WHAT’S NEW IN ENVI DEEP LEARNING 2.1

May 2023

Zach Norman
Bill Okubo

Agenda

Introduction

Deep Learning

Machine Learning

Notables

Q + A

Speakers Slide

Bill Okubo

Product Manager

bill.okubo@nv5.com

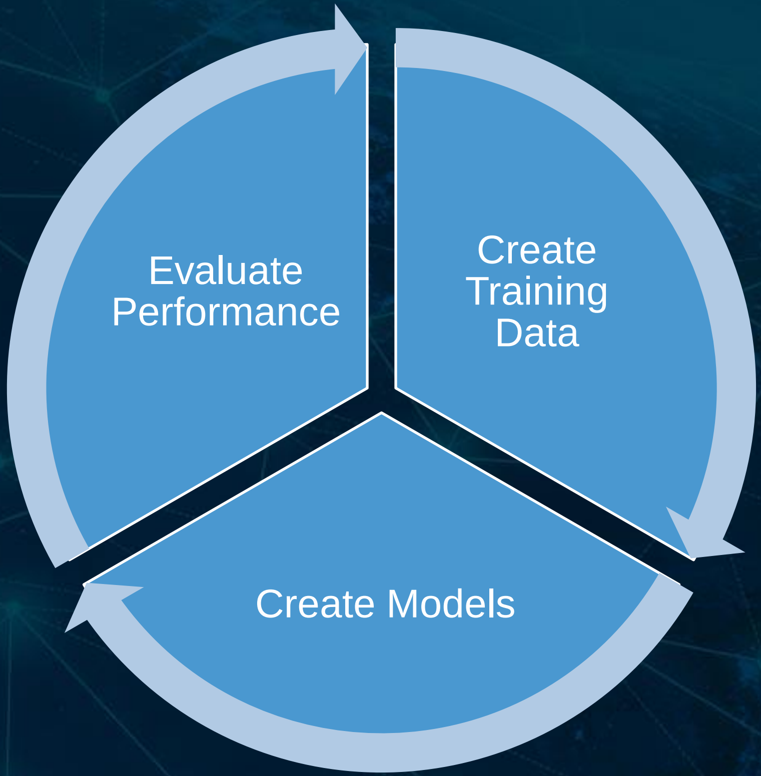
Zach Norman

Product Manager

zachary.norman@nv5.com

Context: The ENVI Deep Learning Module

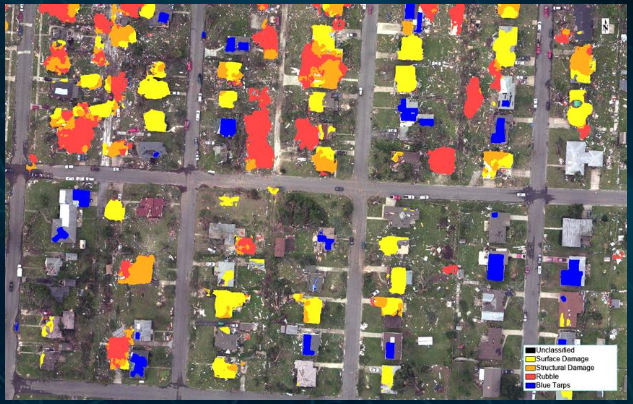
Applied deep learning for geospatial imagery in ENVI, the leading remote sensing and image analysis software



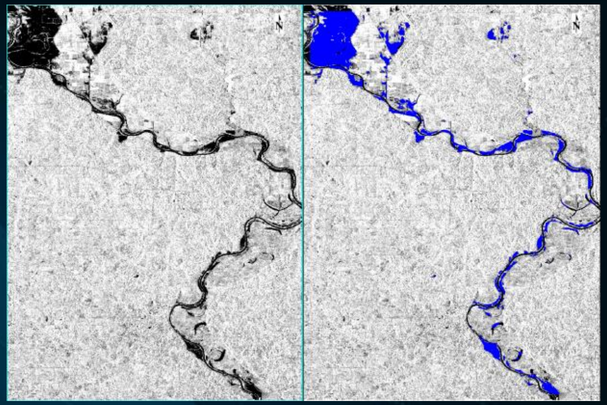
Deep learning workflow in ENVI, built on TensorFlow and Keras

Without needing to program, the capabilities include:

- Segmentation (i.e. cloud masking)
- Object detection (i.e. cars or ships)
- Linear feature extraction (i.e. roads)
- Support for nearly any image format and data modality



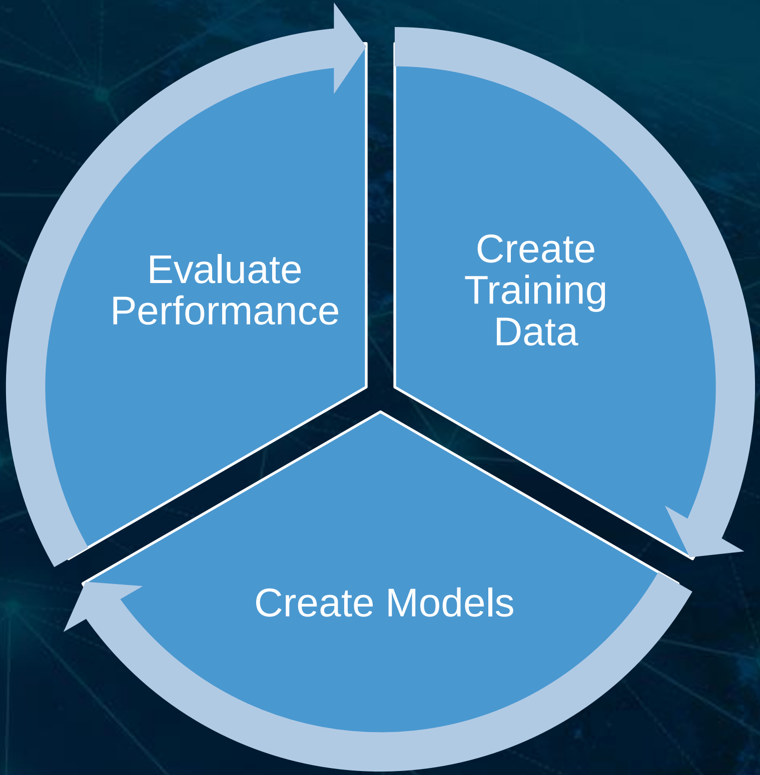
Assess building damage after hurricanes and tornadoes



Automated flood detection using SAR

Context: ENVI Machine Learning

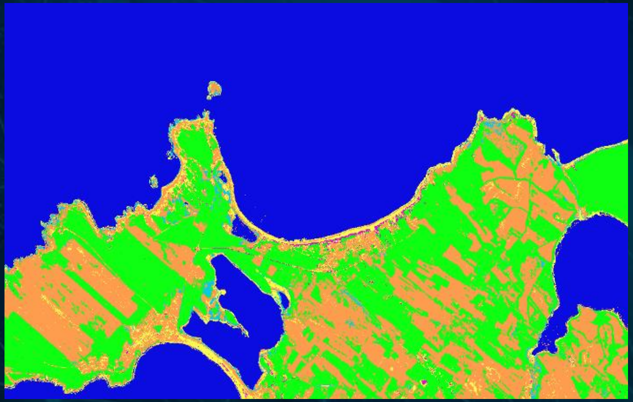
Applied machine learning for geospatial imagery in ENVI, the leading remote sensing and image analysis software



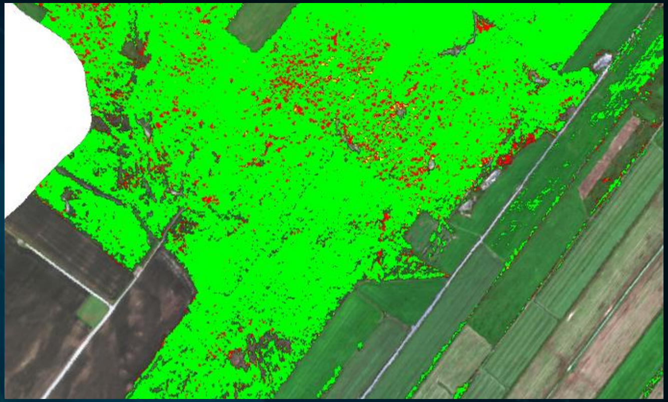
Machine learning follows the same workflow as deep learning, but takes a lot less time

Without needing to program, you have access to:

- Supervised classification (i.e. Random Forest)
- Unsupervised classification (i.e. K-Means)
- Anomaly detection (i.e. Isolation Forest)
- Supports nearly any image format, data modality, or number of bands



Get answers in minutes, not hours using machine learning



Use machine learning to process hyperspectral imagery with any number of bands

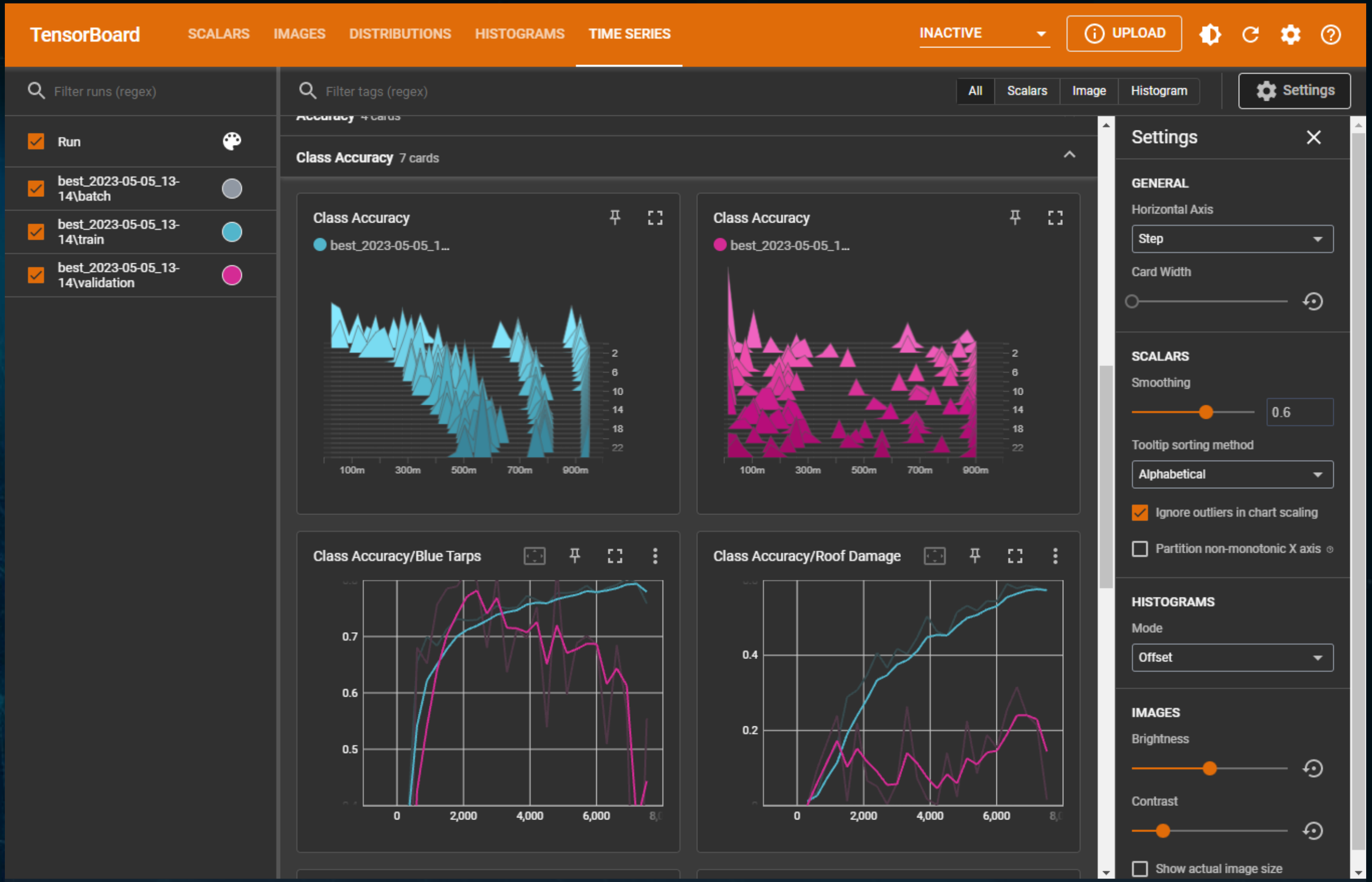
Machine Learning vs Deep Learning: Hardware and Access

MACHINE LEARNING	DEEP LEARNING
• **<u>FREE and no additional license required</u>**, packaged with “ENVI Deep Learning” • Available starting with ENVI 5.6.3 • Uses the CPU • Recommended 16 GB RAM • Libraries optimized for Intel, but AMD also supported • Some algorithms have significant performance improvements with Intel 	• Need to purchase the “ENVI Deep Learning” module to use • Latest version with be 2.0 and will have machine learning • Requires a GPU • Minimum 8 GB GPU RAM • Only NVIDIA GPUs are supported

DEEP LEARNING

Deep Insights Using TensorBoard

- TensorBoard is TensorFlow’s visualization engine for deep learning model performance
- TensorBoard allows us to:
  - Visualize performance metrics (accuracy, precision recall, loss, etc.)
  - Display metrics from separate training sessions in one chart
  - Create visualizations for other types of data, such as images, audio, or text



☑ Run

☑ best_2023-05-05_13-14\batch ●

☑ best_2023-05-05_13-14\train ●

☑ best_2023-05-05_13-14\validation ●

Settings

GENERAL

Horizontal Axis: Step

Card Width

SCALARS

Smoothing: 0.6

Tooltip sorting method: Alphabetical

☑ Ignore outliers in chart scaling

□ Partition non-monotonic X axis

HISTOGRAMS

Mode: Offset

IMAGES

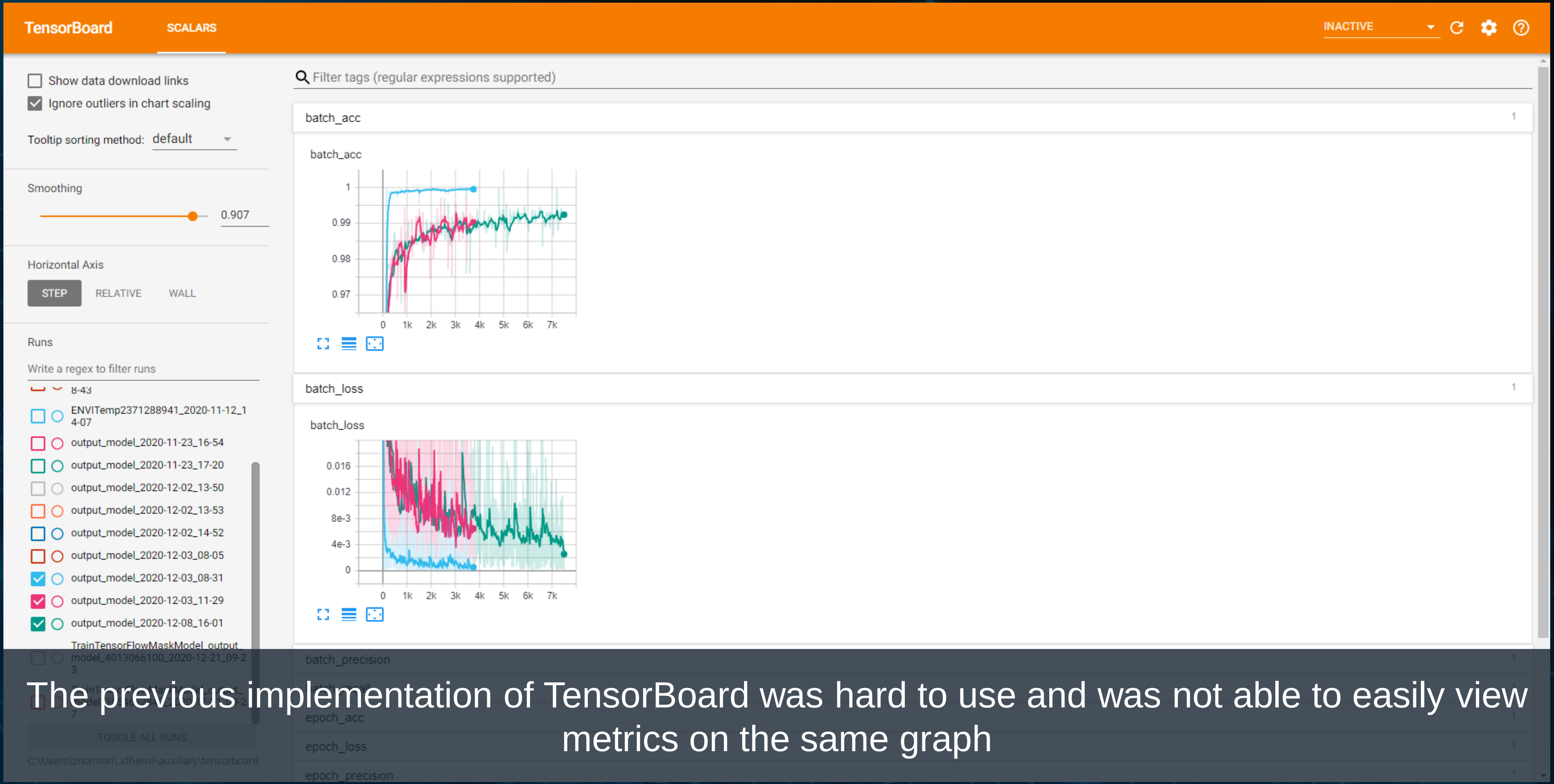
Brightness

Contrast

□ Show actual image size

Screenshot of TensorBoard in ENVI Deep Learning, showing some new metrics that are now reported

