-05', '3.3113e-04', None)

Training and Validation Loss



Analysis of the Model

Accuracy: 7.4318e-01

Epochs Details

Epoch	Traini	Valida	Time	Other Metrics
1	0.95	1.28	00:0	"accuracy": 0.662739 "Dice": 0.631559
2	0.80	0.98	00:0	"accuracy": 0.691342 "Dice": 0.631683

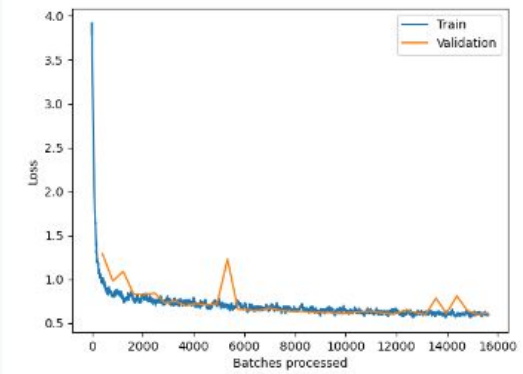
Catalog Geoprocessing History Image Classification

Starting accuracy: 66.3%

Final accuracy: 74.5%

Image Classification

Deep Learning Model Reviewer



Analysis of the Model

Accuracy: 7.4318e-01

Epochs Details

Epoch	Traini	Valida	Time	Other Metrics
35	0.58	0.80	00:0	"accuracy": 0.739890 "Dice": 0.663372
36	0.62	0.64	00:0	"accuracy": 0.744876 "Dice": 0.663784
37	0.61	0.59	00:0	"accuracy": 0.745643 "Dice": 0.657168
38	0.59	0.61	00:0	"accuracy": 0.745068 "Dice": 0.656206

Geoprocessing Image Classification History

Image Classification

Deep Learning Model Reviewer

Model

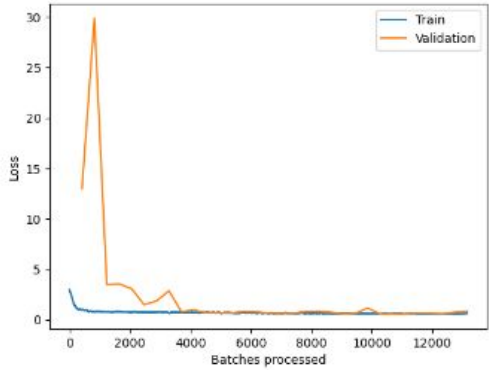
model_dl03h

Model Type: UnetClassifier

Backbone: resnet34

Learning Rate: slice('3.3113e-05', '3.3113e-04', None)

Training and Validation Loss



Analysis of the Model

Accuracy: 7.4005e-01

Epochs Details

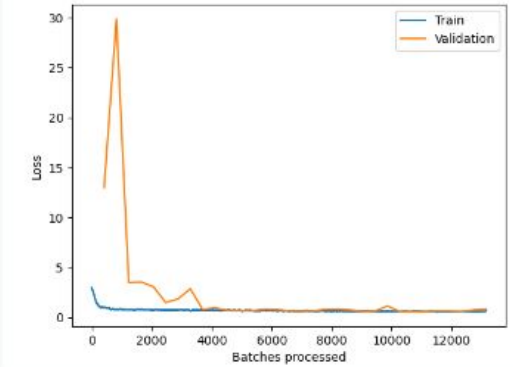
Epoch	Train	Valid	Time	Other Metrics
1	0.9(	13.(	00:(	"accuracy": 0.626876 "Dice": 0.631477
2	0.8	29.(	00:(	"accuracy": 0.636410

Starting accuracy: 62.3%

Geoprocessing | Image Classification | History

Image Classification

Deep Learning Model Reviewer



Analysis of the Model

Accuracy: 7.4005e-01

Epochs Details

Epoch	Train	Valid	Time	Other Metrics
29	0.59(	0.67(	00:0	"accuracy": 0.726328 "Dice": 0.644962
30	0.60	0.62(	00:0	"accuracy": 0.739643 "Dice": 0.669203
31	0.58	0.75(	00:0	"accuracy": 0.728767 "Dice": 0.660330
32	0.64	0.87	00:0	"accuracy": 0.712246 "Dice": 0.665513

Final accuracy: 71.2%

Geoprocessing | Image Classification | History

Image Classification

Deep Learning Model Reviewer

Model

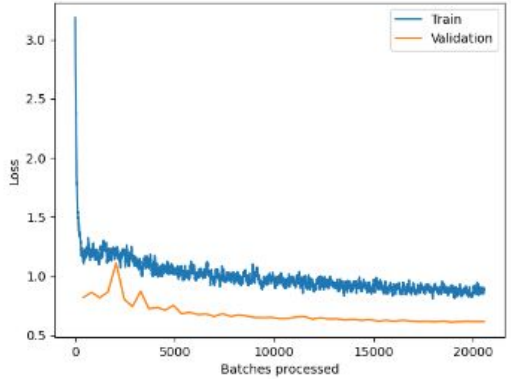
model_dl03i

Model Type: DeepLab

Backbone: resnet18

Learning Rate: slice('9.1201e-04', '9.1201e-03', None)

Training and Validation Loss



Analysis of the Model

Accuracy: 7.4501e-01

Epochs Details

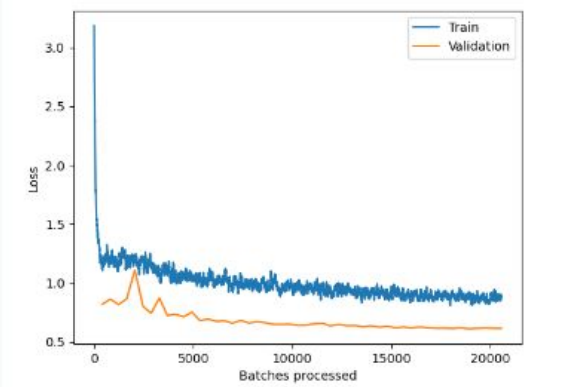
Epoch	Traini	Valida	Time	Other Metrics
1	1.141	0.820	00:03	"accuracy": 0.684383 "Dice": 0.638993
2	1.206	0.866	00:03	"accuracy": 0.673726

Starting accuracy: 68.4%

Geoprocessing | Image Classification | History

Image Classification

Deep Learning Model Reviewer



Analysis of the Model

Accuracy: 7.4501e-01

Epochs Details

Epoch	Traini	Valida	Time	Other Metrics
47	0.876	0.613	00:03	"accuracy": 0.744821 "Dice": 0.657717
48	0.865	0.618	00:03	"accuracy": 0.746082 "Dice": 0.661399
49	0.874	0.614	00:03	"accuracy": 0.745753 "Dice": 0.659400
50	0.884	0.614	00:03	"accuracy": 0.744767 "Dice": 0.654884

Final accuracy: 74.5%

Geoprocessing | Image Classification | History

Image Classification

Deep Learning Model Reviewer

Model

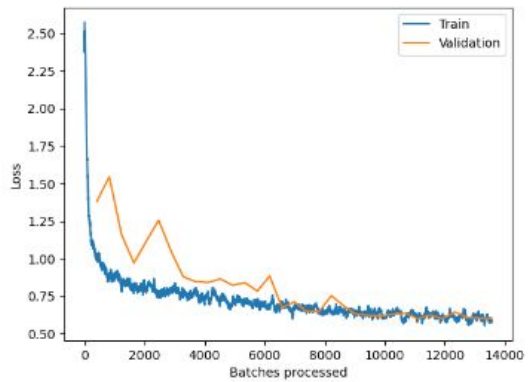
model_dl03j

Model Type: UnetClassifier

Backbone: timm:mobilenetv2_100

Learning Rate: slice('1.0965e-05', '1.0965e-04', None)

Training and Validation Loss



Analysis of the Model

Accuracy: 7.3115e-01

Epochs Details

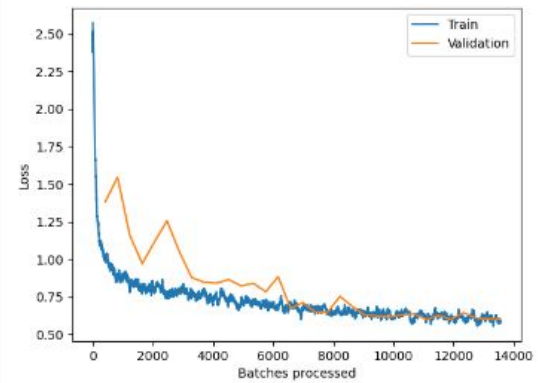
Epoch	Traini	Valida	Time	Other Metrics
1	0.992	1.381	00:11	"accuracy": 0.643808 "Dice": 0.634925
2	0.873	1.546	00:11	"accuracy": 0.629452