The Old Way



☐ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing: 0.907

Horizontal Axis: STEP | RELATIVE | WALL

Runs

Write a regex to filter runs

- 8-43
- ☐ ○ ENVITemp2371288941_2020-11-12_14-07
- ☐ ○ output_model_2020-11-23_16-54
- ☐ ○ output_model_2020-11-23_17-20
- ☐ ○ output_model_2020-12-02_13-50
- ☐ ○ output_model_2020-12-02_13-53
- ☐ ○ output_model_2020-12-02_14-52
- ☐ ○ output_model_2020-12-03_08-05
- ☑ ○ output_model_2020-12-03_08-31
- ☑ ○ output_model_2020-12-03_11-29
- ☑ ○ output_model_2020-12-08_16-01
- ☐ ○ TrainTensorFlowMaskModel_output_model_4013066100_2020-12-21_09-23

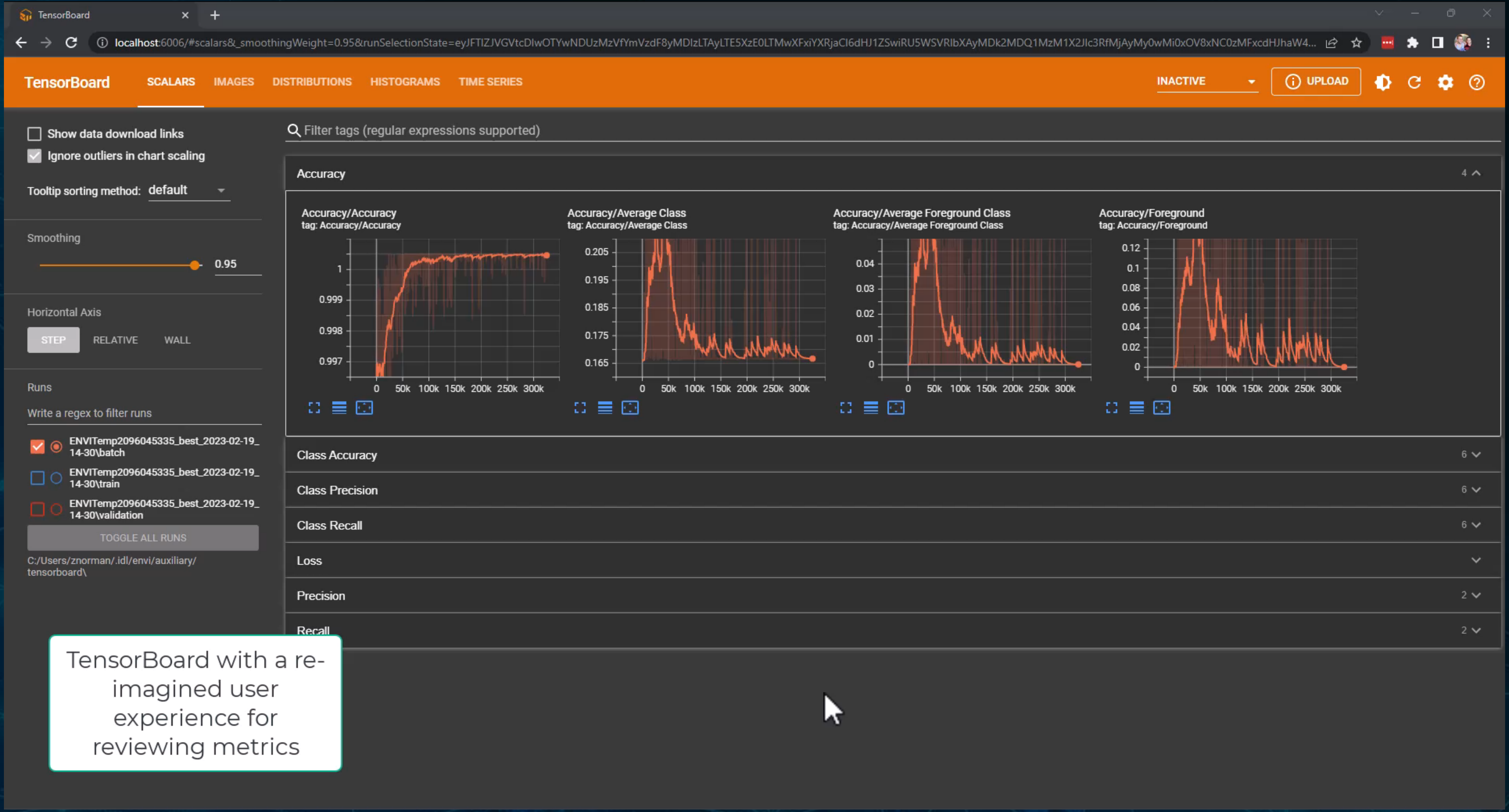
TOGGLE ALL RUNS

C:\Users\znorman\.idl\envi\auxiliary\tensorboard

Filter tags (regular expressions supported)

The previous implementation of TensorBoard was hard to use and was not able to easily view metrics on the same graph

The New Way



☐ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing 0.95

Horizontal Axis: STEP RELATIVE WALL

Runs

Write a regex to filter runs

☑ ENVITemp2096045335_best_2023-02-19_14-30\batch

☐ ENVITemp2096045335_best_2023-02-19_14-30\train

☐ ENVITemp2096045335_best_2023-02-19_14-30\validation

TOGGLE ALL RUNS

C:/Users/znorman/.idl/envi/auxiliary/tensorboard\

TensorBoard with a re-imagined user experience for reviewing metrics

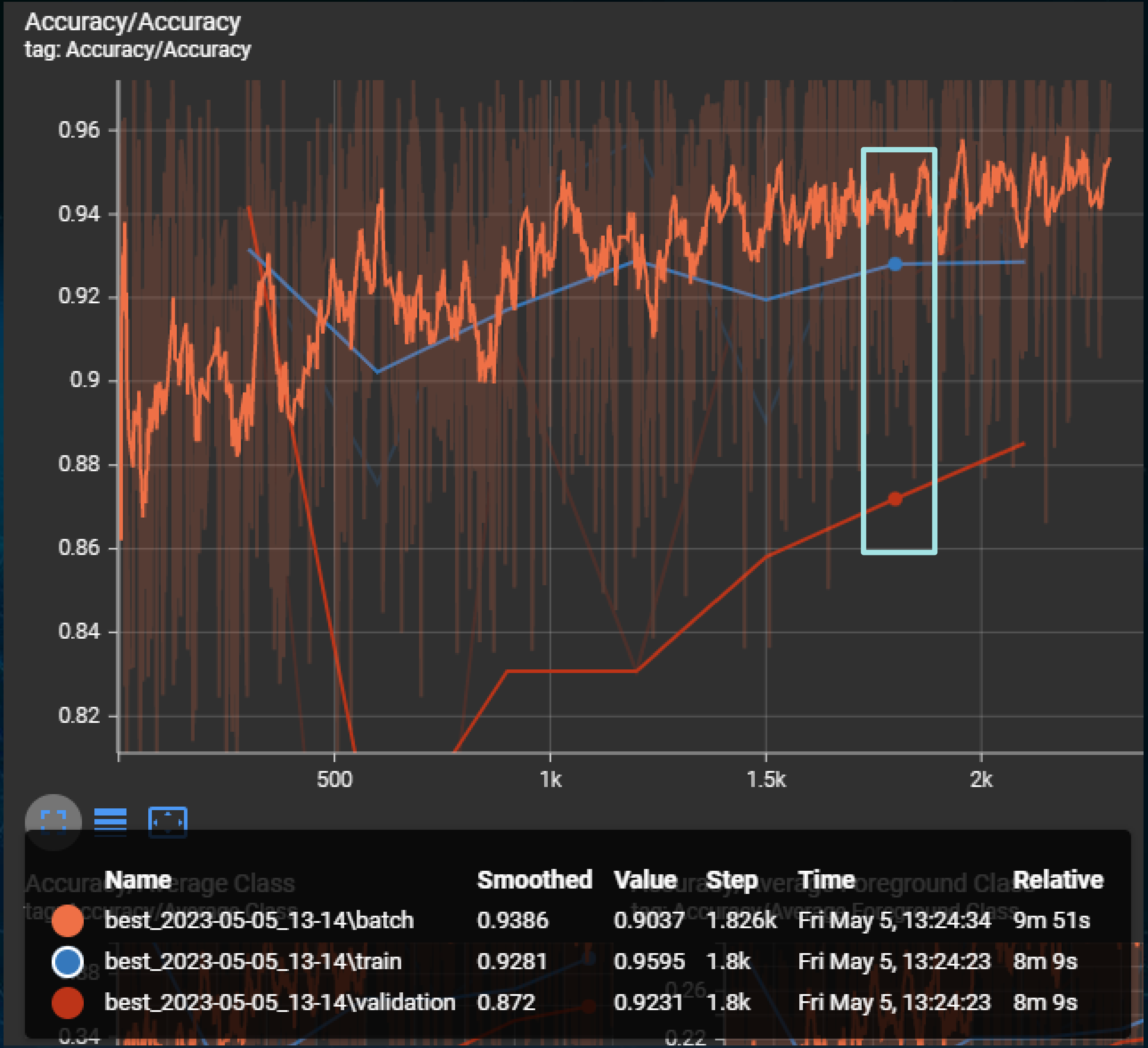
Notable Changes

- Visualize multiple metrics on one graph!
- X-axis represents number of samples that have been trained on
  - Allows comparisons between training sessions with different batch sizes on different GPUs
- As of ENVI Deep learning 2.0, we have had support for dark mode in TensorBoard

- Each training session get’s three TensorBoard logs:
  - “batch” which covers each step of training
    - Updates each step of training
  - “training” which represents the performance of the model on data used directly for training
    - Once per epoch
  - “validation” which represents the performance of the model on data used for validation (i.e. evaluation)
    - Once per epoch

Overall Metrics

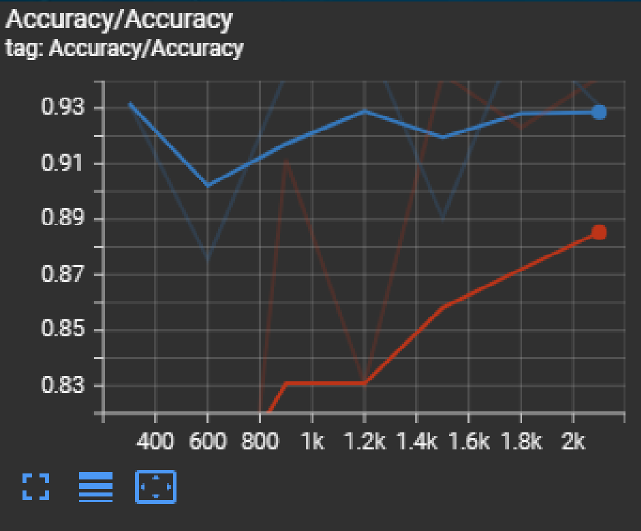
- Accuracy
  - How well the model re-creates its input
  - Higher is better
- Precision
  - Measures the accuracy of positive predictions
  - Higher is better
- Recall
  - Measures the completeness of positive predictions
  - Higher is better
- Loss
  - The overall performance of the model
  - Lower is better



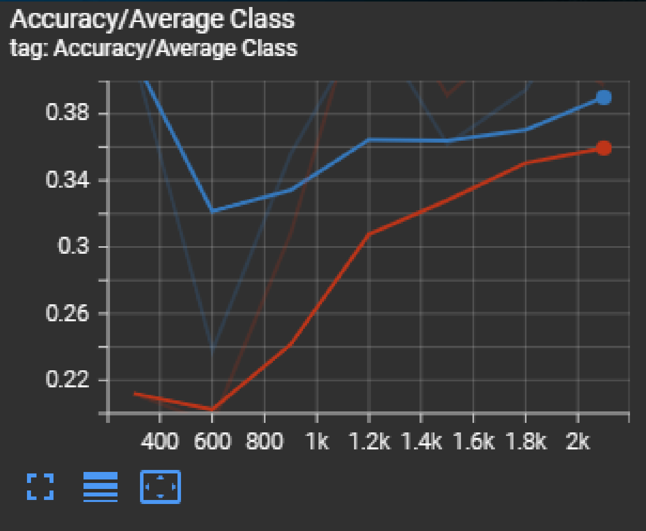
	Name	Smoothed	Value	Step	Time	Relative
●	best_2023-05-05_13-14\batch	0.9386	0.9037	1.826k	Fri May 5, 13:24:34	9m 51s
○	best_2023-05-05_13-14\train	0.9281	0.9595	1.8k	Fri May 5, 13:24:23	8m 9s
●	best_2023-05-05_13-14\validation	0.872	0.9231	1.8k	Fri May 5, 13:24:23	8m 9s

A look at overall accuracy being reported in TensorBoard which allows you to view accuracy during each step, for each epoch, and for validation data in the same plot.

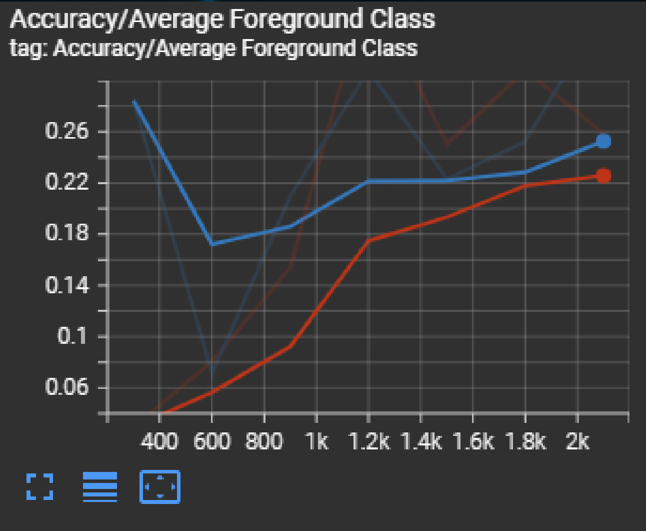
Types of Accuracy



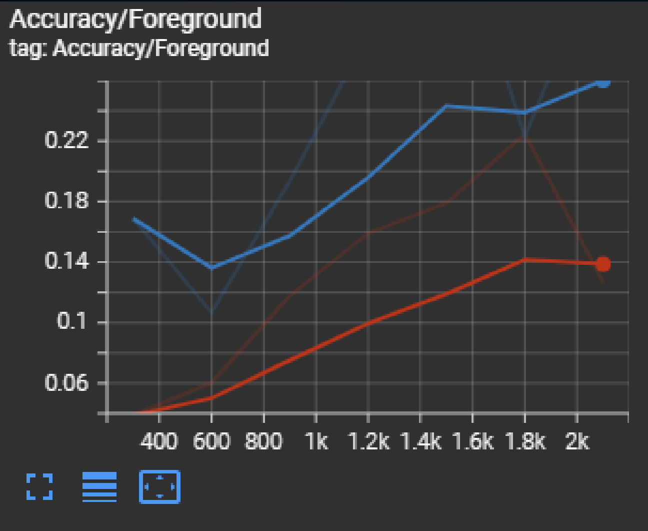
Accuracy/Accuracy
tag: Accuracy/Accuracy



Accuracy/Average Class
tag: Accuracy/Average Class



Accuracy/Average Foreground Class
tag: Accuracy/Average Foreground Class



Accuracy/Foreground
tag: Accuracy/Foreground

- Accuracy
  - Overall, traditional value

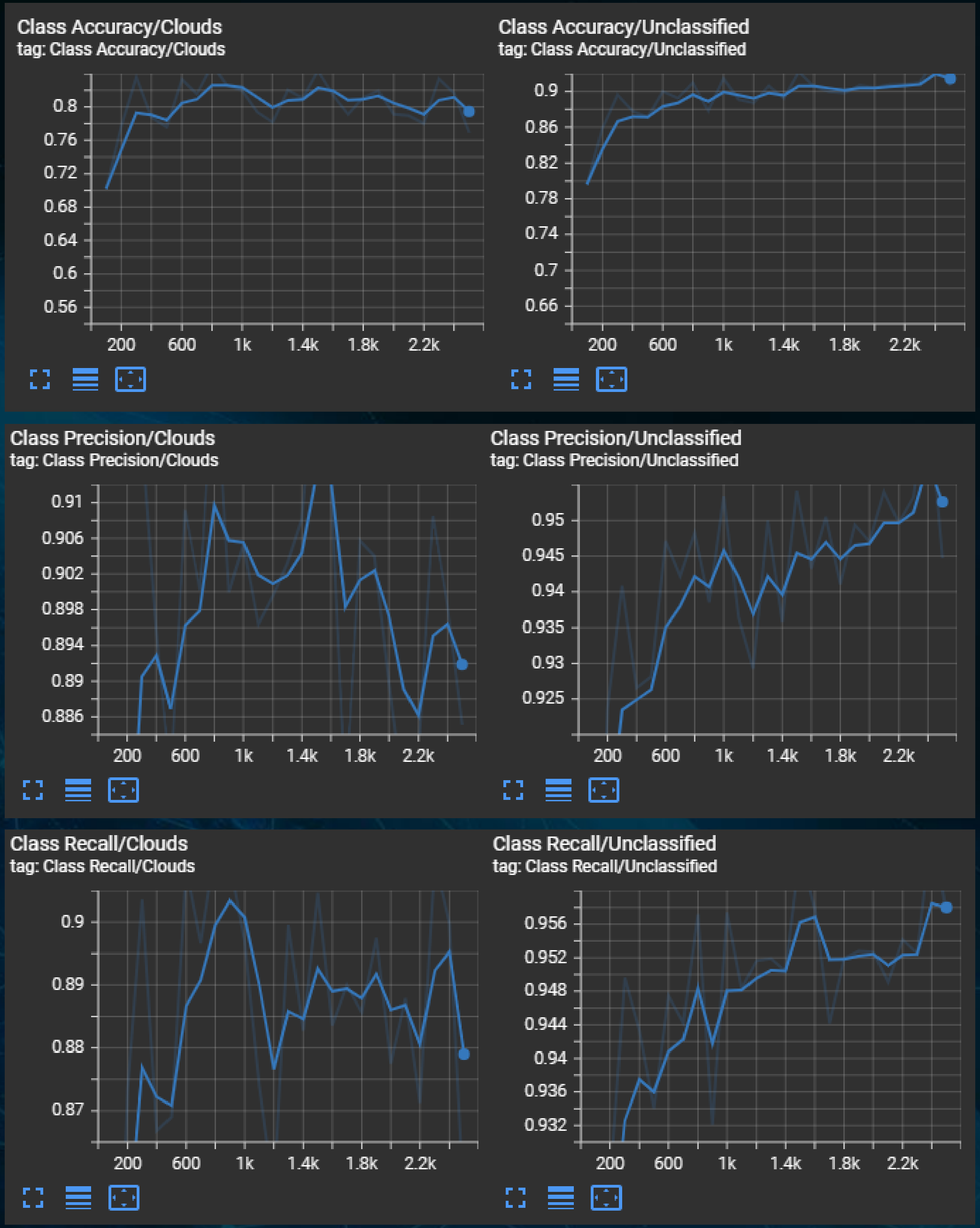
- Average class
  - Accuracy of each individual class, averaged together

- Average foreground class
  - Accuracy of each foreground classed, averaged together (i.e. features you are interested in and have labeled)
  - Useful for rare or small features when background outnumbers foreground

- Foreground
  - Accuracy for non-background features (i.e. features you are interested in and have labeled)
  - Useful for rare or small features when background outnumbers foreground

Per Class Metrics

- At the end of each epoch of training, we now report the accuracy, precision, and recall of each class
- You can view this for training and validation, but not for each batch
- Depending on how rare your features are, and how much training data you have, these values may jump around a bit

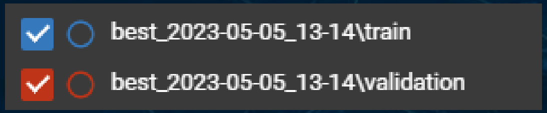


Multi-Class Class Metrics

Example of accuracy, precision, and recall for foreground classes

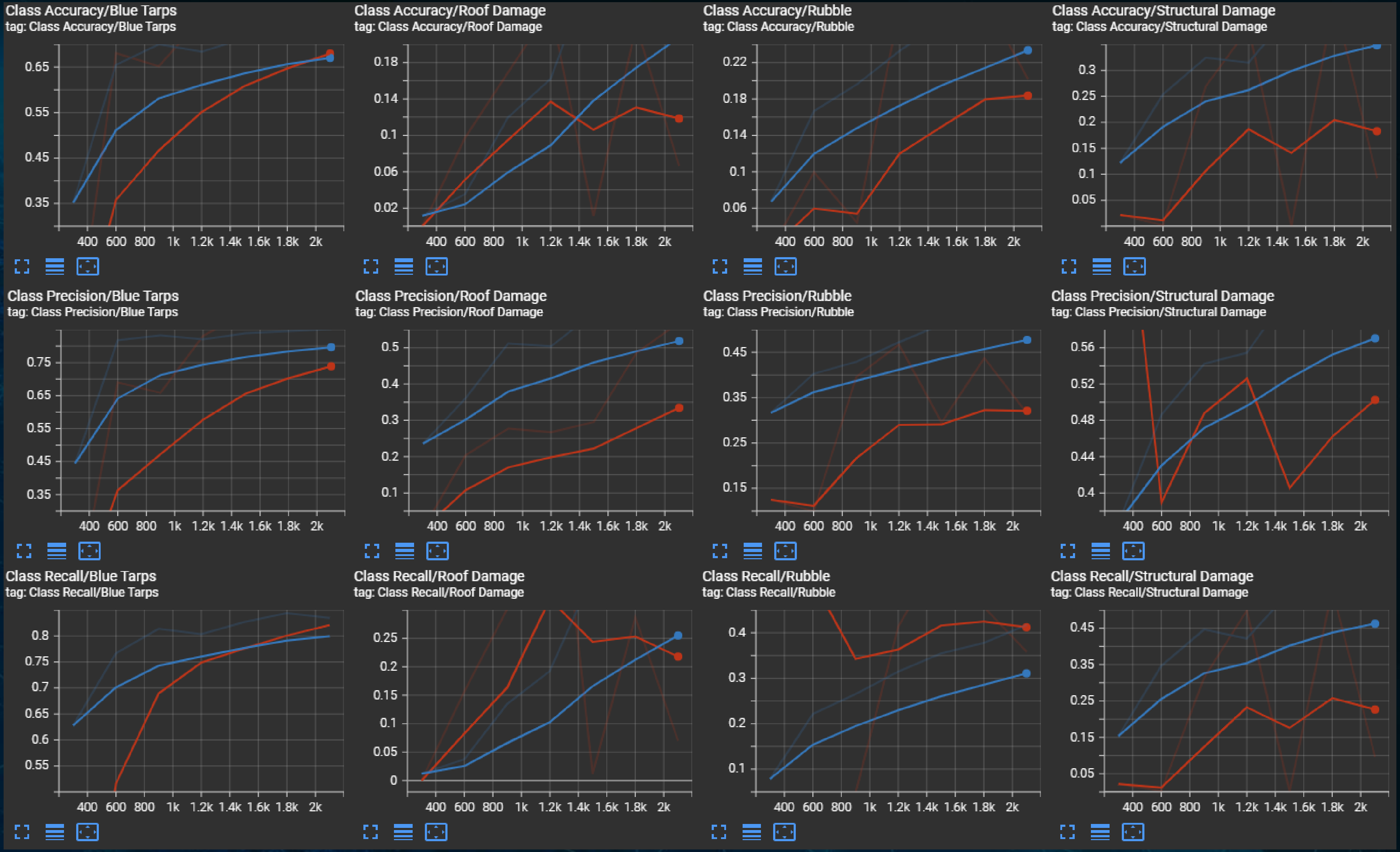
Blue represents training and **red** is validation

Here's the legend you'll see in TensorBoard

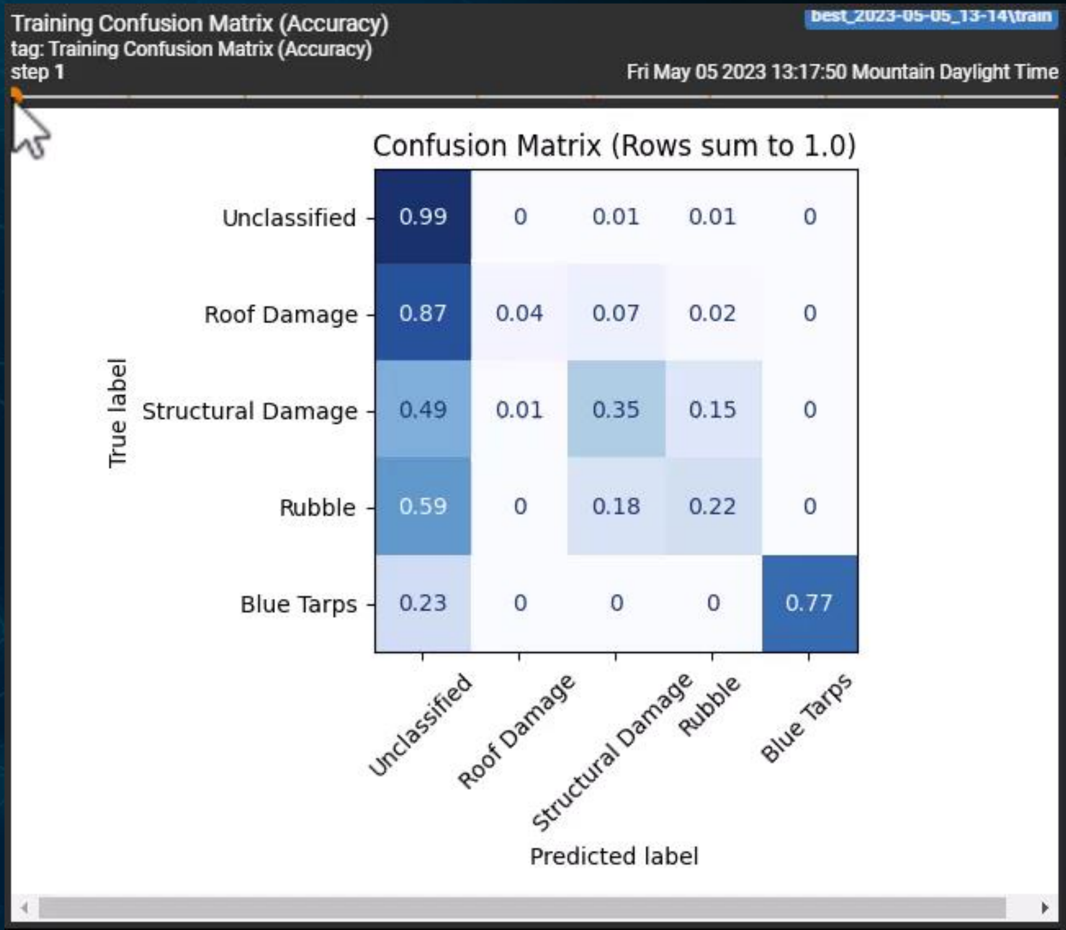


☑ ○ best_2023-05-05_13-14\train

☑ ○ best_2023-05-05_13-14\validation



Confusion Matrices



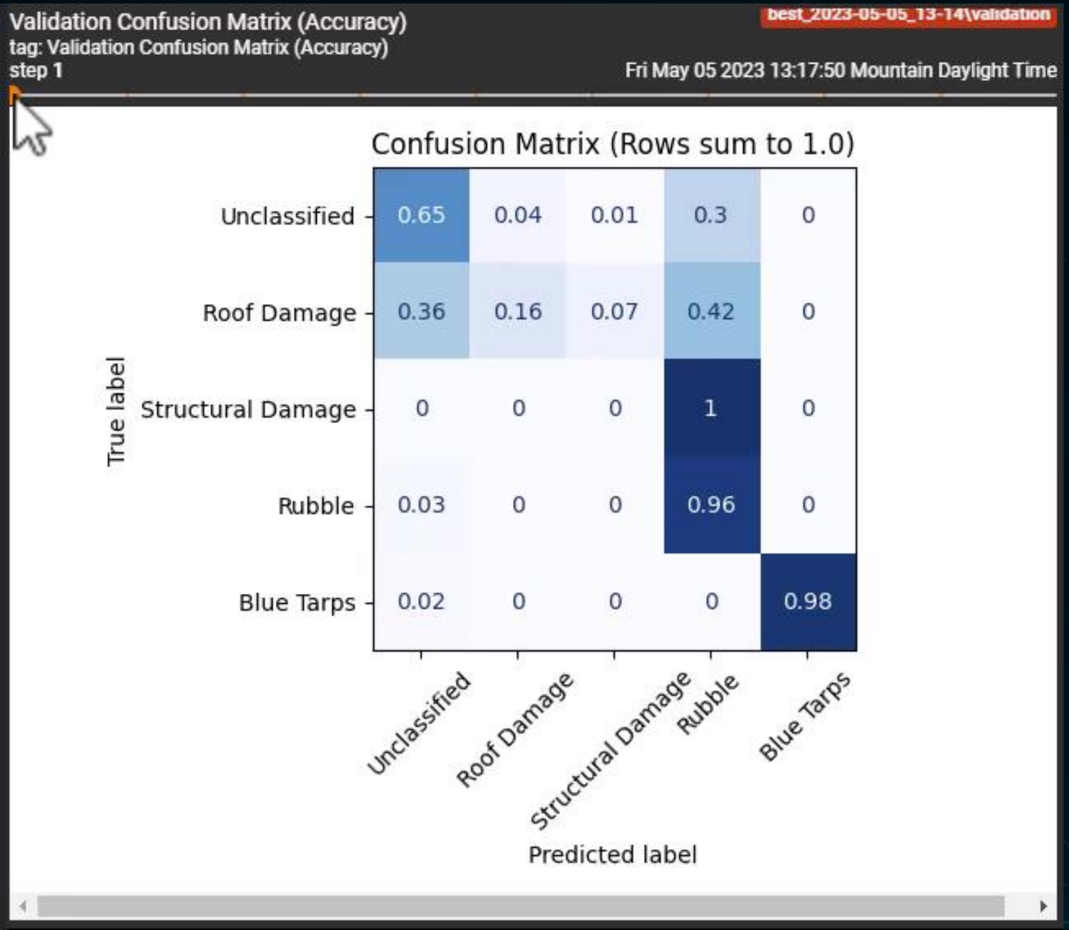
Training Confusion Matrix (Accuracy)
tag: Training Confusion Matrix (Accuracy)
step 1

best_2023-05-05_13-14\train

Fri May 05 2023 13:17:50 Mountain Daylight Time

Confusion Matrix (Rows sum to 1.0)

True label \ Predicted label	Unclassified	Roof Damage	Structural Damage	Rubble	Blue Tarps
Unclassified	0.99	0	0.01	0.01	0
Roof Damage	0.87	0.04	0.07	0.02	0
Structural Damage	0.49	0.01	0.35	0.15	0
Rubble	0.59	0	0.18	0.22	0
Blue Tarps	0.23	0	0	0	0.77



Validation Confusion Matrix (Accuracy)
tag: Validation Confusion Matrix (Accuracy)
step 1

best_2023-05-05_13-14\validation

Fri May 05 2023 13:17:50 Mountain Daylight Time

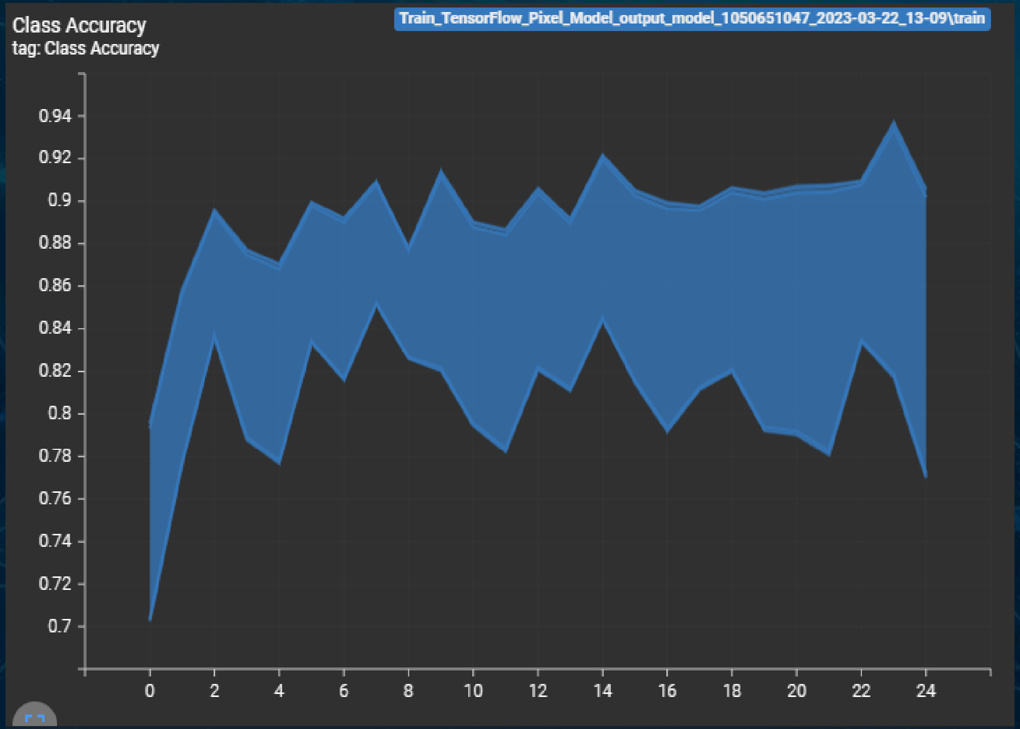
Confusion Matrix (Rows sum to 1.0)

True label \ Predicted label	Unclassified	Roof Damage	Structural Damage	Rubble	Blue Tarps
Unclassified	0.65	0.04	0.01	0.3	0
Roof Damage	0.36	0.16	0.07	0.42	0
Structural Damage	0	0	0	1	0
Rubble	0.03	0	0	0.96	0
Blue Tarps	0.02	0	0	0	0.98