Starting accuracy: 64.4%

Geoprocessing | Image Classification | History

Image Classification

Deep Learning Model Reviewer



Analysis of the Model

Accuracy: 7.3115e-01

Epochs Details

Epoch	Traini	Valida	Time	Other Metrics
30	0.603	0.645	00:09	"accuracy": 0.719369 "Dice": 0.681493
31	0.585	0.609	00:09	"accuracy": 0.722685 "Dice": 0.641512
32	0.622	0.603	00:09	"accuracy": 0.734685 "Dice": 0.677773
33	0.591	0.603	00:09	"accuracy": 0.734411 "Dice": 0.662576

Final accuracy: 73.4%

Geoprocessing | Image Classification | History

Image Classification

Deep Learning Model Reviewer

Model

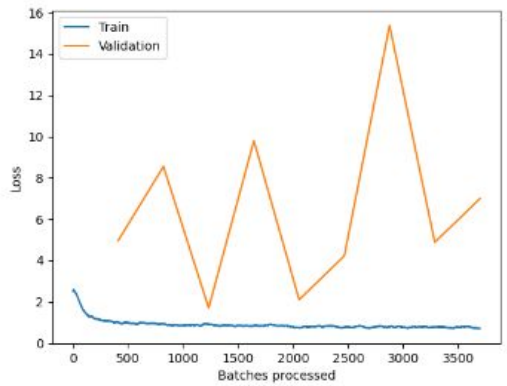
model_dl03k

Model Type: UnetClassifier

Backbone: timm:mobilenetv3_rw

Learning Rate: slice('1.5849e-05', '1.5849e-04', None)

Training and Validation Loss



Analysis of the Model

Accuracy: 6.4356e-01

Epochs Details

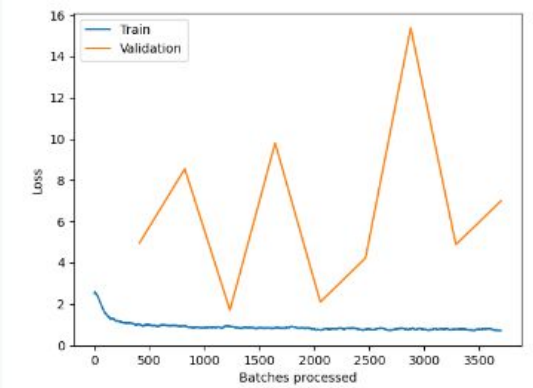
Epoch	Traini	Valida	Time	Other Metrics
1	1.023	4.969	00:07	"accuracy": 0.631808 "Dice": 0.628891
2	0.924	8.550	00:07	"accuracy": 0.618794

Starting accuracy: 63.2%

Geoprocessing | Image Classification | History

Image Classification

Deep Learning Model Reviewer



Analysis of the Model

Accuracy: 6.4356e-01

Epochs Details

Epoch	Traini	Valida	Time	Other Metrics
6	0.776	4.230	00:07	"accuracy": 0.657424 "Dice": 0.636235
7	0.781	15.37	00:07	"accuracy": 0.635013 "Dice": 0.632625
8	0.773	4.883	00:07	"accuracy": 0.629643 "Dice": 0.631131
9	0.711	7.001	00:08	"accuracy": 0.652657 "Dice": 0.633930

Final accuracy: 65.3%

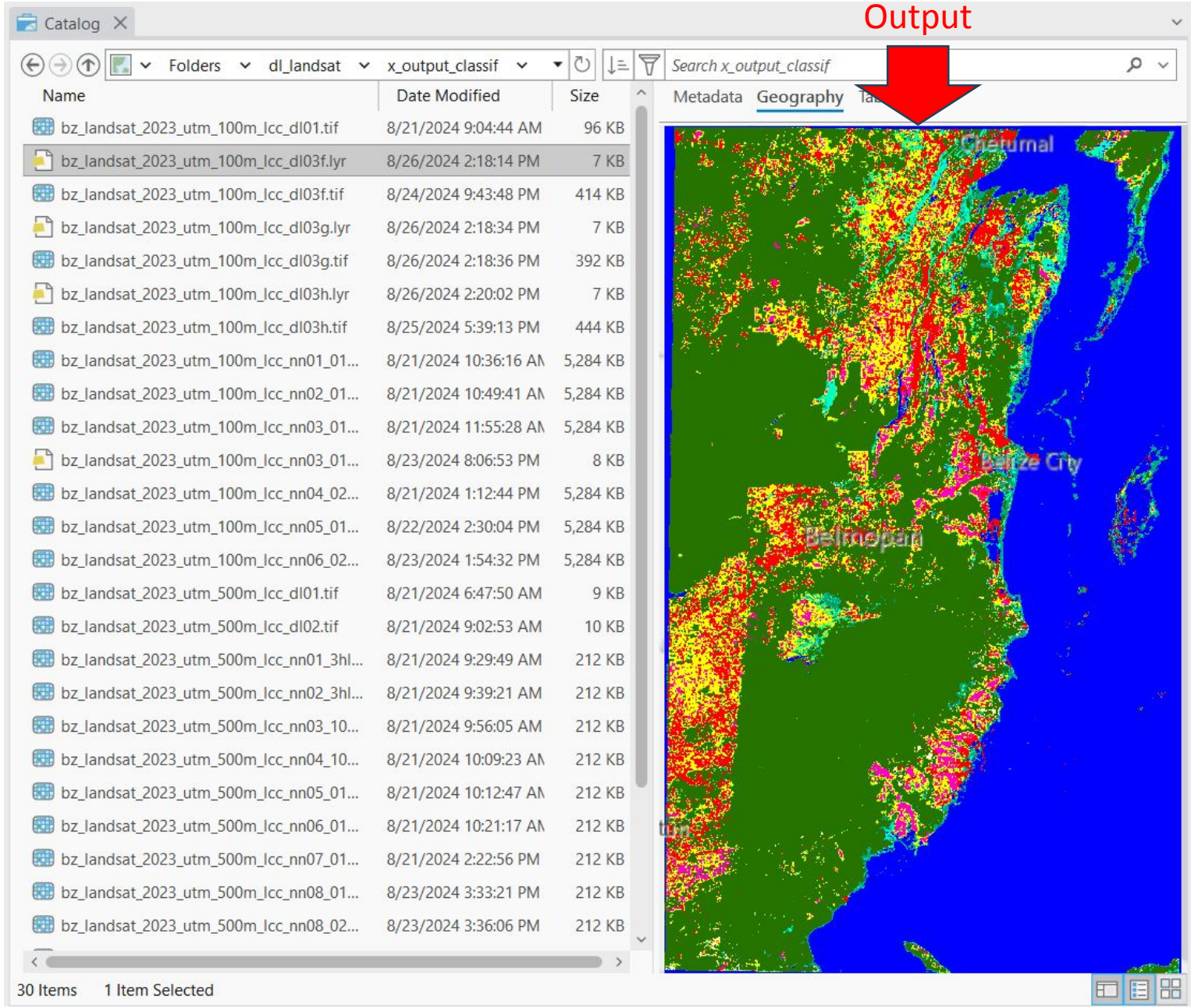
Geoprocessing | Image Classification | History

Step 5: Reviewing the model outputs

U-Net

ResNet-18

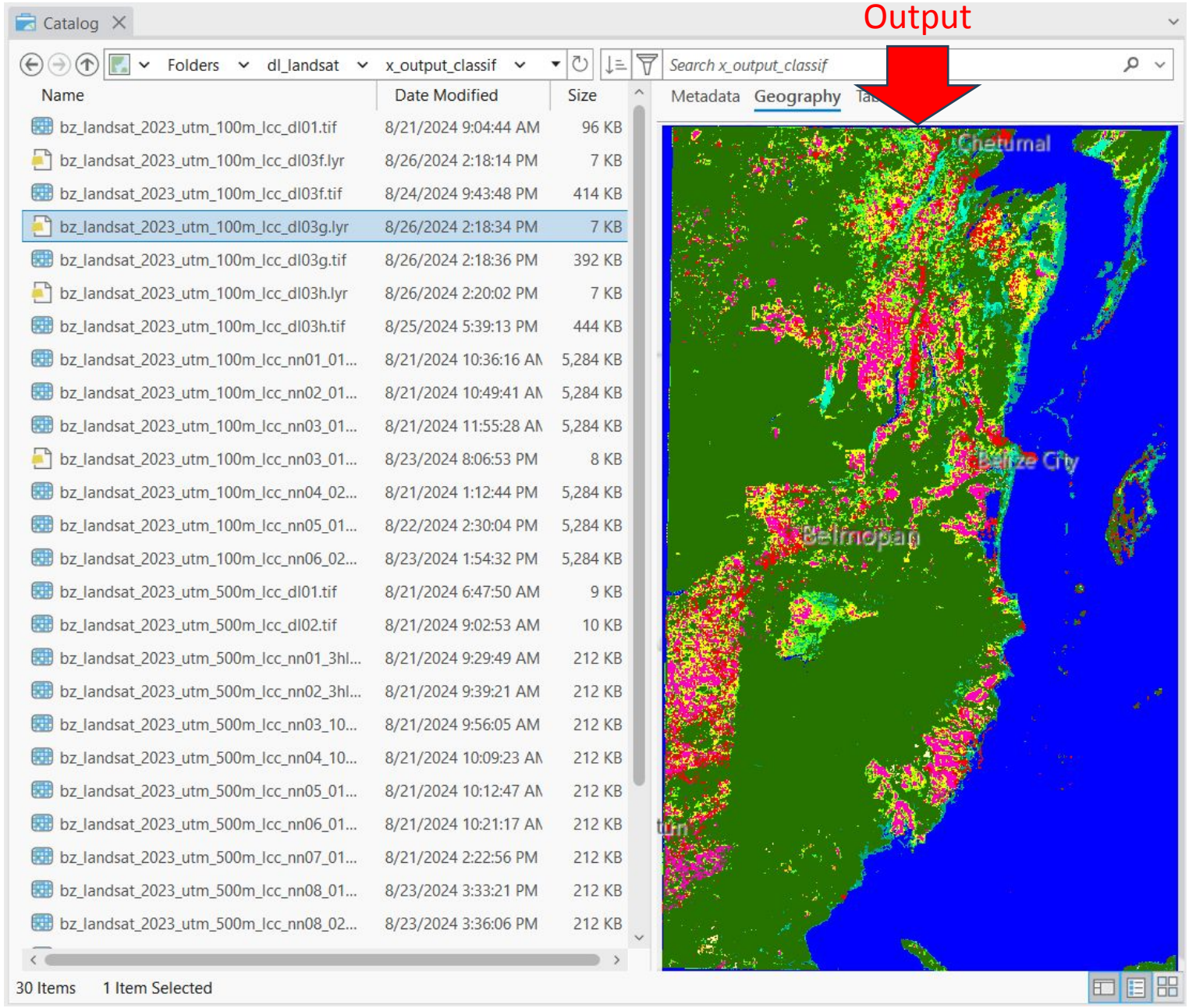
20 epochs



U-Net

ResNet-18

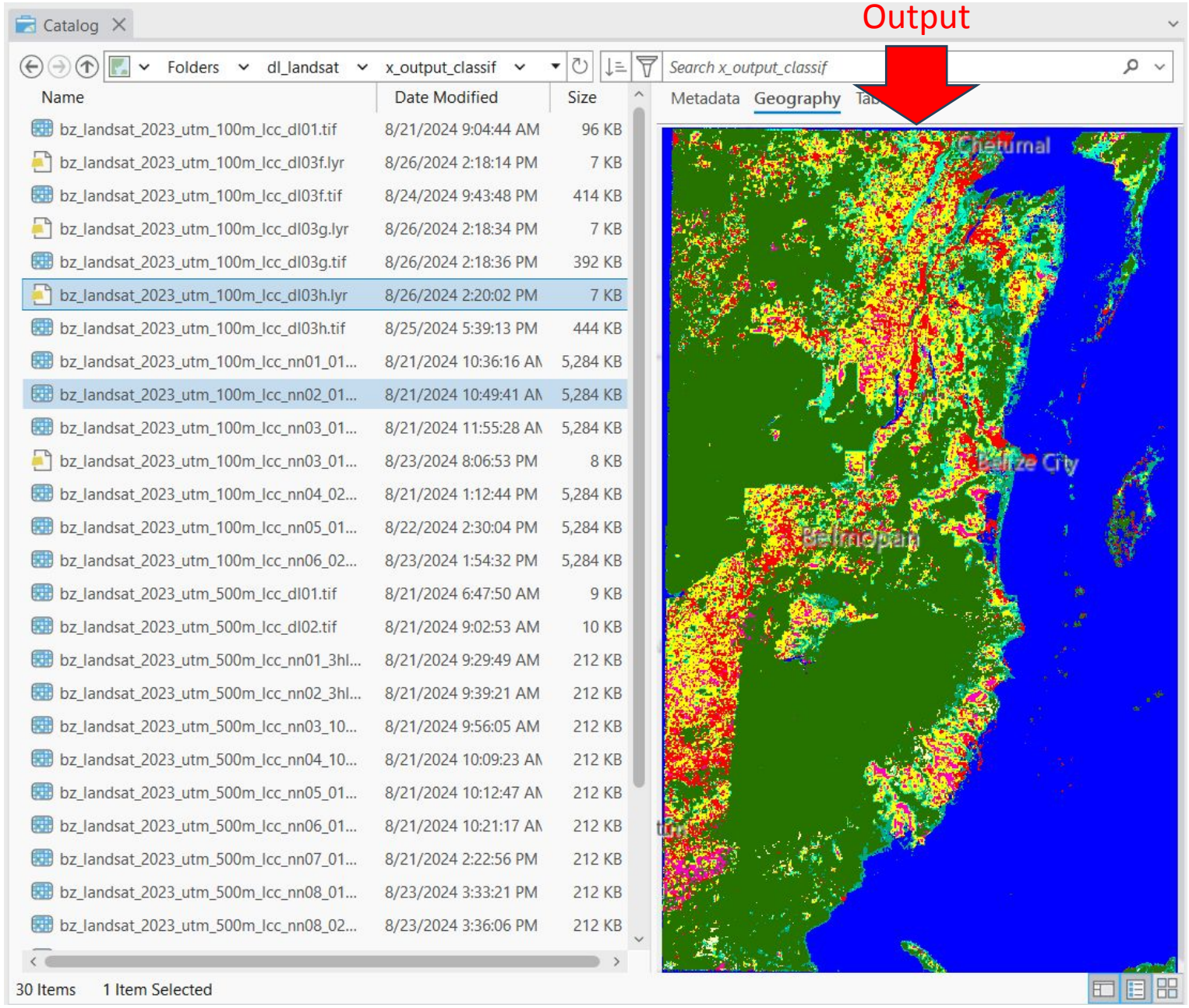
50 epochs



U-Net

ResNet-34

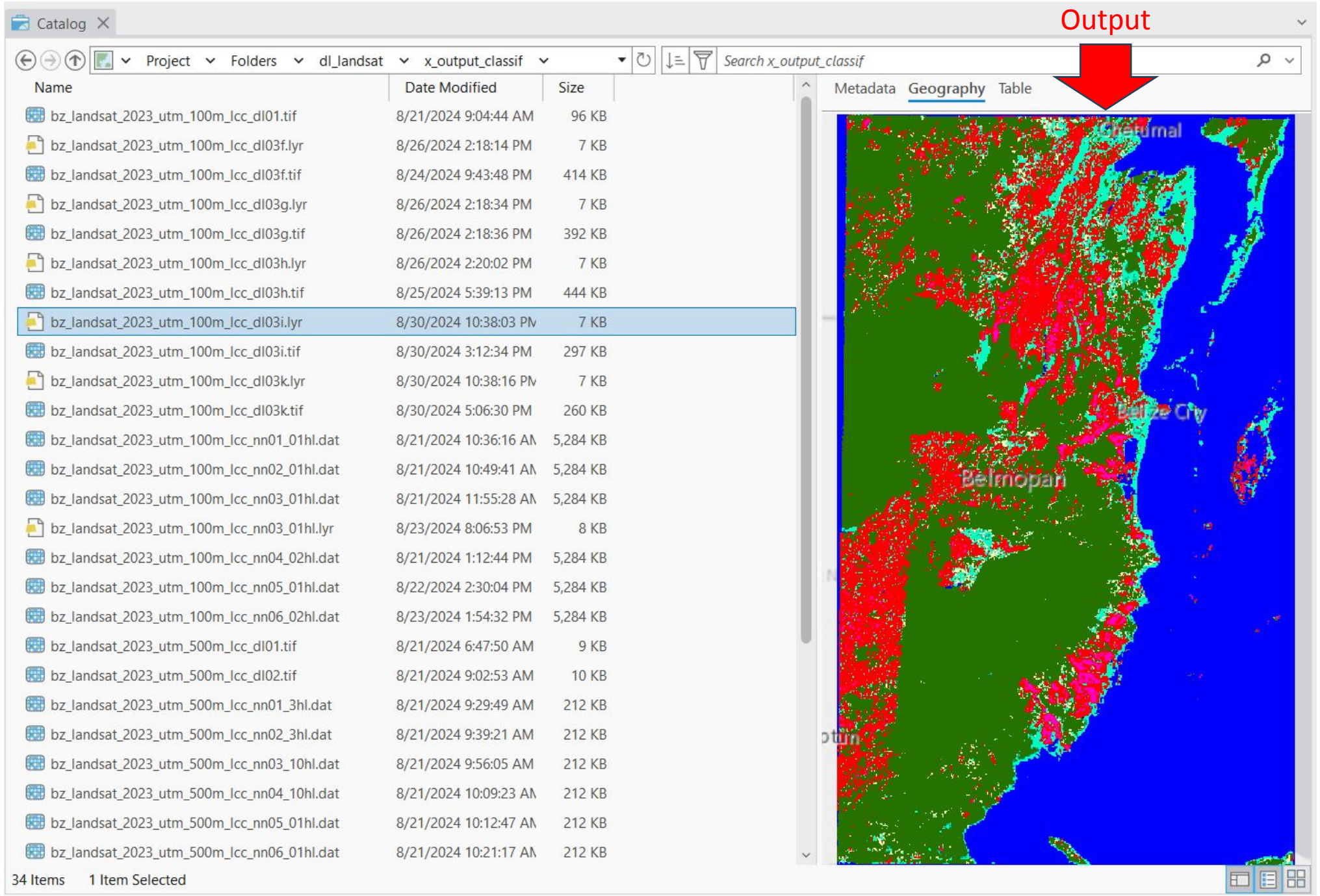
50 epochs



DeepLab V3

ResNet-18

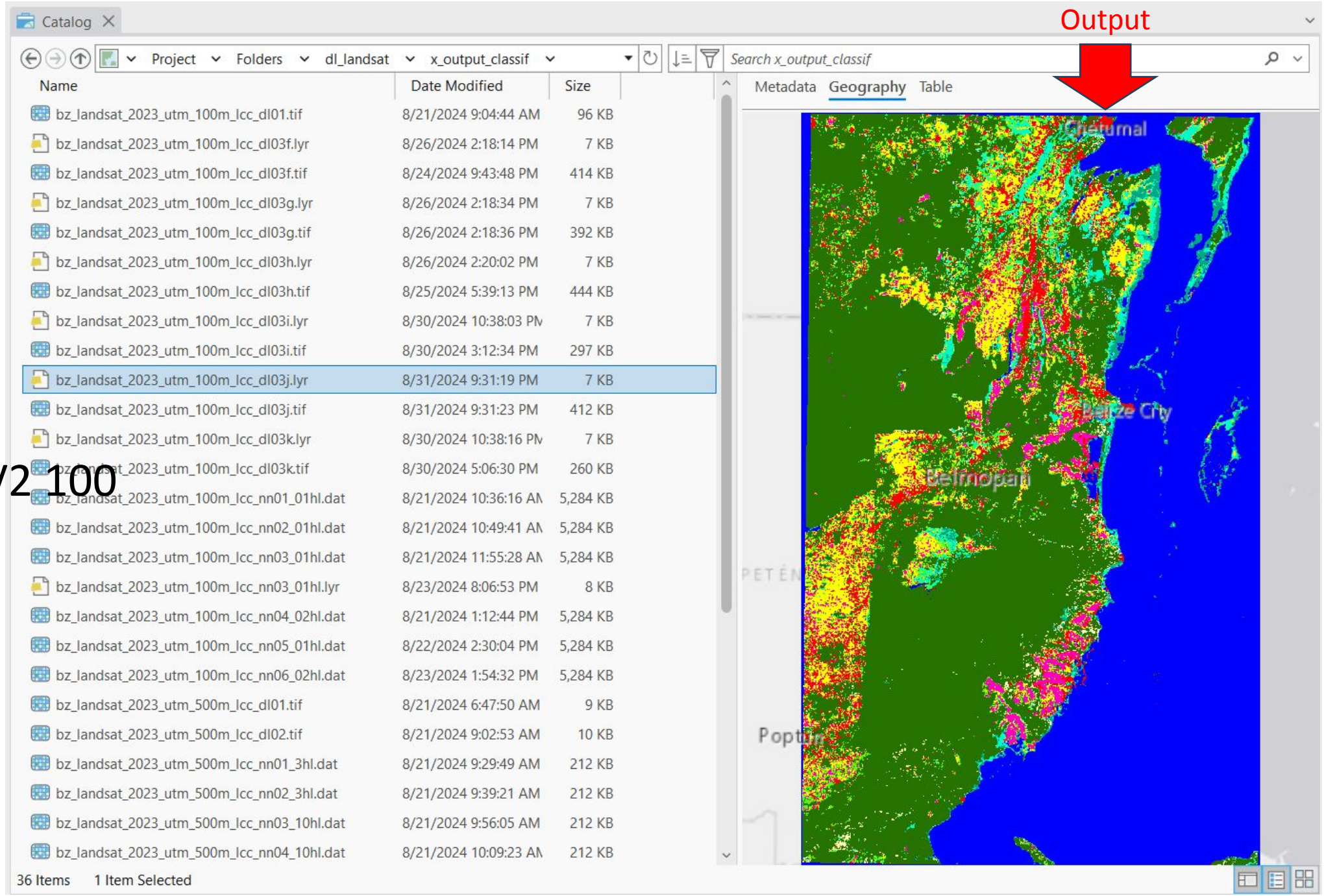
50 epochs



U-Net

MobileNet V2_100

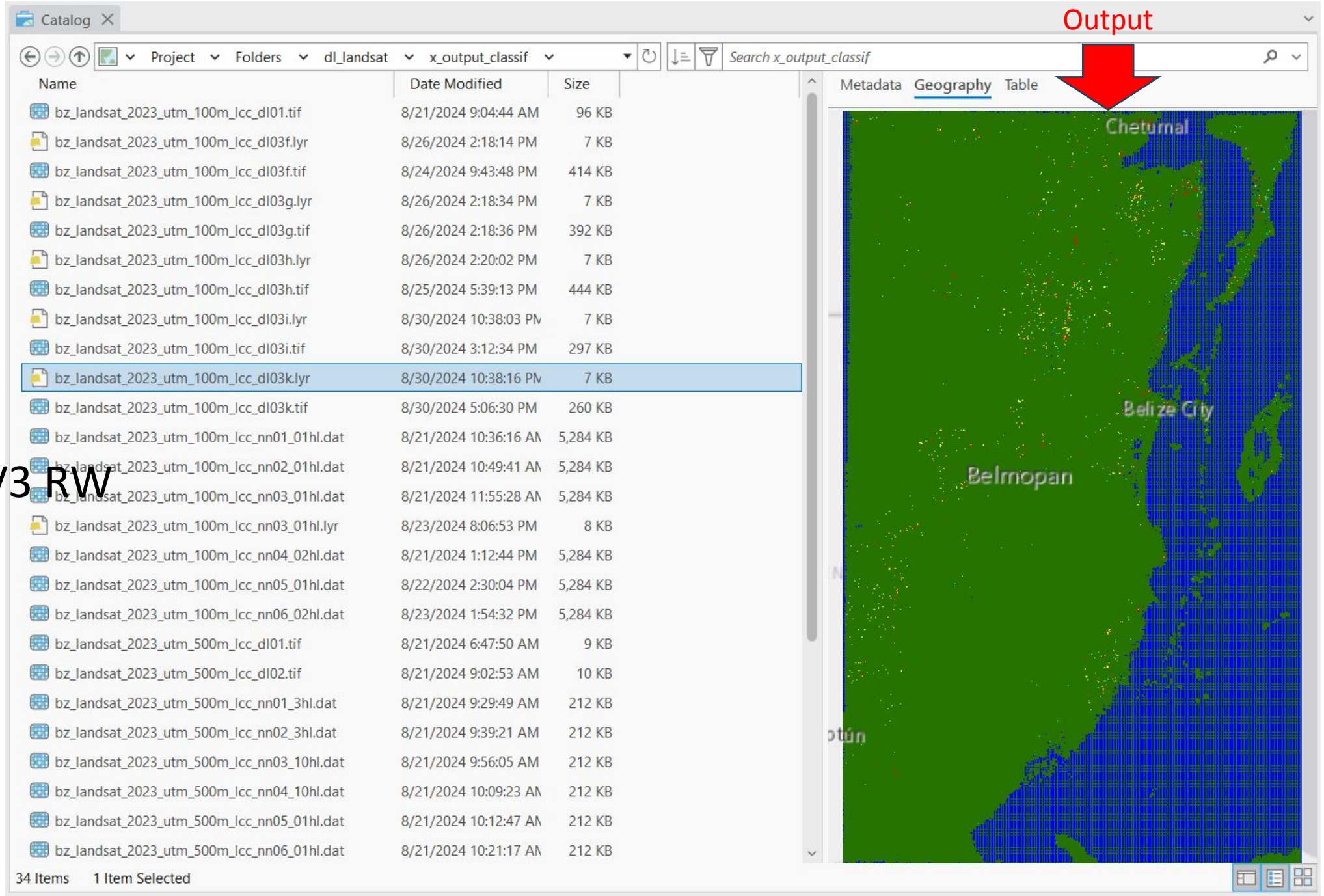
50 epochs



U-Net

MobileNet V3 RW

50 epochs



Documenting model runs

Run no.	162	163	164	165	166	167	168	169	170	171	172	173	174	175
Batch alias	Batch 27	Batch 27	Batch 27	Batch 27	Batch 27	Batch 28	Batch 28	Batch 28	Batch 28	Batch 28	Batch 28	Batch 28	Batch 28	Batch 28
Operation type	classification	classification	classification	classification	classification	classification	classification	classification	classification	classification	classification	classification	classification	classification
Platform	ArcGIS Pro 3.4	ArcGIS Pro 3.4	ArcGIS Pro 3.4	ArcGIS Pro 3.4	ArcGIS Pro 3.4	ArcGIS Pro 3.4	ArcGIS Pro 3.4	ArcGIS Pro 3.5	ArcGIS Pro 3.4	ArcGIS Pro 3.4	ArcGIS Pro 3.4	ArcGIS Pro 3.5	ArcGIS Pro 3.5	ArcGIS Pro 3.5