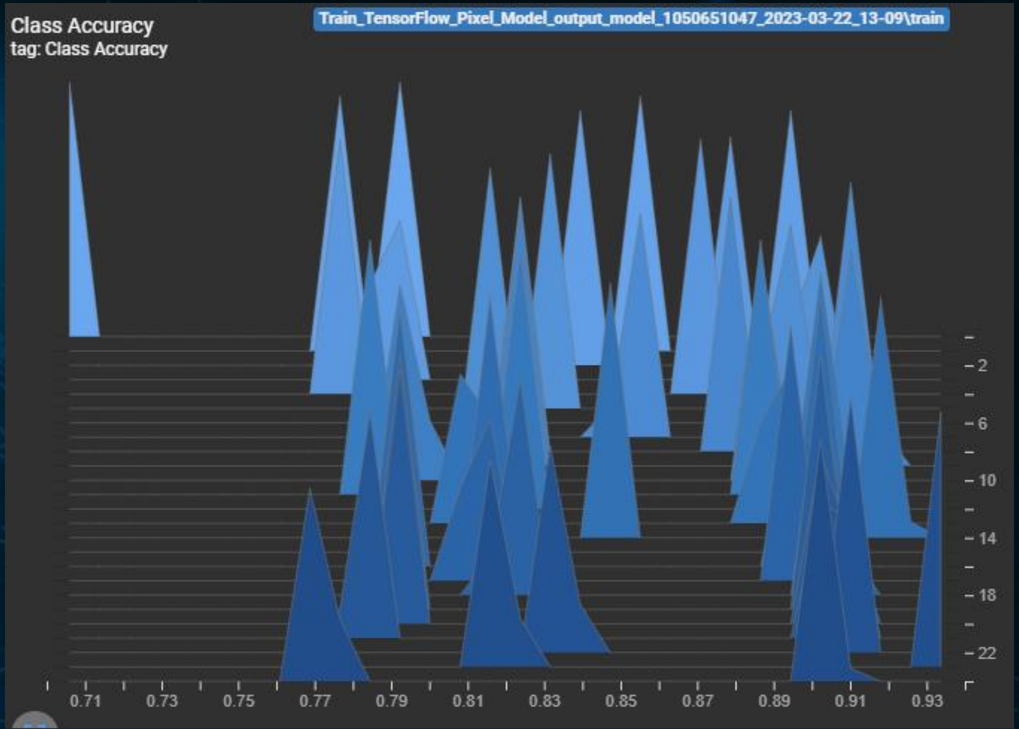
After each epoch, we generate confusion matrices for training and validation. For readability, the rows are normalized from 0 to 1 (you don't get raw class counts)

Distributions and Histograms

Two new styles of metrics: distributions and histograms give insights into the range of values for a metric in a single visual



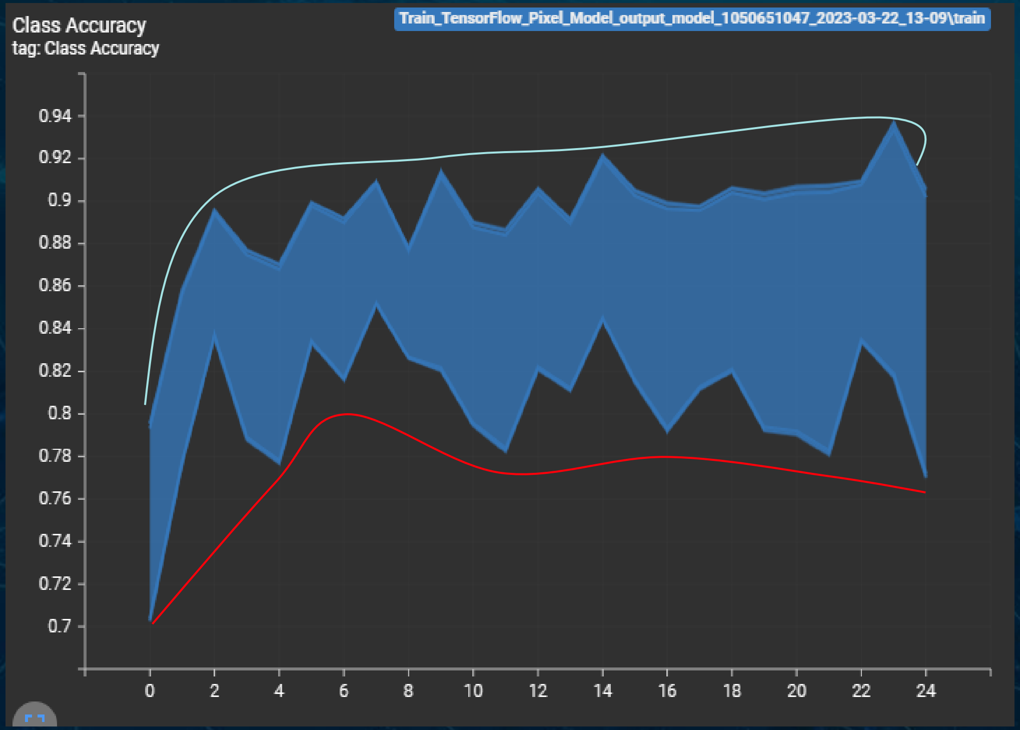
Distributions show the range of values for a given metric

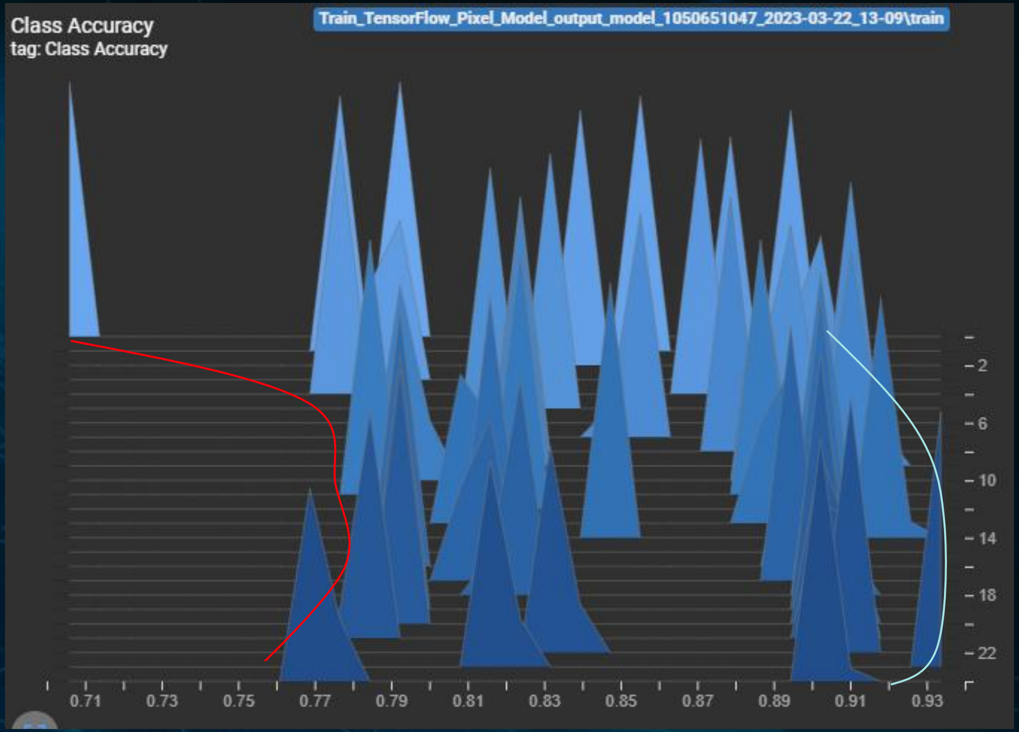


Histograms show the range of values for a metric along with the distribution of values

Distributions and Histograms

Each plot shows the similar information, as shown by the red and light blue lines in the graphs below, corresponding to extents of the values



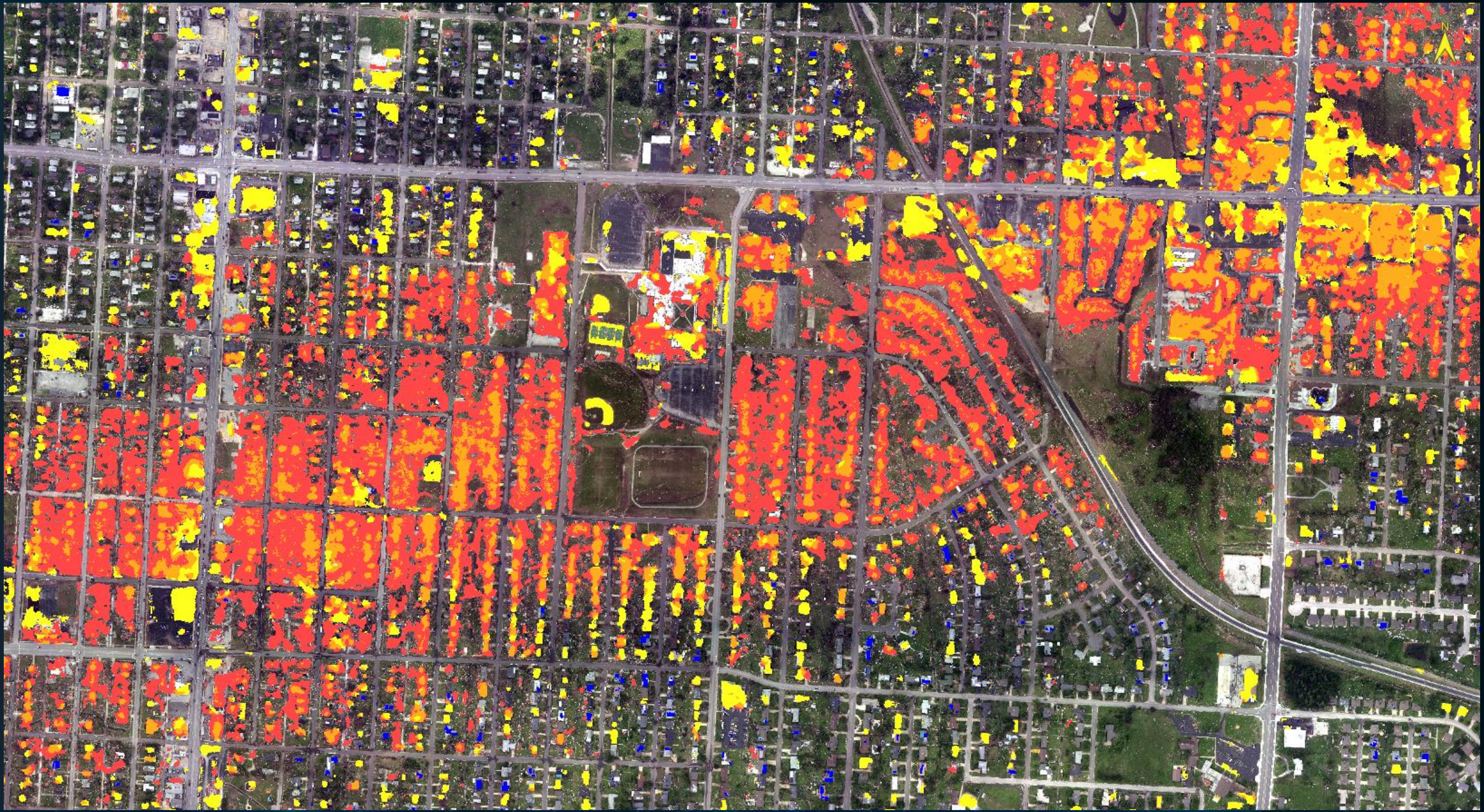


TensorBoard: Metrics vs Real-World Performance

With TensorBoard, you might be surprised by what gets reported for the accuracy, precision, or recall of different classes

Zach’s pro tip is to always process an image and have a look at the results yourself

Also keep in mind that we can use additional post-processing to clean up our results



Reminder: Cleanup Tools

RASTER

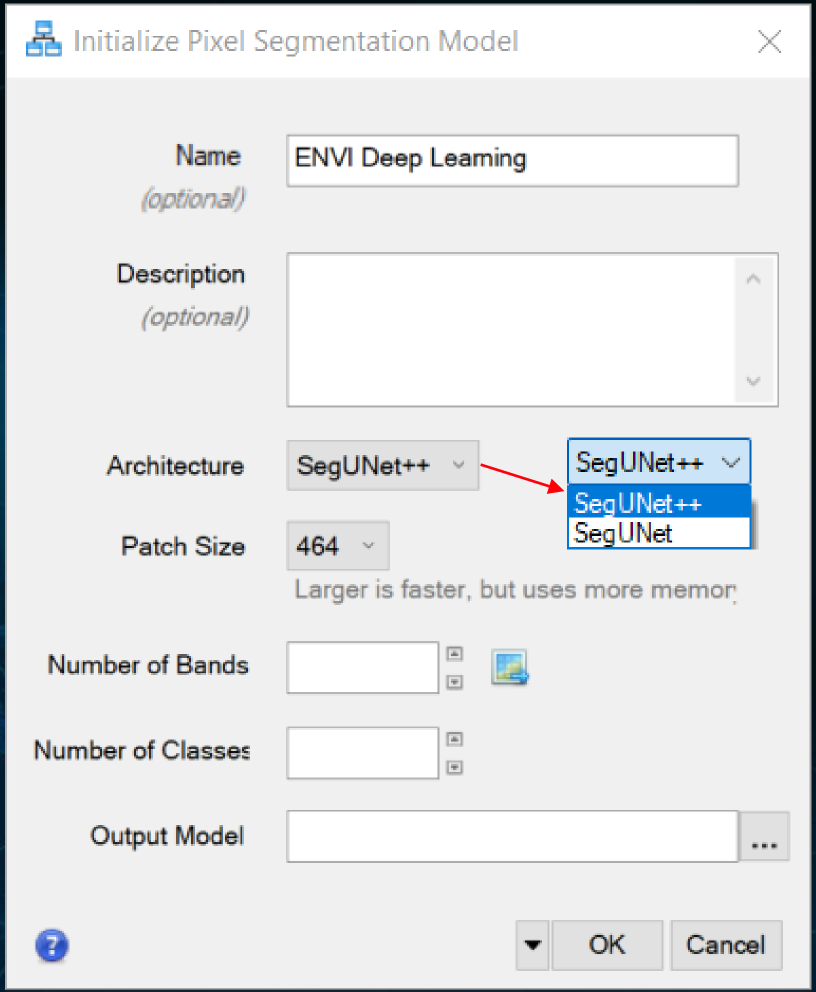
- **Classification Aggregation**
  - This task aggregates smaller class regions to a larger, adjacent region as part of the classification cleanup.
- **Classification Clumping**
  - This task performs a clumping method on a classification image. This operation clumps adjacent similar classified areas using morphological operators.
- **Classification Sieving**
  - This task removes isolated classified pixels using blob grouping.
- **Classification Smoothing**
  - This task removes speckling noise from a classification image. It uses majority analysis to change spurious pixels within a large single class to that class.
- **Convert Classification to Vector Shapefile**
  - Converts a classification image to a shapefile
- **Class Activation to Polyline Shapefile**
  - Converts a linear mask of features to a polyline, where the polyline represents the centerline (i.e polygon of roads converted to the centerline of roads)

VECTOR

- **Filter Vector**
  - Remove geometry from a vector based on area (pixels and normal units such as meters
- **Smooth Vector**
  - Reduce the number of vertices in a vector
- **Vector Record to Bounding Box**
  - Convert polygons to a bounding box (extent-based or oriented with the shape of the features)
- **Vector Records to Centroid**
  - Convert polygons to a centroid
- **Postprocess Classification Vector**
  - Takes bounding boxes from object detection and allows you to do additional filtering to cleanup results

New Architecture: Pixel Segmentation

- Creating a model for segmentation now includes two architectures:
  - SegUNet++
  - SegUNet
- Differences with the new model:
  - Better ability to detect features with smaller training sessions
  - Generates less noise and fewer spurious (i.e. standalone) pixels
  - About 10% faster
  - Slightly smaller network size



Initialize Pixel Segmentation Model

Name *(optional)*: ENVI Deep Learning

Description *(optional)*:

Architecture: SegUNet++

Patch Size: 464

Larger is faster, but uses more memor

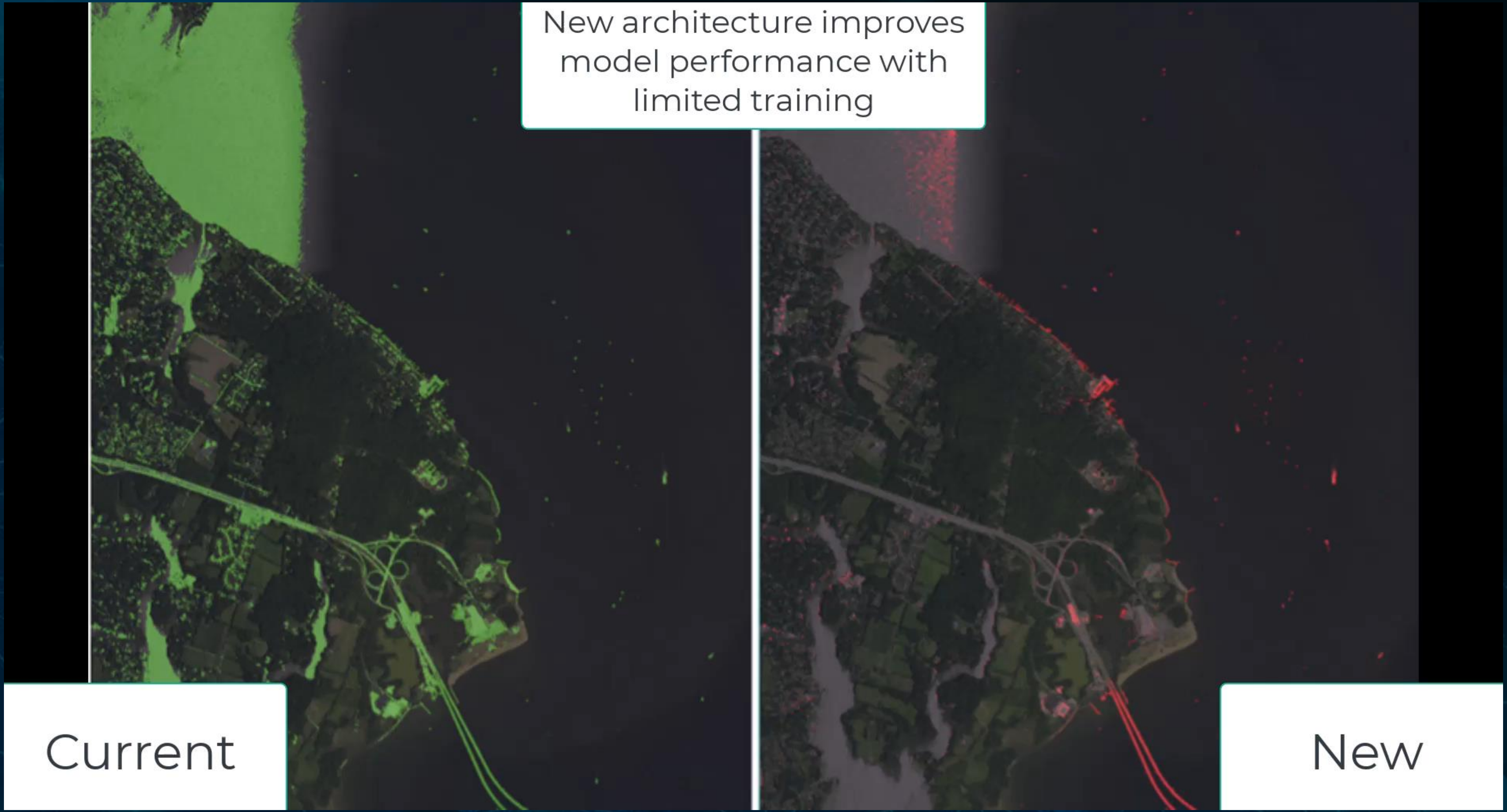
Number of Bands:

Number of Classes:

Output Model:

Architecture selection in the model initialization dialog

Model Comparison: Limited Training

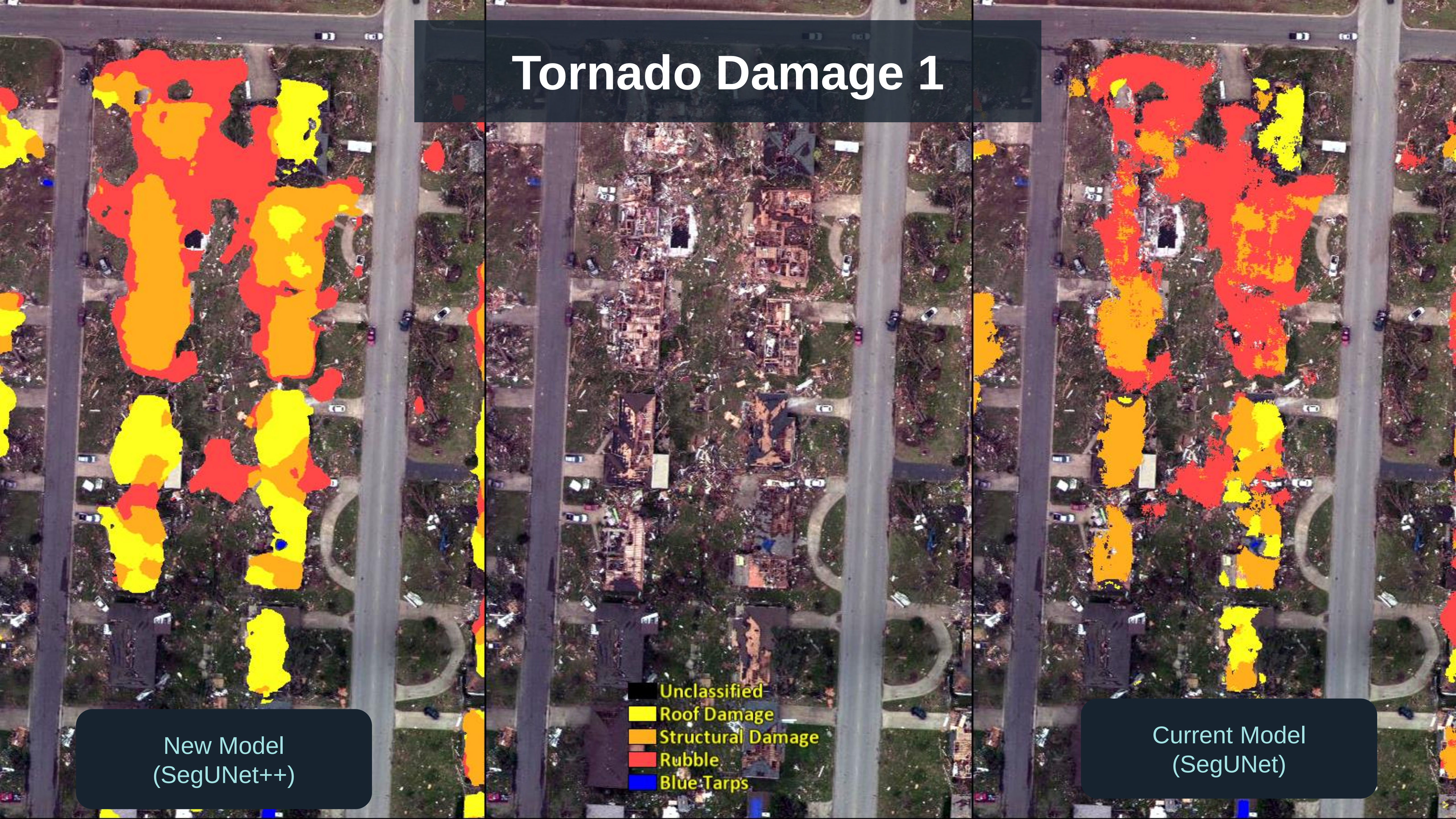


New architecture improves model performance with limited training

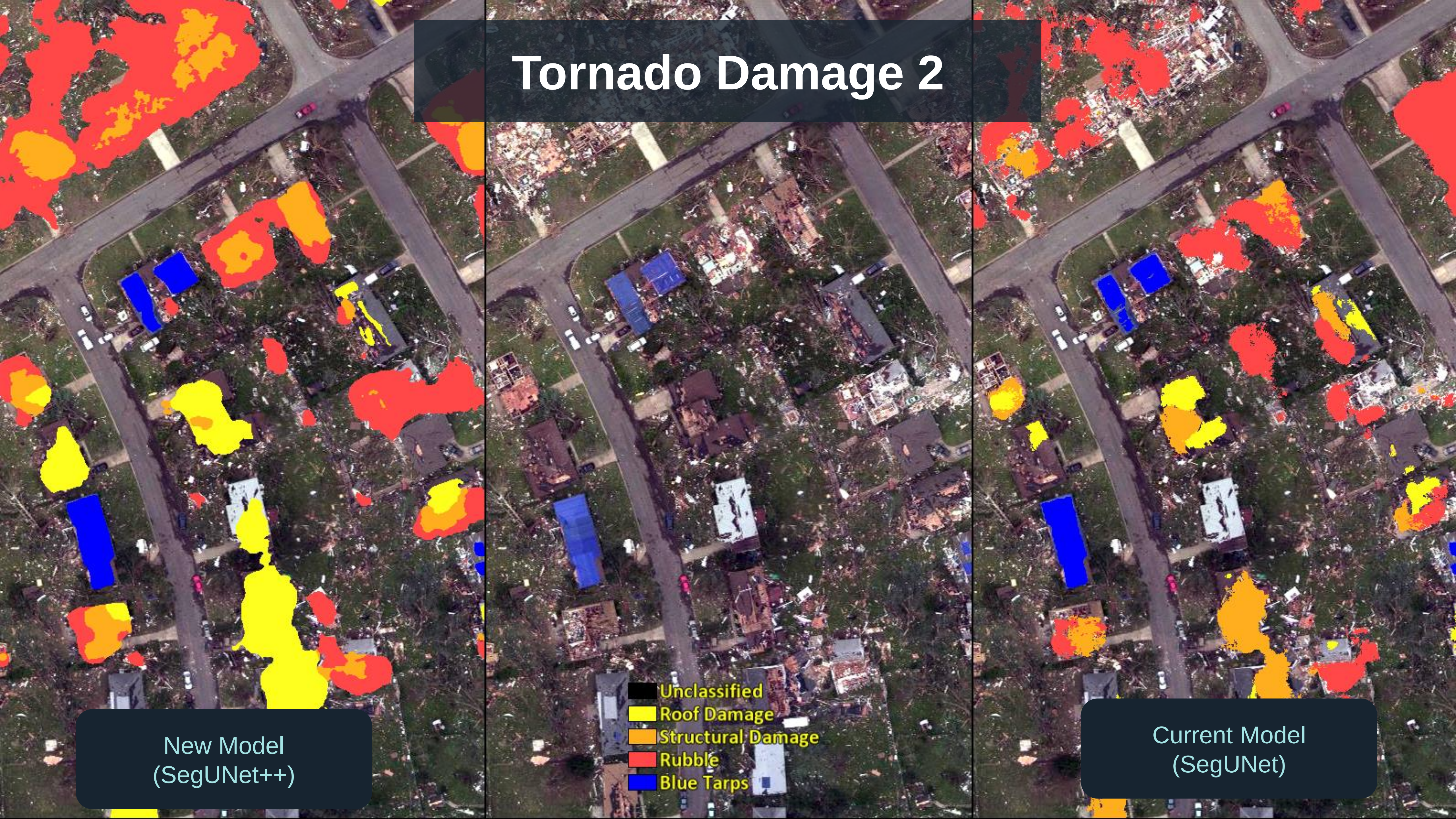
Current

New

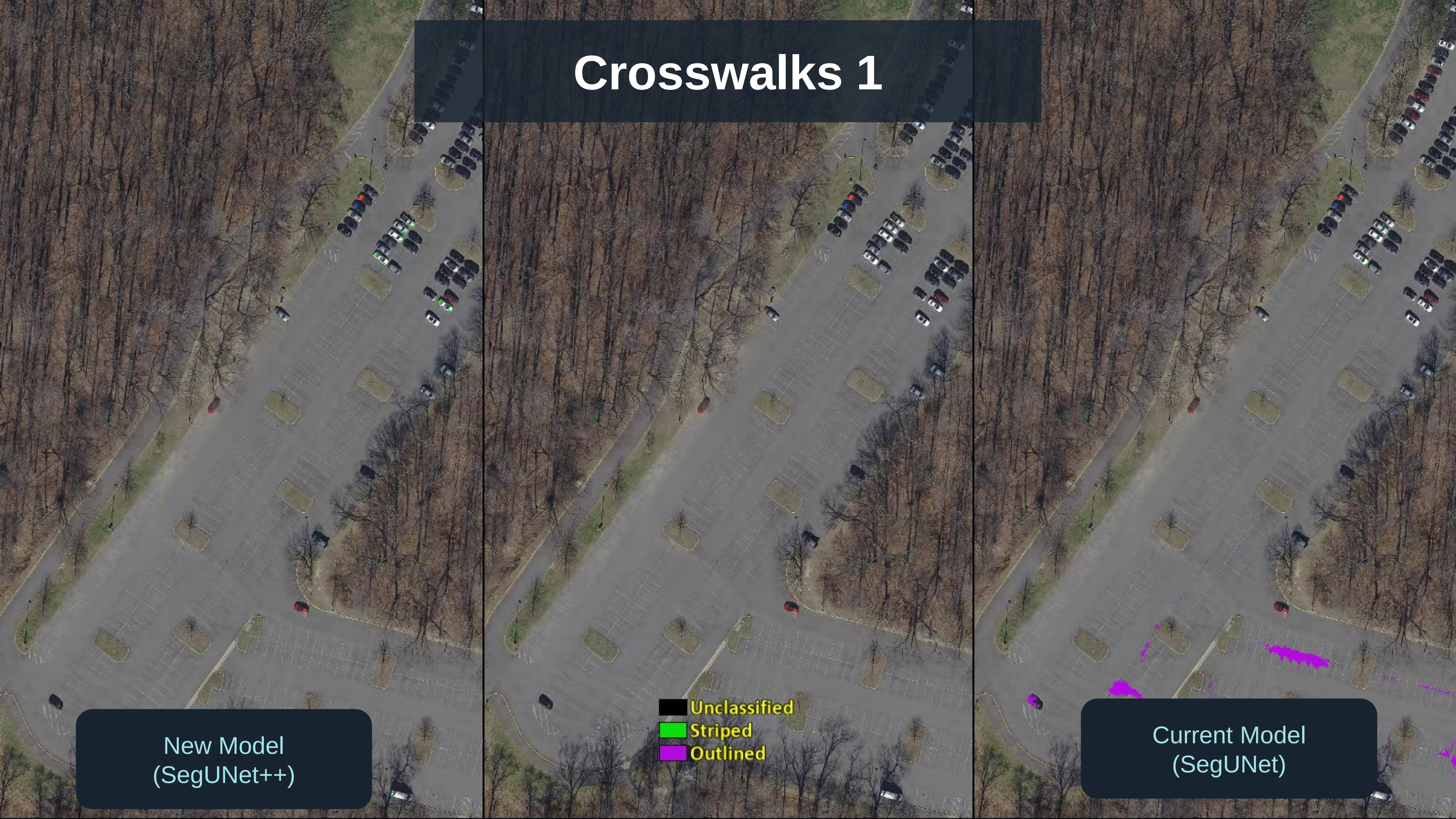
Tornado Damage 1



Tornado Damage 2



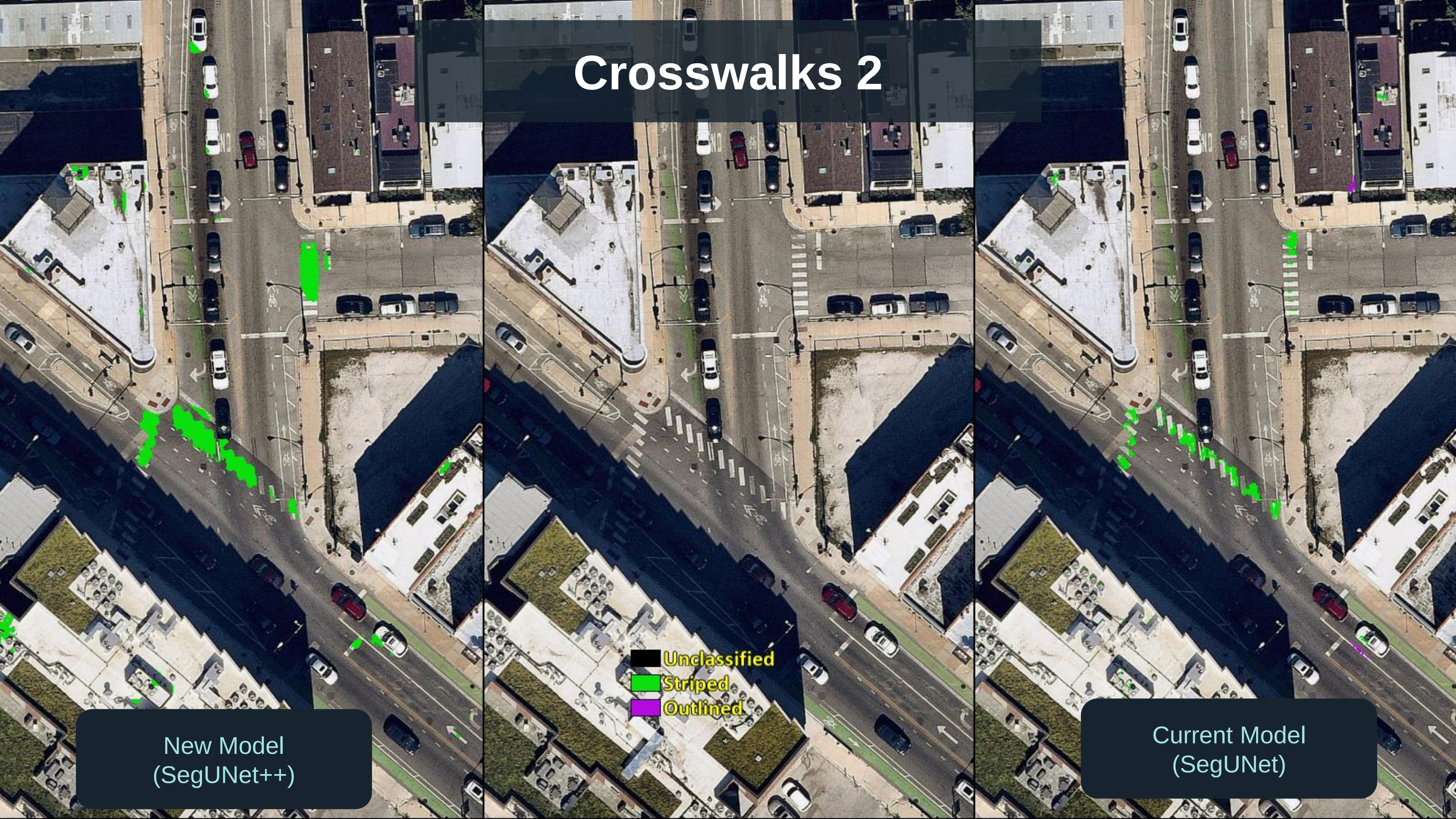
Crosswalks 1



New Model
(SegUNet++)