Input training data	v2_2023	v2_2023	v2_2023	v2_2023	v2_2023	v2_2023	v2_2023	v2_2023	v2_2023	v2_2023	v2_2023	v2_2023	v2_2023	v2_2023
Geographic domain	BZ	BZ	BZ	BZ	BZ	BZ	BZ	BZ	BZ	BZ	BZ	BZ	BZ	BZ
Input imagery	Landsat + S1*	Landsat + S1*	Landsat + S1	Landsat + S1	Landsat + S1	EFM v2	EFM v2	EFM v2	Landsat	EFM v2	EFM v2	EFM v2	EFM v2	EFM v2
Imagery year	2024	2024	2024	2024	2024	2024	2024	2024	2023	2024	2024	2024	2024	2024
No. bands	64	64	64	132	132	64	64	64	6	64	64	64	64	64
Spatial resolution	100m	100m	100m	100m	100m	100m	100m	100m	30m	100m	100m	100m	100m	100m
Tile width / chip size (px)	10	10	10	10	10	10	10	10	32	10	10	10	10	10
Model type	U-Net	U-Net	U-Net	U-Net	U-Net	U-Net	U-Net	U-Net	U-Net	U-Net	U-Net	U-Net	U-Net	U-Net
Backbone model	RESNET18	RESNET18	RESNET18	RESNET18	RESNET18	RESNET18	RESNET34	RESNET18	RESNET34	RESNET34	RESNET50	RESNET34	RESNET34	RESNET18
No. layers	18	18	18	18	18	18	34	18	34	34	50	34	34	18
Epochs (max set)	20	50	20	20	20	20	20	20	20	20	20	20	20	20
Epochs (actual)	20	24	20	20	20	20	20	20	20	18	20	N/A	15	20
Accuracy (%)	75.84%	72.67%	63.13%	57.88%	80.30%	75.34%	75.62%	76.48%	75.69%	75.72%	75.97%	N/A	75.10%	75.46%