Current Model
(SegUNet)

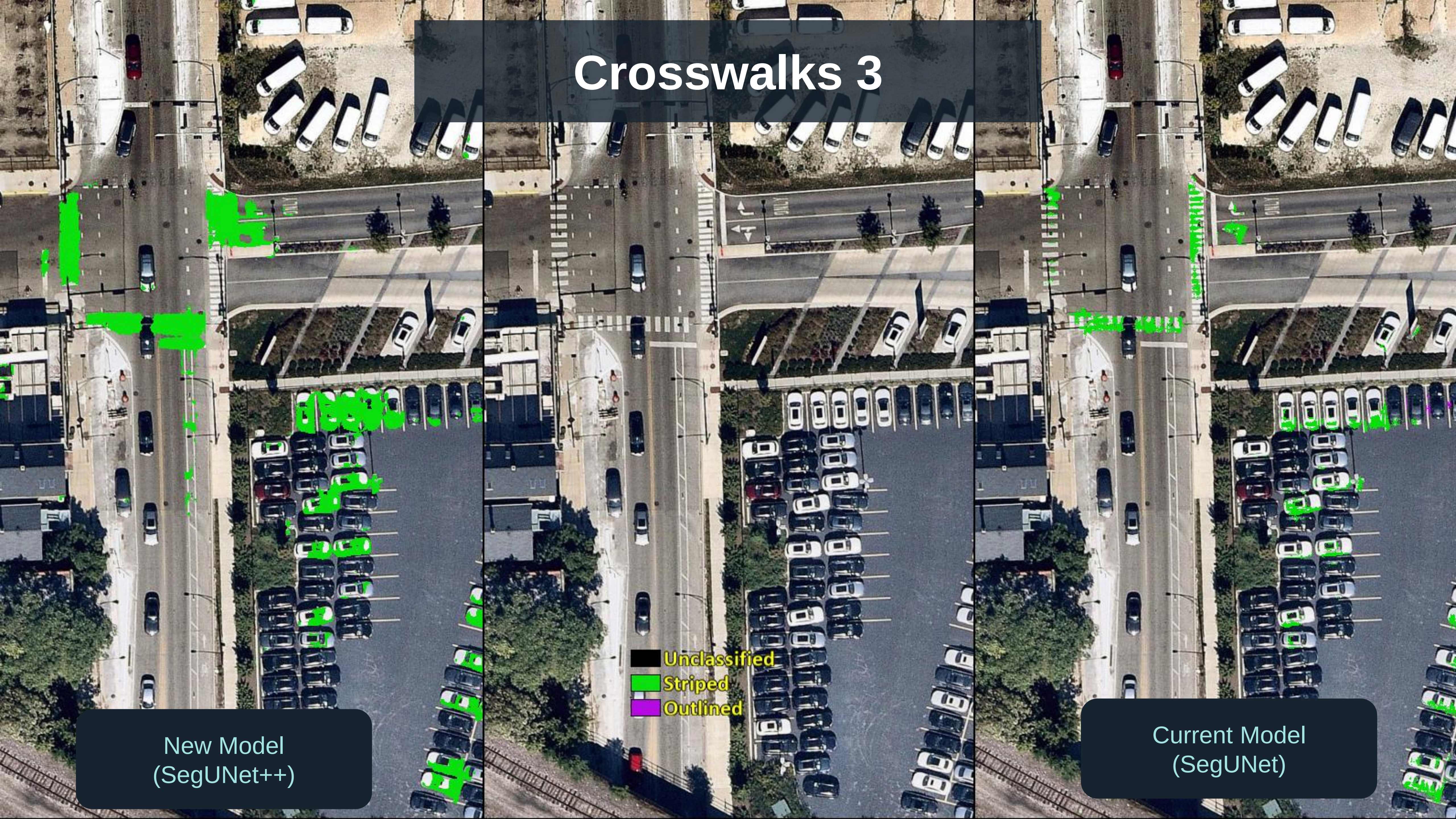
Crosswalks 2



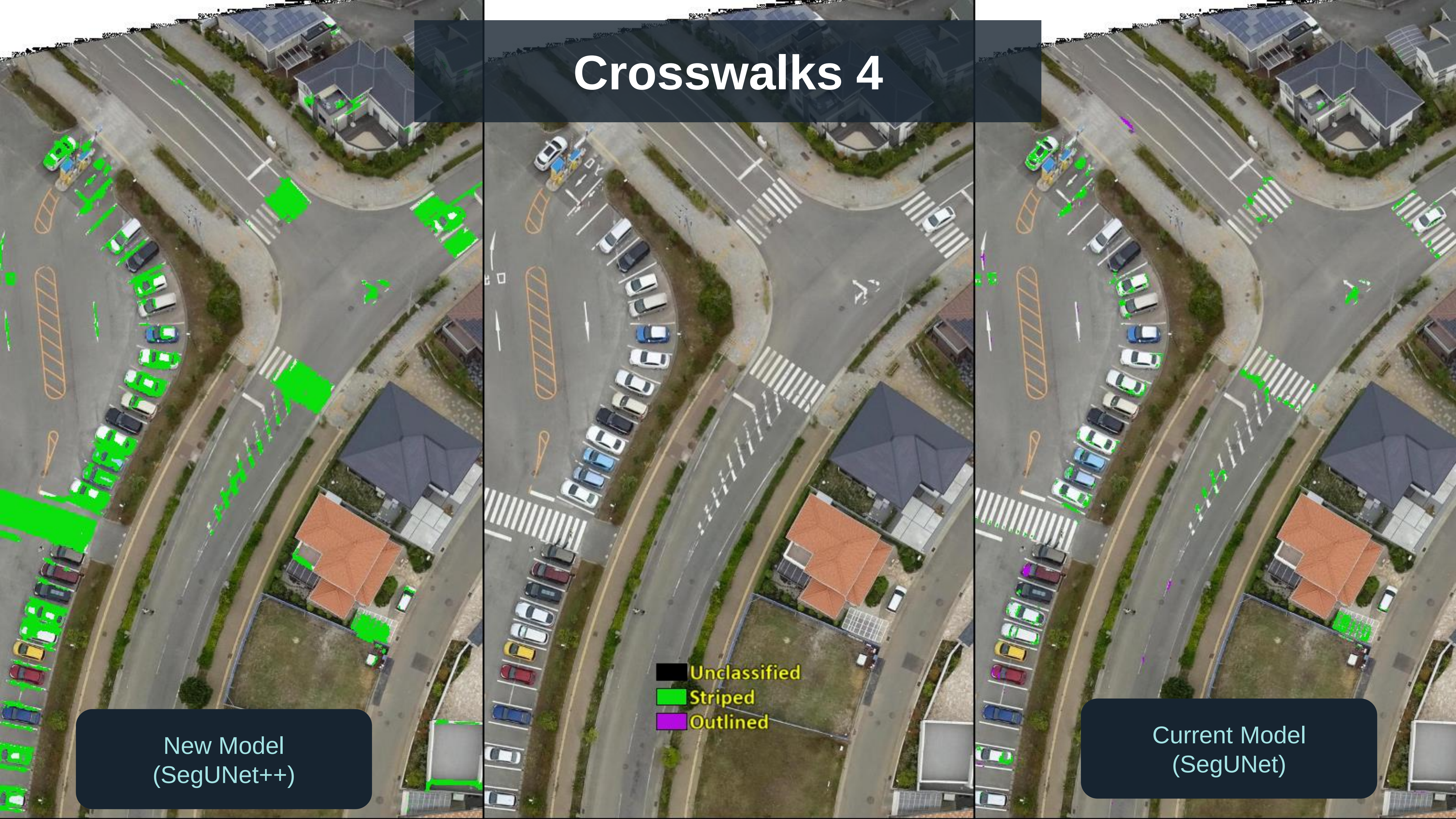
New Model
(SegUNet++)

Current Model
(SegUNet)

Crosswalks 3



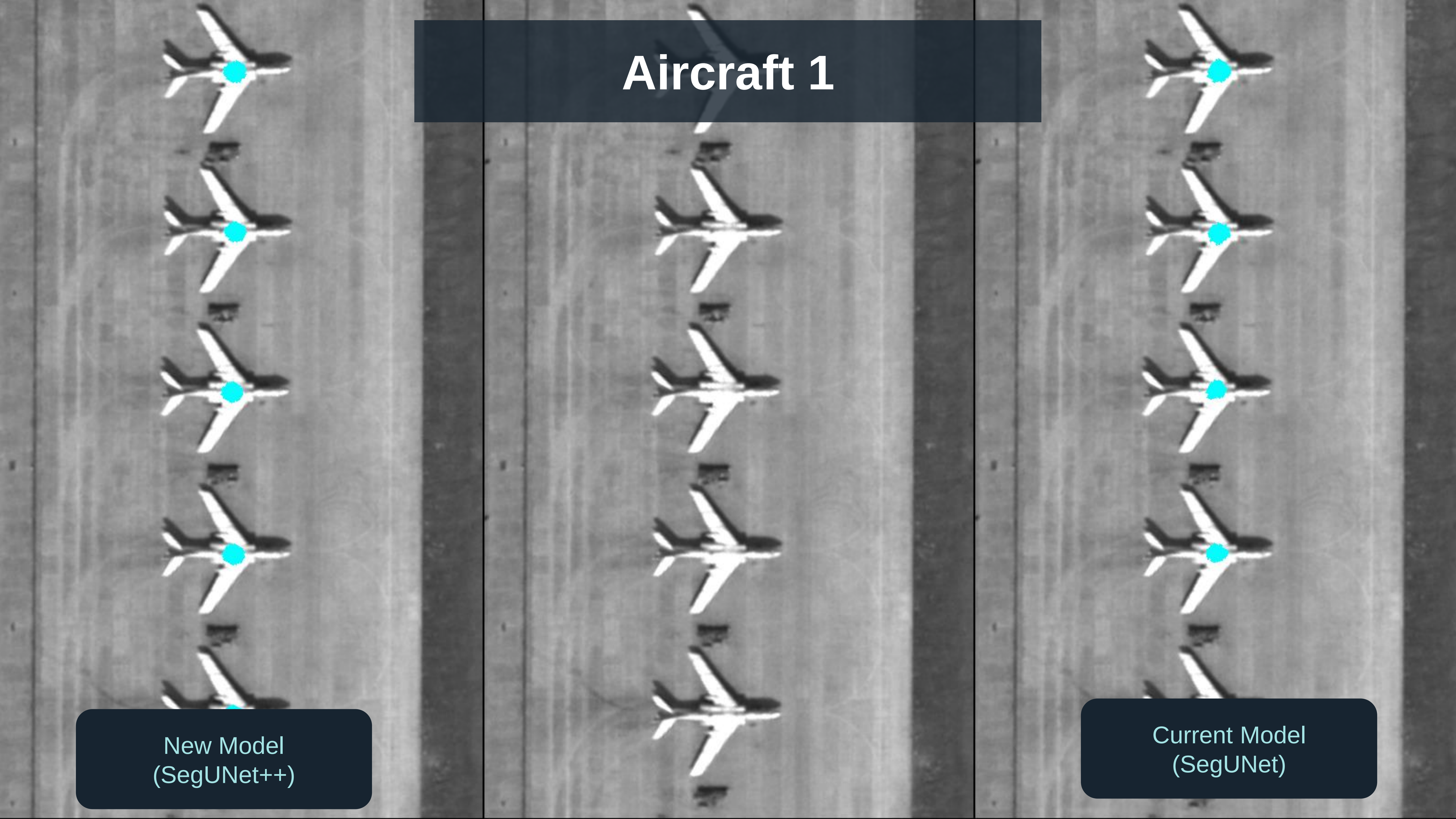
Crosswalks 4



New Model
(SegUNet++)

Current Model
(SegUNet)

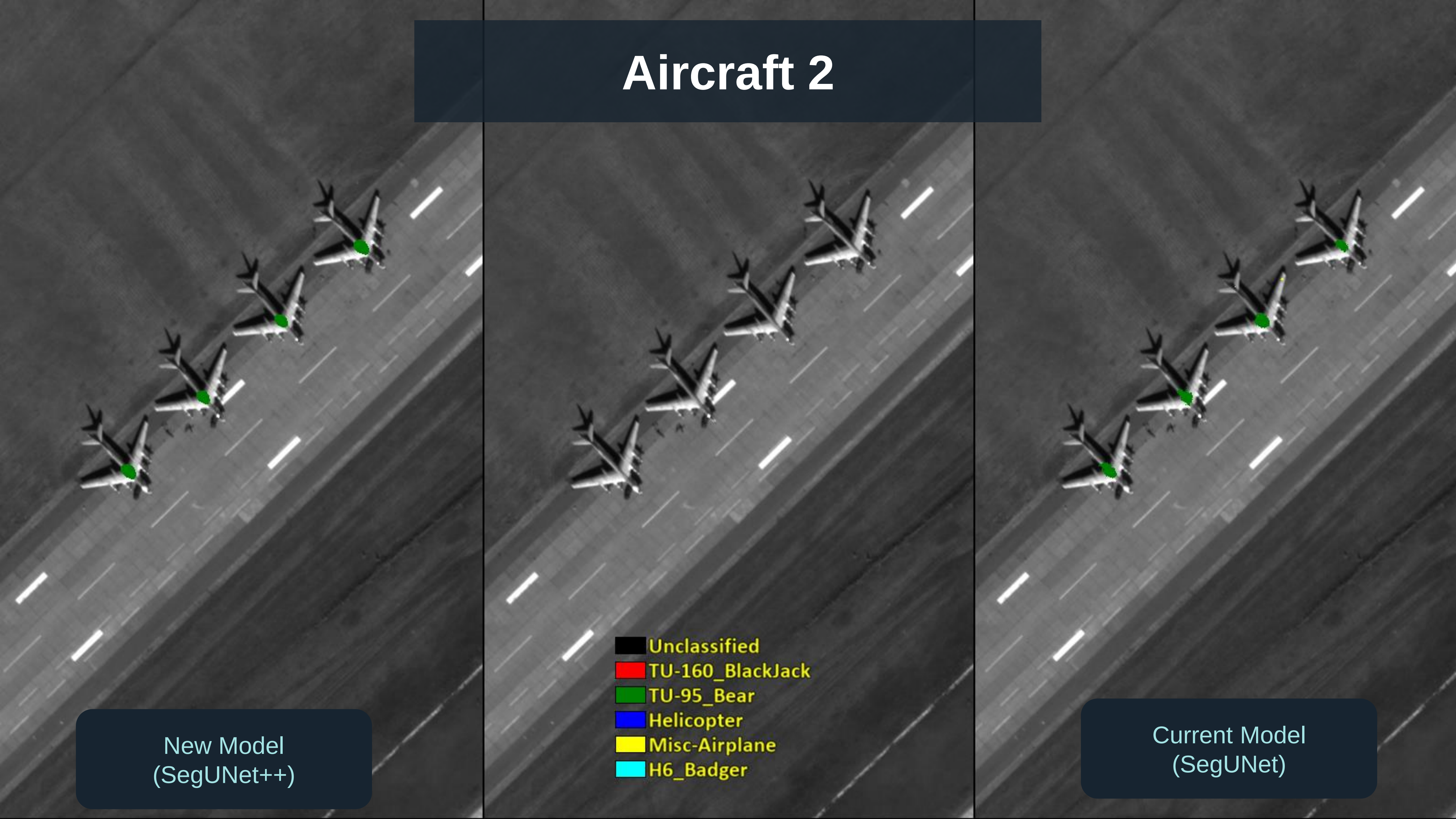
Aircraft 1



New Model
(SegUNet++)

Current Model
(SegUNet)