Dice (%)	61.62%	59.04%	61.24%	55.86%	73.99%	63.61%	60.39%	63.62%	69.32%	62.67%	62.88%	N/A	59.69%	59.76%
Date run	15-03-2025	15-03-2025	15-03-2025	15-03-2025	15-03-2025	20-05-2025	20-05-2025	20-05-2025	26-05-2025	26-05-2025	26-05-2025	26-05-2025	26-05-2025	28-05-2025
Time: Training	4 min	5 min	5 min	6 min	6 min	4 min	4 min	4 min	11 min	4 min	4 min	2 min	3 min	4 min
Computer alias	PC6	PC6	PC6	PC6	PC6	PC4	PC4	PC6	PC5	PC5	PC5	PC5	PC6	PC4
Model size (MB)	92 MB	92 MB	93 MB	94 MB	94 MB	192 MB	266 MB	192 MB	303 MB	266 MB	1.83 GB	266 MB	266 MB	192 MB
Inferenced?	Yes	Yes	Yes	Yes	Yes	Yes	Yes	No	No	No	No	No	No	No
Time: Inferencing	8 min	8 min	8 min	9 min	8 min	5 min	7 min							
Feasible output?	Yes	Somewhat	No	No	Somewhat	Yes	Somewhat							
Comments on the run		*Looks overfit re: urban areas*				*Output looks fairly good.*	*Urban areas look overfit.*							
Belize City well defined?	Yes	Yes	No	No	Somewhat	Yes	Yes							
Toledo sec forests defined?	Yes	Yes	No	No	No	Yes	Somewhat							
Urban areas balanced?	Yes	No	No	No	Somewhat	Somewhat	No							
bit.ly/geoml_model_comp														

<u>Issue 3:</u> ***What happens if we run the model w/ inappropriate inputs + parameters?***