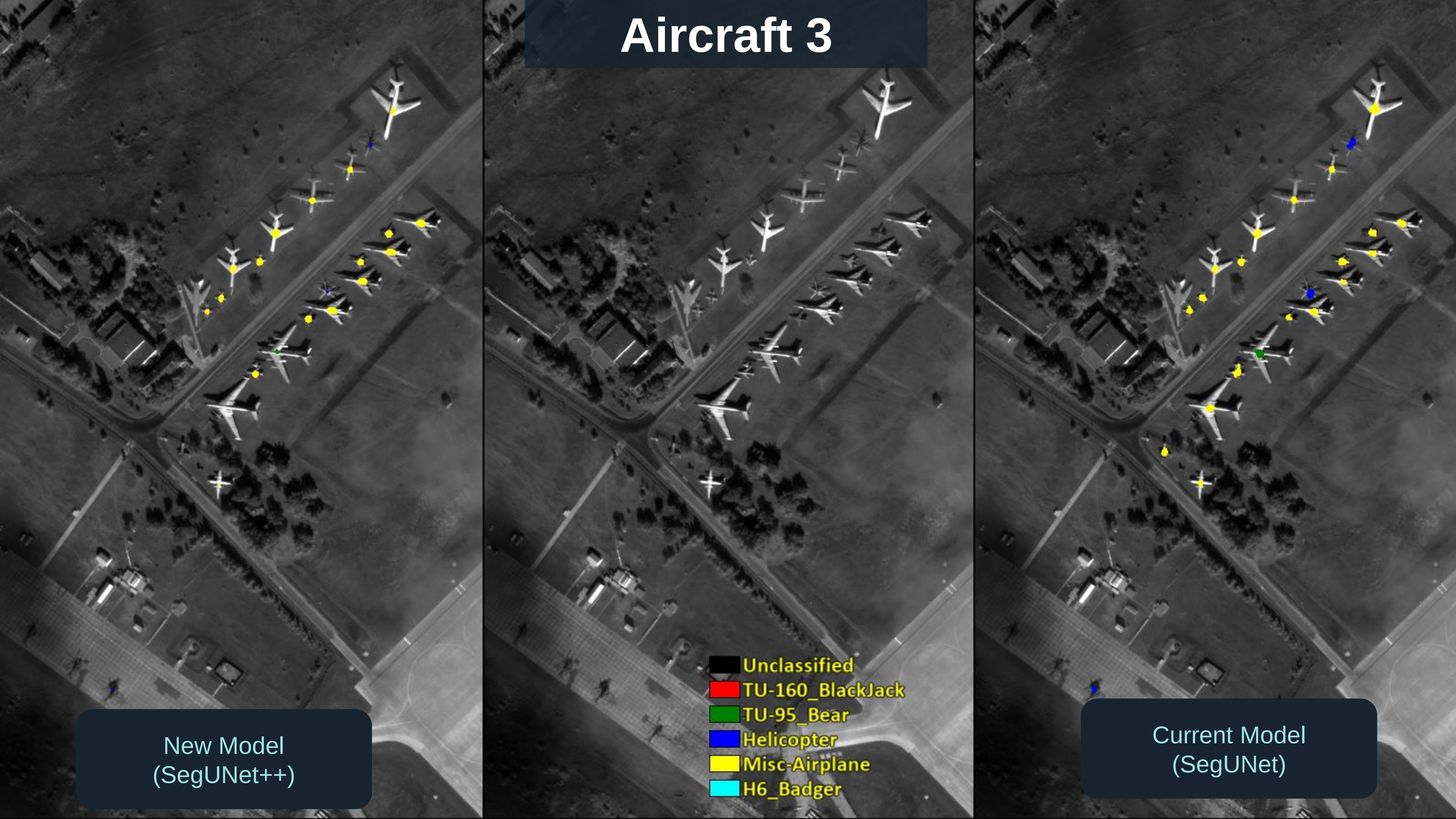
Aircraft 2



New Model
(SegUNet++)

Current Model
(SegUNet)

Aircraft 3

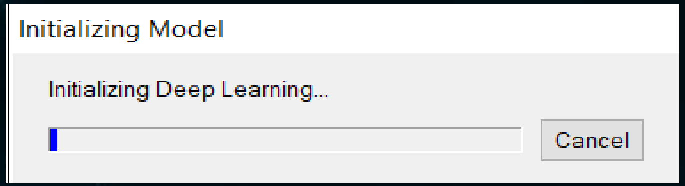

New Model
(SegUNet++)
⬛ Unclassified
🟥 TU-160_BlackJack
🟩 TU-95_Bear
🟦 Helicopter
🟨 Misc-Airplane
■ H6_Badger
Current Model
(SegUNet)

BREAKING CHANGE

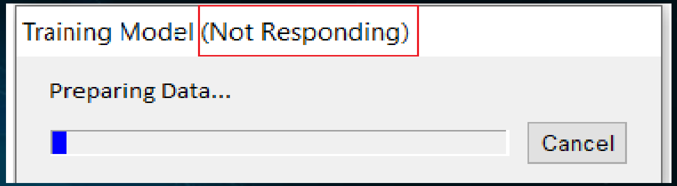
- The legacy “Single Class Model” can no longer be created in ENVI Deep Learning
- This is an old concept and no longer applicable to the current state of the application
- You can still process imagery using legacy single-class models

Progress Reporting

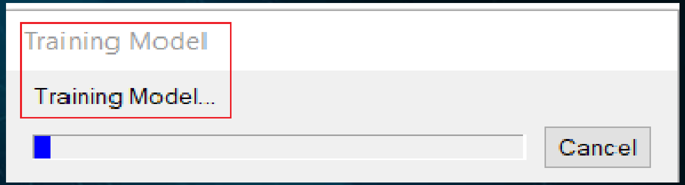
- Progress dialogs in ENVI when using ENVI Deep Learning were not always helpful or accurate
- They would also show “Not Responding” while we were waiting for message updates
- TensorBoard was more useful to track progress because you would get more frequent updates


Initializing Model

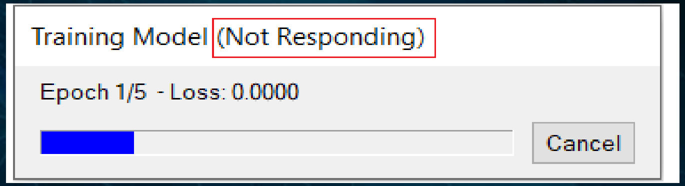
Initializing Deep Learning...


Training Model (Not Responding)

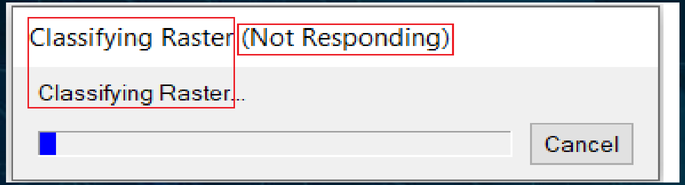
Preparing Data...


Training Model

Training Model...


Training Model (Not Responding)

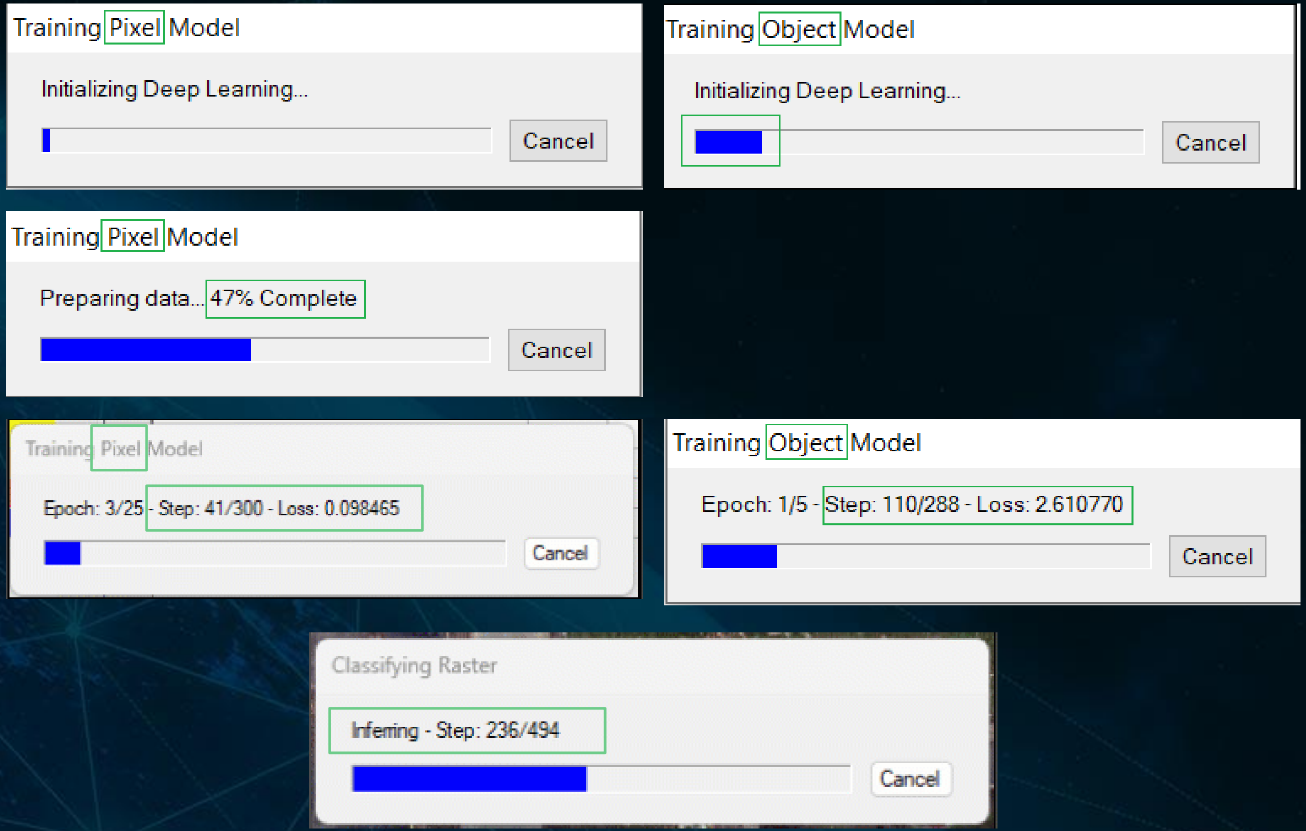
Epoch 1/5 - Loss: 0.0000


Classifying Raster (Not Responding)

Classifying Raster...

Progress Reporting: New User Experience

- Now, all dialogs provide much more insight into what is going on
- Many more progress updates provided for training to give you progress through each epoch
- Different types of deep learning now present the same progress dialogs



Training Pixel Model

Initializing Deep Learning...

Training Object Model

Initializing Deep Learning...

Training Pixel Model

Preparing data... 47% Complete

Training Pixel Model

Epoch: 3/25 - Step: 41/300 - Loss: 0.098465

Training Object Model

Epoch: 1/5 - Step: 110/288 - Loss: 2.610770

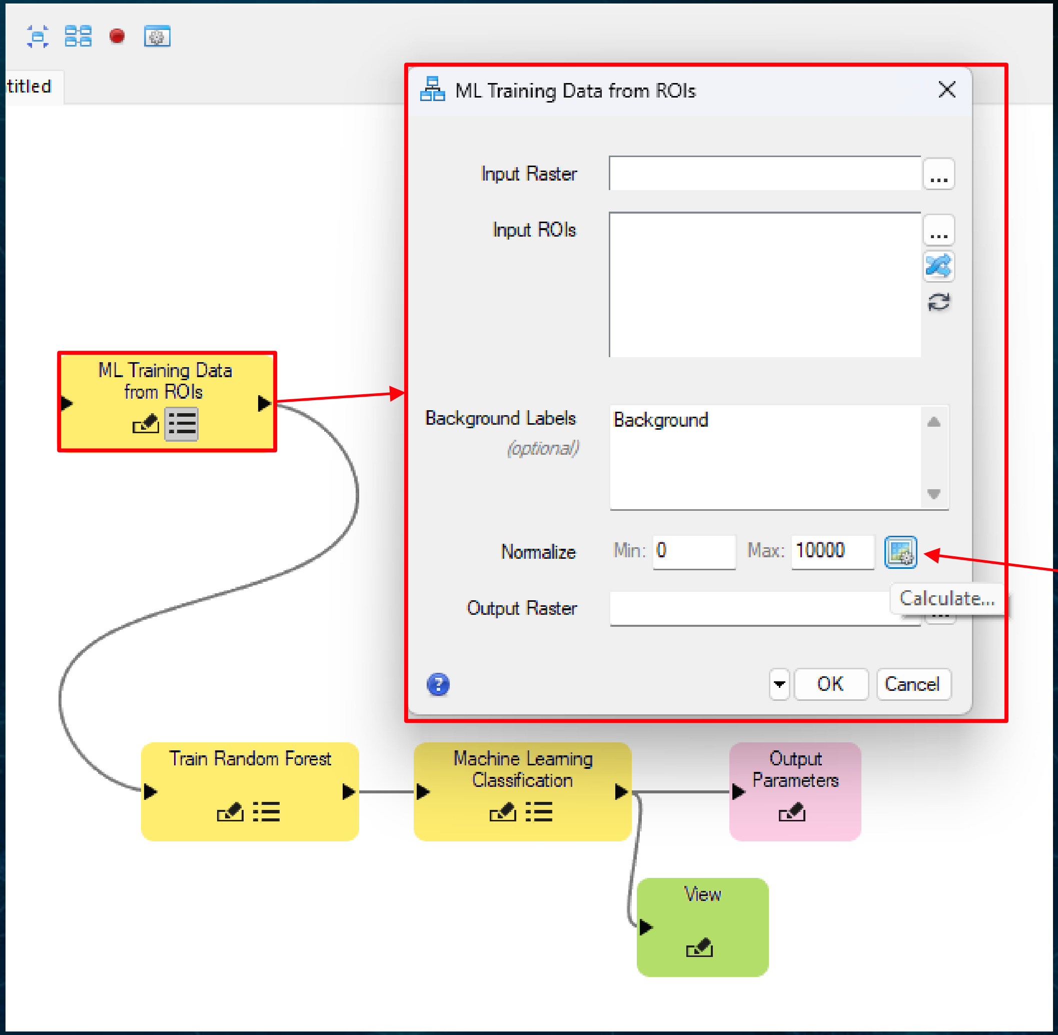
Classifying Raster

Inferring - Step: 236/494

Screenshots showing new dialogs and differences from previous versions

MACHINE LEARNING

Machine Learning: Normalization



ML Training Data from ROIs

Input Raster

Input ROIs

Background Labels *(optional)*: Background

Normalize Min: 0 Max: 10000

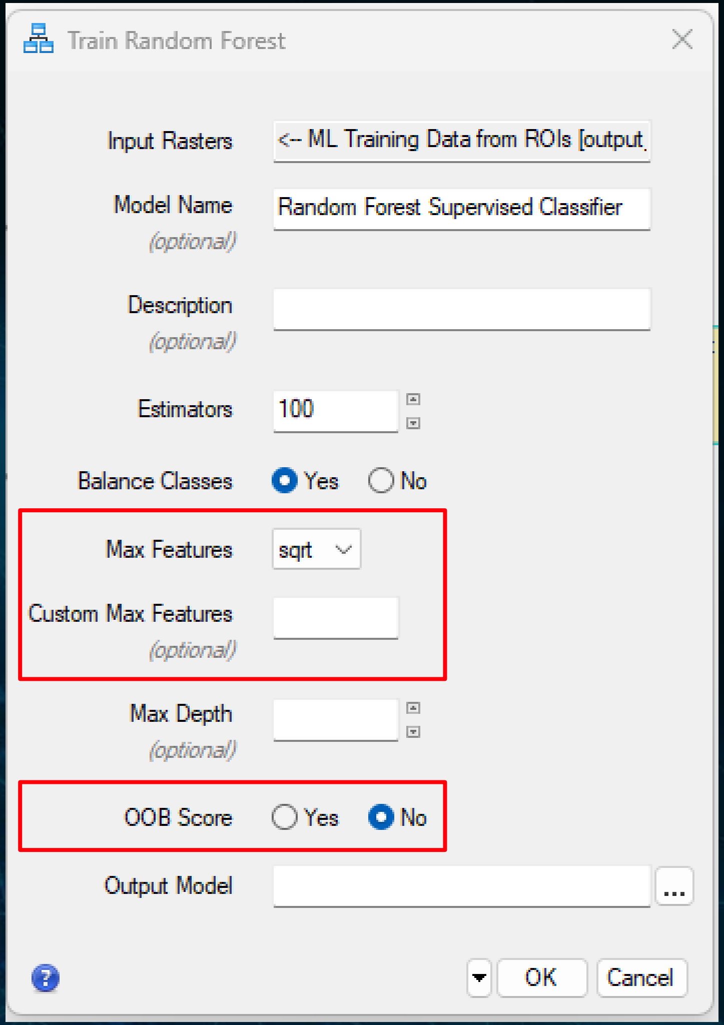
Output Raster

We have improved the automatically generated normalization statistics for machine learning to work better with hyperspectral datasets.