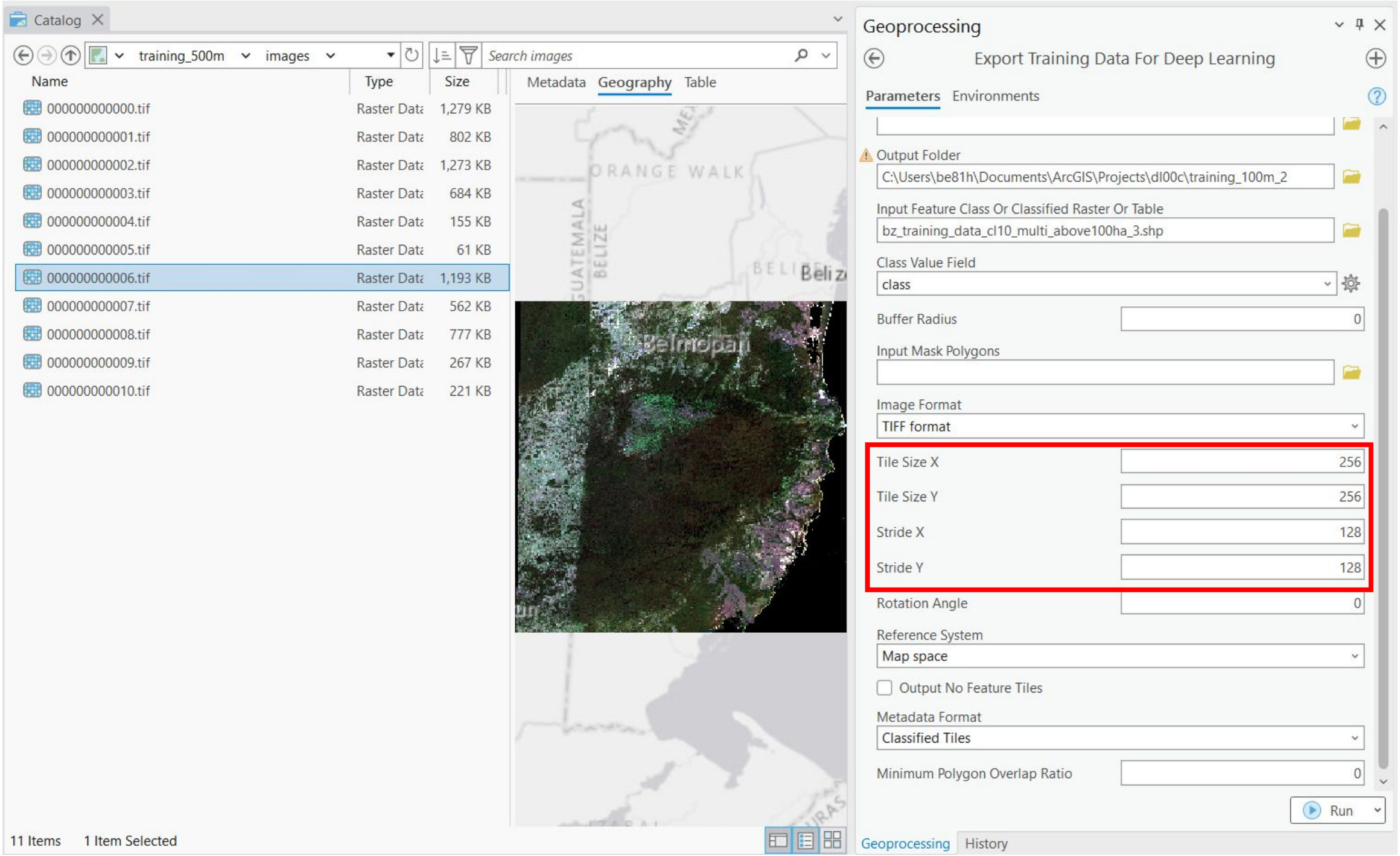

Catalog
training_500m
images
Search images
Name
Type
Size
000000000000.tif Raster Data 1,279 KB
000000000001.tif Raster Data 802 KB
000000000002.tif Raster Data 1,273 KB
000000000003.tif Raster Data 684 KB
000000000004.tif Raster Data 155 KB
000000000005.tif Raster Data 61 KB
000000000006.tif Raster Data 1,193 KB
000000000007.tif Raster Data 562 KB
000000000008.tif Raster Data 777 KB
000000000009.tif Raster Data 267 KB
000000000010.tif Raster Data 221 KB
Metadata
Geography
Table
ORANGE WALK
GUATEMALA
BELIZE
Belmopan
11 Items
1 Item Selected
Geoprocessing
Export Training Data For Deep Learning
Parameters
Environments
Output Folder
C:\Users\be81h\Documents\ArcGIS\Projects\dl00c\training_100m_2
Input Feature Class Or Classified Raster Or Table
bz_training_data_cl10_multi_above100ha_3.shp
Class Value Field
class
Buffer Radius
0
Input Mask Polygons
Image Format
TIFF format
Tile Size X
256
Tile Size Y
256
Stride X
128
Stride Y
128
Rotation Angle
0
Reference System
Map space
Output No Feature Tiles
Metadata Format
Classified Tiles
Minimum Polygon Overlap Ratio
0
Run
Geoprocessing
History

Image Classification

Deep Learning Model Reviewer

Model

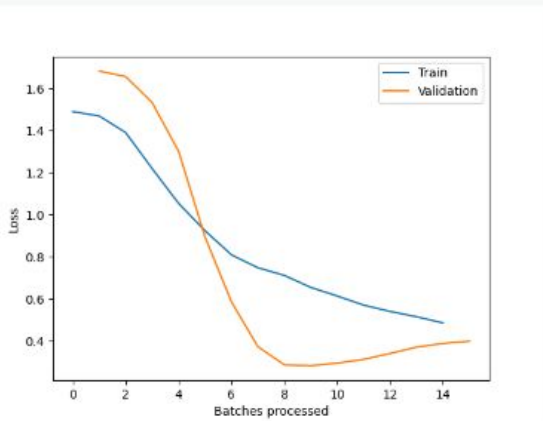
model_dl03d

Model Type: UnetClassifier

Backbone: timm:vgg11_bn

Learning Rate: slice('4.3652e-06', '4.3652e-05', None)

Training and Validation Loss

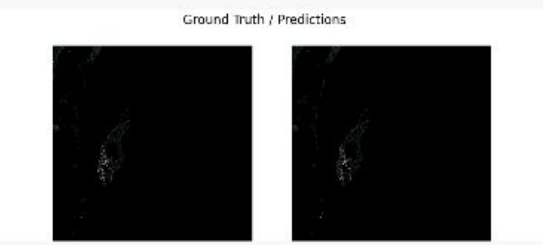


Analysis of the Model

Accuracy: 9.9963e-01

Final accuracy: 99.9%

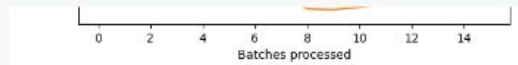
Results



Geoprocessing | Image Classification | History

Image Classification

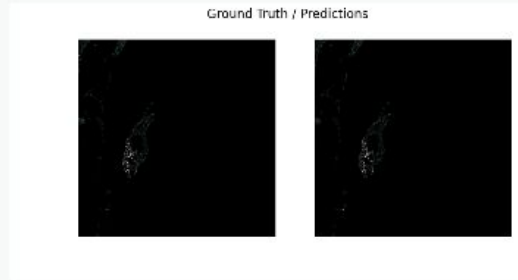
Deep Learning Model Reviewer



Analysis of the Model

Accuracy: 9.9963e-01

Results



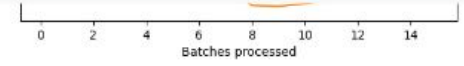
Epochs Details

Epoch	Traini	Valid	Time	Other Metrics
1	1.49	1.68	00:0	"accuracy": 0.489944 "Dice": 5.981576
2	1.46	1.60	00:0	"accuracy": 0.511276 "Dice": 6.242586
3	1.39	1.53	00:0	"accuracy": 0.734802 "Dice": 0.000115
4	1.21	1.30	00:0	"accuracy": 0.977401 "Dice": 0.001342

Geoprocessing | Image Classification | History

Image Classification

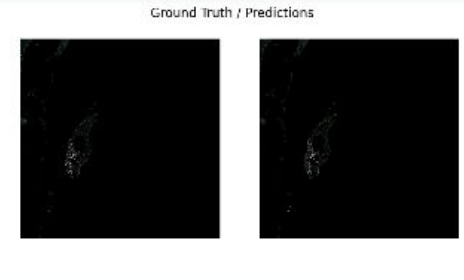
Deep Learning Model Reviewer



Analysis of the Model

Accuracy: 9.9963e-01

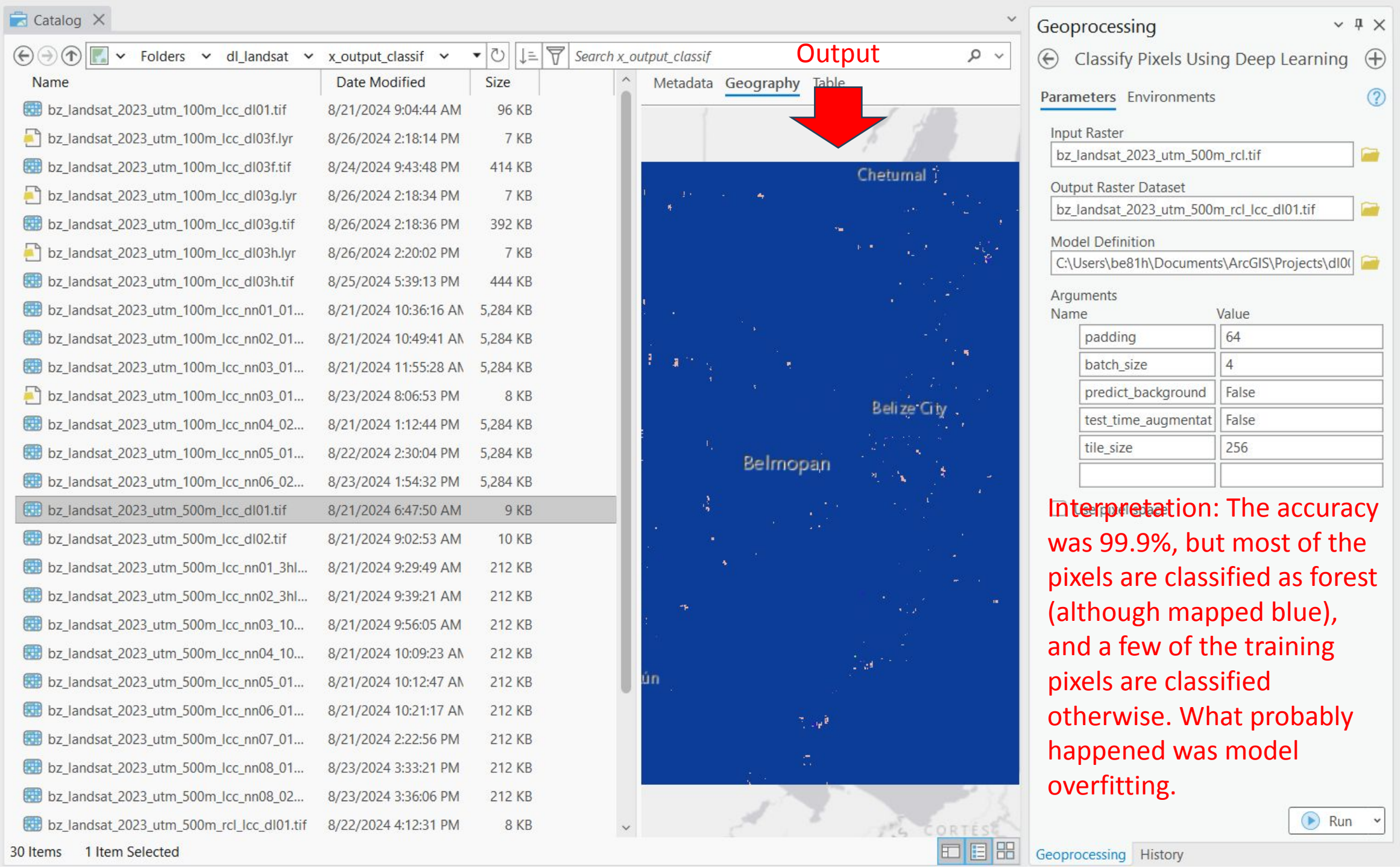
Results



Epochs Details

Epoch	Traini	Valid	Time	Other Metrics
12	0.57	0.34	00:0	"accuracy": 0.999710 "Dice": 0
13	0.54	0.37	00:0	"accuracy": 0.999694 "Dice": 0
14	0.51	0.38	00:0	"accuracy": 0.999710 "Dice": 0
15	0.48	0.39	00:0	"accuracy": 0.999710 "Dice": 0

Geoprocessing | Image Classification | History


Catalog
Folders
dl_landsat
x_output_classif
Search x_output_classif
Output
Name
Date Modified
Size
bz_landsat_2023_utm_100m_lcc_dl01.tif 8/21/2024 9:04:44 AM 96 KB
bz_landsat_2023_utm_100m_lcc_dl03f.lyr 8/26/2024 2:18:14 PM 7 KB
bz_landsat_2023_utm_100m_lcc_dl03f.tif 8/24/2024 9:43:48 PM 414 KB
bz_landsat_2023_utm_100m_lcc_dl03g.lyr 8/26/2024 2:18:34 PM 7 KB
bz_landsat_2023_utm_100m_lcc_dl03g.tif 8/26/2024 2:18:36 PM 392 KB
bz_landsat_2023_utm_100m_lcc_dl03h.lyr 8/26/2024 2:20:02 PM 7 KB
bz_landsat_2023_utm_100m_lcc_dl03h.tif 8/25/2024 5:39:13 PM 444 KB
bz_landsat_2023_utm_100m_lcc_nn01_01... 8/21/2024 10:36:16 AM 5,284 KB
bz_landsat_2023_utm_100m_lcc_nn02_01... 8/21/2024 10:49:41 AM 5,284 KB
bz_landsat_2023_utm_100m_lcc_nn03_01... 8/21/2024 11:55:28 AM 5,284 KB
bz_landsat_2023_utm_100m_lcc_nn03_01... 8/23/2024 8:06:53 PM 8 KB
bz_landsat_2023_utm_100m_lcc_nn04_02... 8/21/2024 1:12:44 PM 5,284 KB
bz_landsat_2023_utm_100m_lcc_nn05_01... 8/22/2024 2:30:04 PM 5,284 KB
bz_landsat_2023_utm_100m_lcc_nn06_02... 8/23/2024 1:54:32 PM 5,284 KB
bz_landsat_2023_utm_500m_lcc_dl01.tif 8/21/2024 6:47:50 AM 9 KB
bz_landsat_2023_utm_500m_lcc_dl02.tif 8/21/2024 9:02:53 AM 10 KB
bz_landsat_2023_utm_500m_lcc_nn01_3hl... 8/21/2024 9:29:49 AM 212 KB
bz_landsat_2023_utm_500m_lcc_nn02_3hl... 8/21/2024 9:39:21 AM 212 KB
bz_landsat_2023_utm_500m_lcc_nn03_10... 8/21/2024 9:56:05 AM 212 KB
bz_landsat_2023_utm_500m_lcc_nn04_10... 8/21/2024 10:09:23 AM 212 KB
bz_landsat_2023_utm_500m_lcc_nn05_01... 8/21/2024 10:12:47 AM 212 KB
bz_landsat_2023_utm_500m_lcc_nn06_01... 8/21/2024 10:21:17 AM 212 KB
bz_landsat_2023_utm_500m_lcc_nn07_01... 8/21/2024 2:22:56 PM 212 KB
bz_landsat_2023_utm_500m_lcc_nn08_01... 8/23/2024 3:33:21 PM 212 KB
bz_landsat_2023_utm_500m_lcc_nn08_02... 8/23/2024 3:36:06 PM 212 KB
bz_landsat_2023_utm_500m_rcl_lcc_dl01.tif 8/22/2024 4:12:31 PM 8 KB
30 Items
1 Item Selected
Metadata
Geography
Table
Chetumal
Belize City
Belmopan
Geoprocessing
Classify Pixels Using Deep Learning
Parameters
Environments
Input Raster
bz_landsat_2023_utm_500m_rcl.tif
Output Raster Dataset
bz_landsat_2023_utm_500m_rcl_lcc_dl01.tif
Model Definition
C:\Users\be81h\Documents\ArcGIS\Projects\dl0(
Arguments
Name
Value
padding
64
batch_size
4
predict_background
False
test_time_augmentat
False
tile_size
256
Use current extent
Interpretation: The accuracy was 99.9%, but most of the pixels are classified as forest (although mapped blue), and a few of the training pixels are classified otherwise. What probably happened was model overfitting.
Run
Geoprocessing
History

So what have we learned?

…Deep learning doesn't have to be unattainable.

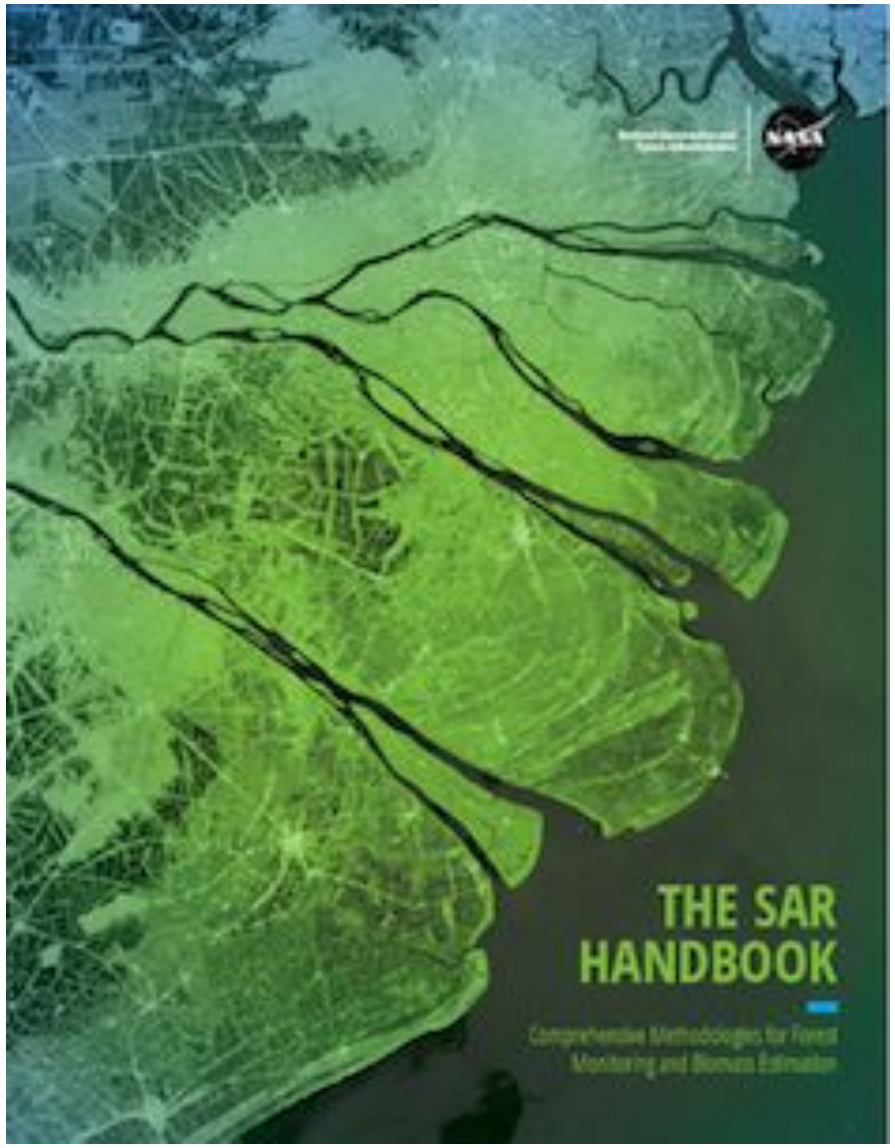
Summary

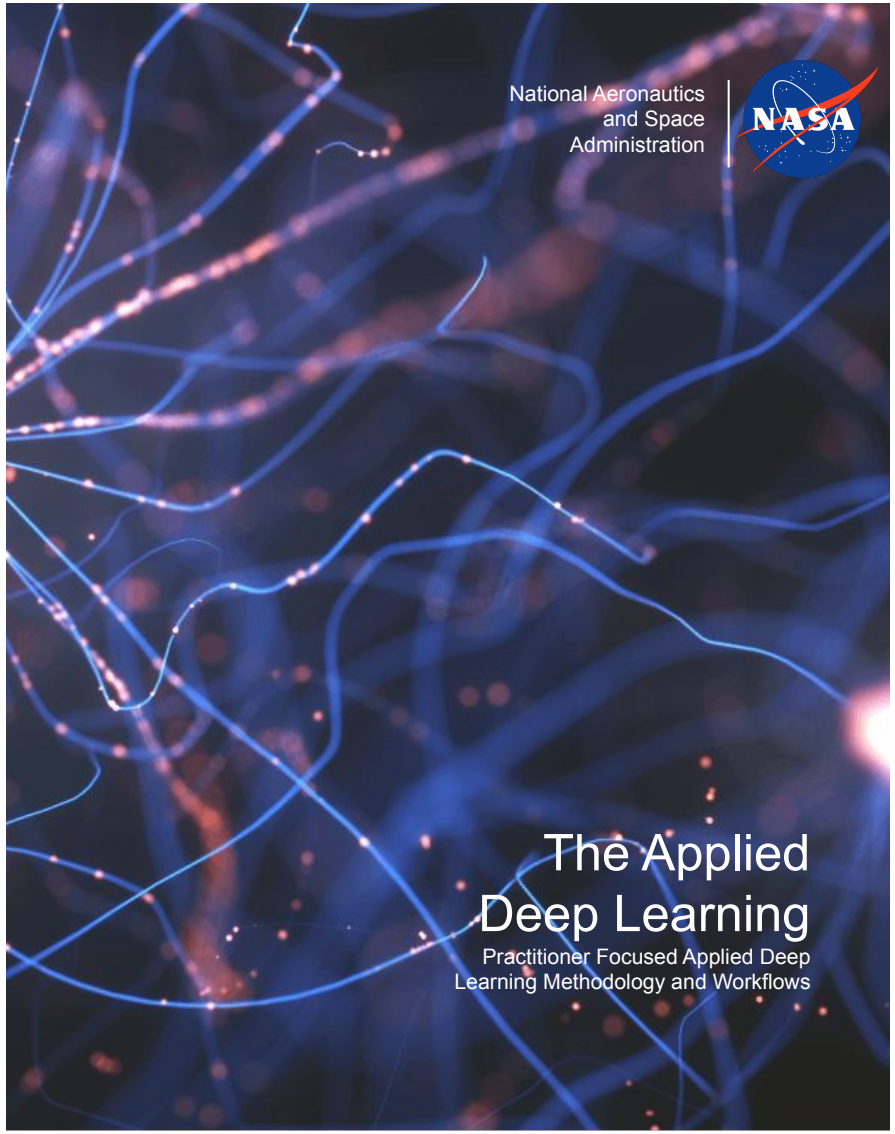
- With ArcGIS Pro 3.x, it is possible to execute a deep learning-based land cover classification in just three steps, provided we have:
  - installed the requisite external deep learning libraries
  - used the appropriate input satellite imagery and training data
  - selected appropriate parameters (e.g., training data metadata format, tile size, backbone model)
- Three steps (we need to):
  1. export our training data in a format ArcGIS can ingest to
  2. train a deep learning model, to then
  3. perform the classification.
- However, *if we only use ArcGIS' default parameters* and do not adjust the parameters carefully, **the classification will not work [properly]**.
- Depending on the dimensions of the input data, the entire process (*excluding collecting training data*) may take ~3-4 hours on a standard laptop (w/o a GPU).
- ArcGIS Pro provides additional tools for evaluating the training data + the model performance and visualizing the model outputs.

Future Work

- If you have been able to implement the classification, congratulations! You have successfully on-ramped to the world of deep learning in Earth observation.
- You might want to look into other ways to implement deep learning-based workflows, incl.:
  - the other options in the Deep Learning toolbox in ArcGIS Pro
  - other options using QGIS (https://plugins.qgis.org/search/?q=deep+learning)
  - cloud-based options outside of ArcGIS (e.g., TensorFlow + Vertex AI via GEE / Colab)
- For additional background:
  - NASA ARSET: “Fundamentals of Machine Learning for Earth Science” (2023)
  - EO College: “Introduction to Machine Learning in Earth Observation” MOOC (2024)
  - Google: “Deep Learning with TensorFlow and Earth Engine” resources (2022)

Knowledge Products





README Apache-2.0 license

Geospatial Artificial Intelligence (Geo AI) / Geospatial Machine Learning (Geo ML) Compendium:

Resources 👷 focused on land cover monitoring 🏞️ 🛰️

🚨 *This compendium is a work in progress.* 🚨

DOI 10.5281/zenodo.16735094 last updated yesterday visitors 69

Overview

1. Objectives
2. Schematics
3. Resources: Machine learning (classical)
4. Resources: Deep learning
5. Resources: Uncertainty assessment / accuracy assessment
6. Resources: Other
7. Software installers

Readme
Apache-2.0 license
Cite this repository
Activity
1 star
0 watching
0 forks

Releases 1

v.1.0.0 Latest
2 days ago

Packages

No packages published
Publish your first package

Acknowledgements

- Tim Mayer (SCO) + Biplov Bhandari (Woolpert)
- Dr. Kel Markert + Dr. Nick Clinton (Google)
- Dr. David Saah (U. San Francisco)
- Dr. Modou Mbaye (UCAD), Ujaval Gandhi (Spatial Thoughts)
- Dr. Glory Enaruvbe (AFRIGIST), Edgar Correa (BZ FD)
- Drs. Pavan Yadav + Sangeet Matthew (Esri)

Questions?

email me @

emil.cherrington@uah.edu