However, if you know the range of your data (from 0% to 100% surface reflectance) it is recommended to use those values

Updates: Random Forest and Extra Trees

- New parameters for both:
  - **Max features** controls the number of features used to find the best split
  - **Custom max features** lets you manually specify the number of features used for the best fit
- Random forest only:
  - **OOB score** tells ENVI Machine Learning to calculate and return the OOB (Out-of-Bag) score when it generates the model



Train Random Forest

Input Rasters: <-- ML Training Data from ROIs [output

Model Name *(optional)*: Random Forest Supervised Classifier

Description *(optional)*:

Estimators: 100

Balance Classes: ● Yes ○ No

Max Features: sqrt

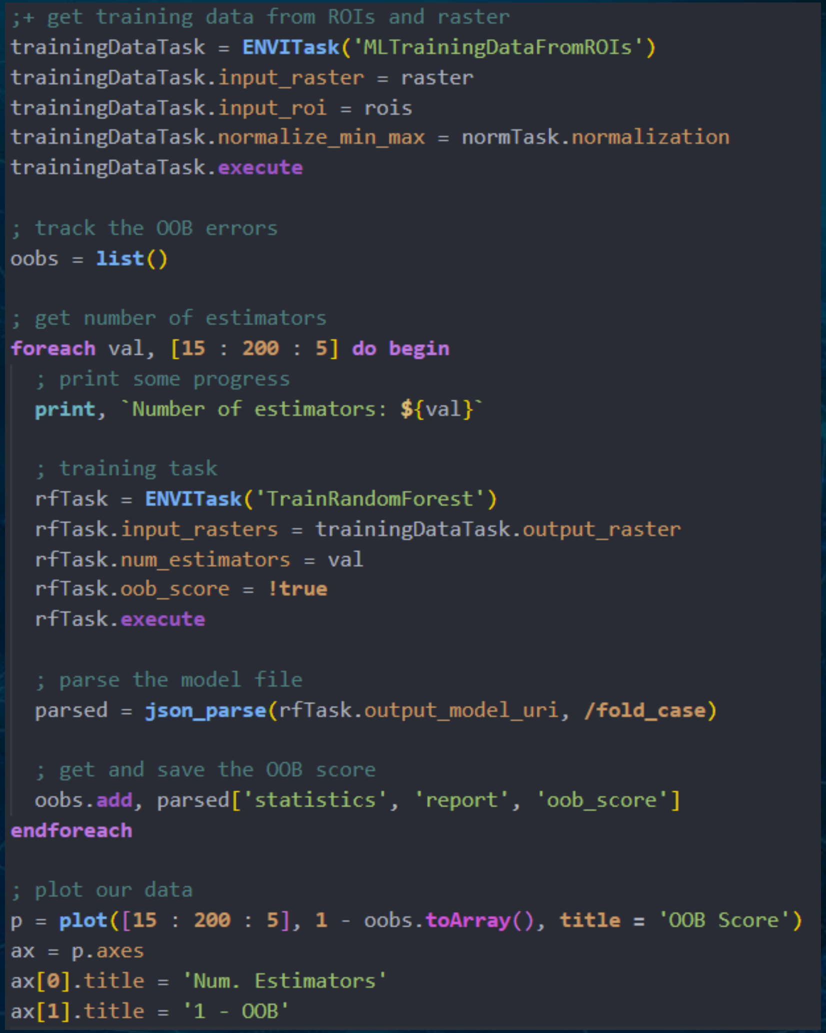
Custom Max Features *(optional)*:

Max Depth *(optional)*:

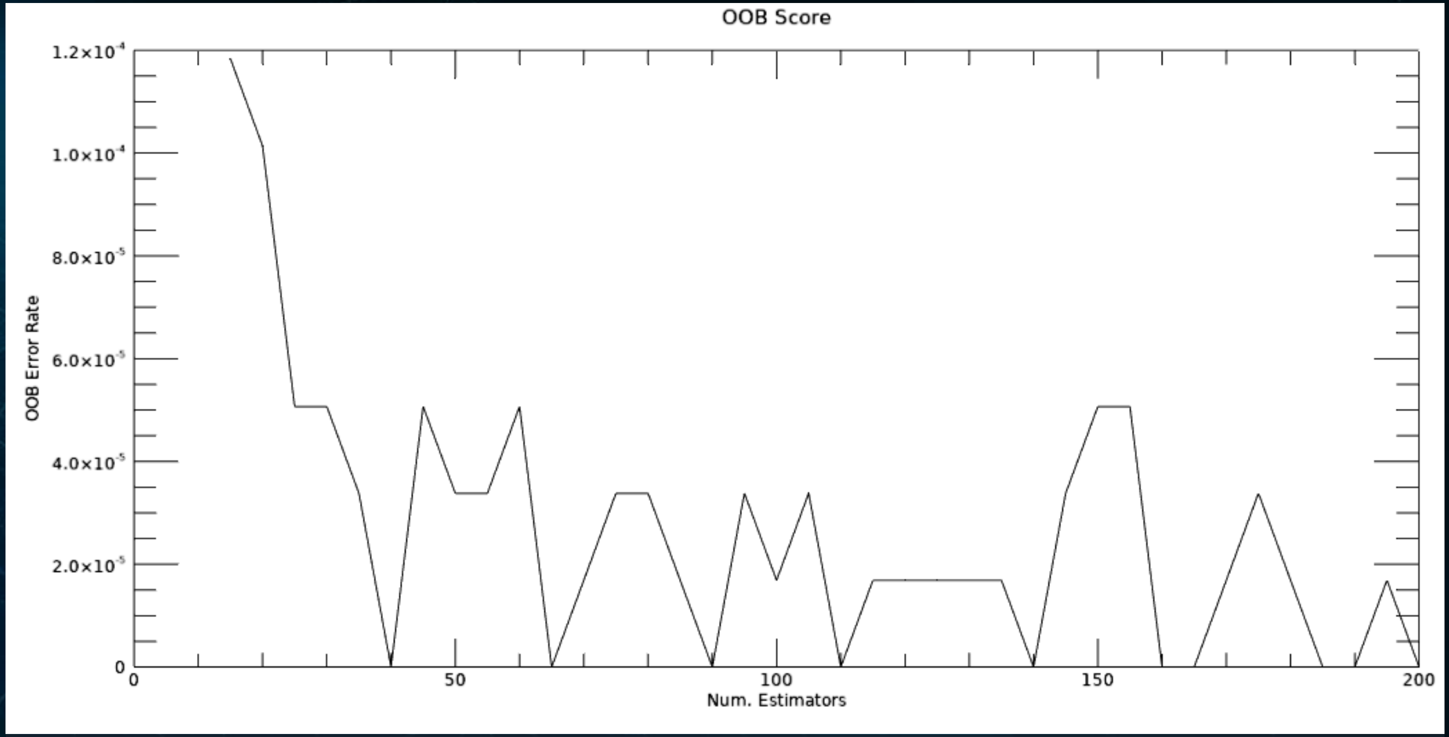
OOB Score: ○ Yes ● No

Output Model:

Using OOB Score



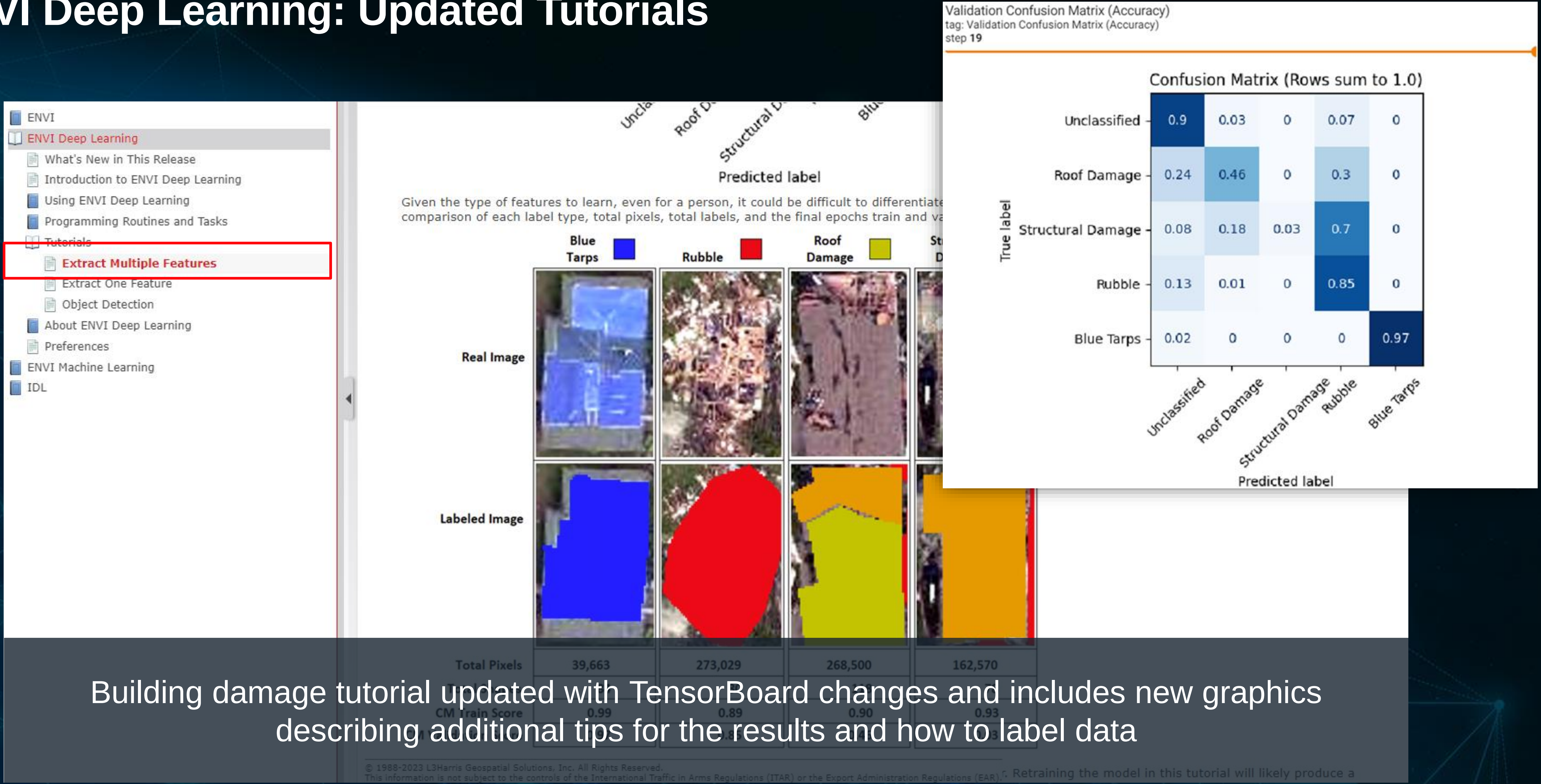
```
;+ get training data from ROIs and raster
trainingDataTask = ENVITask('MLTrainingDataFromROIs')
trainingDataTask.input_raster = raster
trainingDataTask.input_roi = rois
trainingDataTask.normalize_min_max = normTask.normalization
trainingDataTask.execute

; track the OOB errors
oobs = list()

; get number of estimators
foreach val, [15 : 200 : 5] do begin
  ; print some progress
  print, `Number of estimators: ${val}`

  ; training task
  rfTask = ENVITask('TrainRandomForest')
  rfTask.input_rasters = trainingDataTask.output_raster
  rfTask.num_estimators = val
  rfTask.oob_score = !true
  rfTask.execute

  ; parse the model file
  parsed = json_parse(rfTask.output_model_uri, /fold_case)

  ; get and save the OOB score
  oobs.add, parsed['statistics', 'report', 'oob_score']
endforeach

; plot our data
p = plot([15 : 200 : 5], 1 - oobs.toArray(), title = 'OOB Score')
ax = p.axes
ax[0].title = 'Num. Estimators'
ax[1].title = '1 - OOB'
```



Out-of-Bag (OOB) error rate plot from random forest classifiers made with ENVI Machine Learning and the ENVI API in IDL

NOTABLES

ENVI Deep Learning: Updated Tutorials



Given the type of features to learn, even for a person, it could be difficult to differentiate
comparison of each label type, total pixels, total labels, and the final epochs train and va

	Blue Tarps 🟦	Rubble 🟥	Roof Damage 🟨	St D
Real Image				
Labeled Image				
Total Pixels	39,663	273,029	268,500	162,570
CM Train Score	0.99	0.89	0.90	0.93

© 1988-2023 L3Harris Geospatial Solutions, Inc. All Rights Reserved.
This information is not subject to the controls of the International Traffic in Arms Regulations (ITAR) or the Export Administration Regulations (EAR).

r. Retraining the model in this tutorial will likely produce a

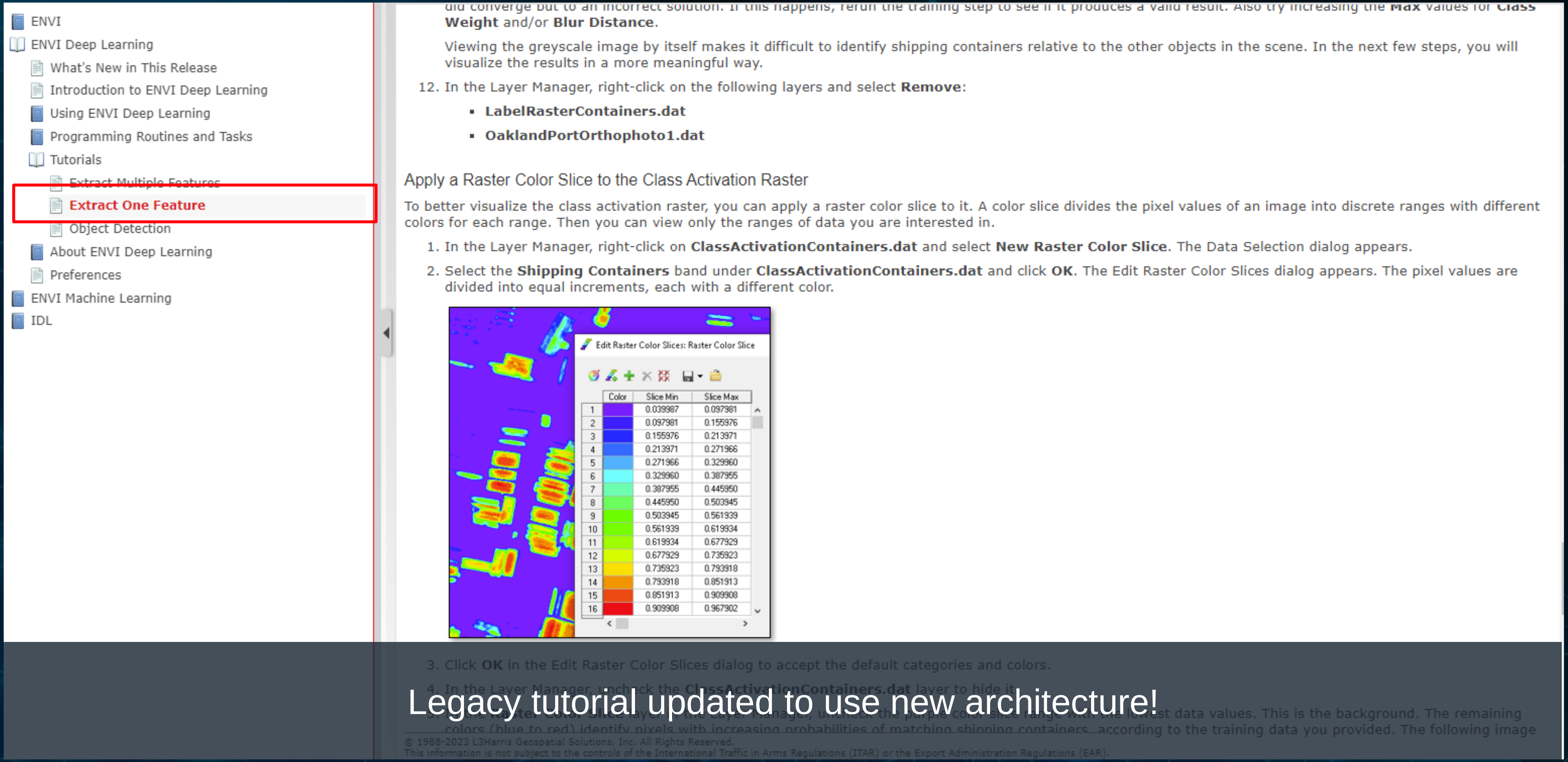
Validation Confusion Matrix (Accuracy)
tag: Validation Confusion Matrix (Accuracy)
step 19

Confusion Matrix (Rows sum to 1.0)

True label \ Predicted label	Unclassified	Roof Damage	Structural Damage	Rubble	Blue Tarps
Unclassified	0.9	0.03	0	0.07	0
Roof Damage	0.24	0.46	0	0.3	0
Structural Damage	0.08	0.18	0.03	0.7	0
Rubble	0.13	0.01	0	0.85	0
Blue Tarps	0.02	0	0	0	0.97

Building damage tutorial updated with TensorBoard changes and includes new graphics describing additional tips for the results and how to label data

ENVI Deep Learning: Updated Tutorials



- ENVI
- ENVI Deep Learning
  - What's New in This Release
  - Introduction to ENVI Deep Learning
  - Using ENVI Deep Learning
  - Programming Routines and Tasks
  - Tutorials
    - Extract Multiple Features
    - Extract One Feature
    - Object Detection
  - About ENVI Deep Learning
  - Preferences
- ENVI Machine Learning
- IDL

did converge but to an incorrect solution. If this happens, rerun the training step to see if it produces a valid result. Also try increasing the **Max** values for **Class Weight** and/or **Blur Distance**.

Viewing the greyscale image by itself makes it difficult to identify shipping containers relative to the other objects in the scene. In the next few steps, you will visualize the results in a more meaningful way.

12. In the Layer Manager, right-click on the following layers and select **Remove**:
    - **LabelRasterContainers.dat**
    - **OaklandPortOrthophoto1.dat**

Apply a Raster Color Slice to the Class Activation Raster

To better visualize the class activation raster, you can apply a raster color slice to it. A color slice divides the pixel values of an image into discrete ranges with different colors for each range. Then you can view only the ranges of data you are interested in.

1. In the Layer Manager, right-click on **ClassActivationContainers.dat** and select **New Raster Color Slice**. The Data Selection dialog appears.
2. Select the **Shipping Containers** band under **ClassActivationContainers.dat** and click **OK**. The Edit Raster Color Slices dialog appears. The pixel values are divided into equal increments, each with a different color.

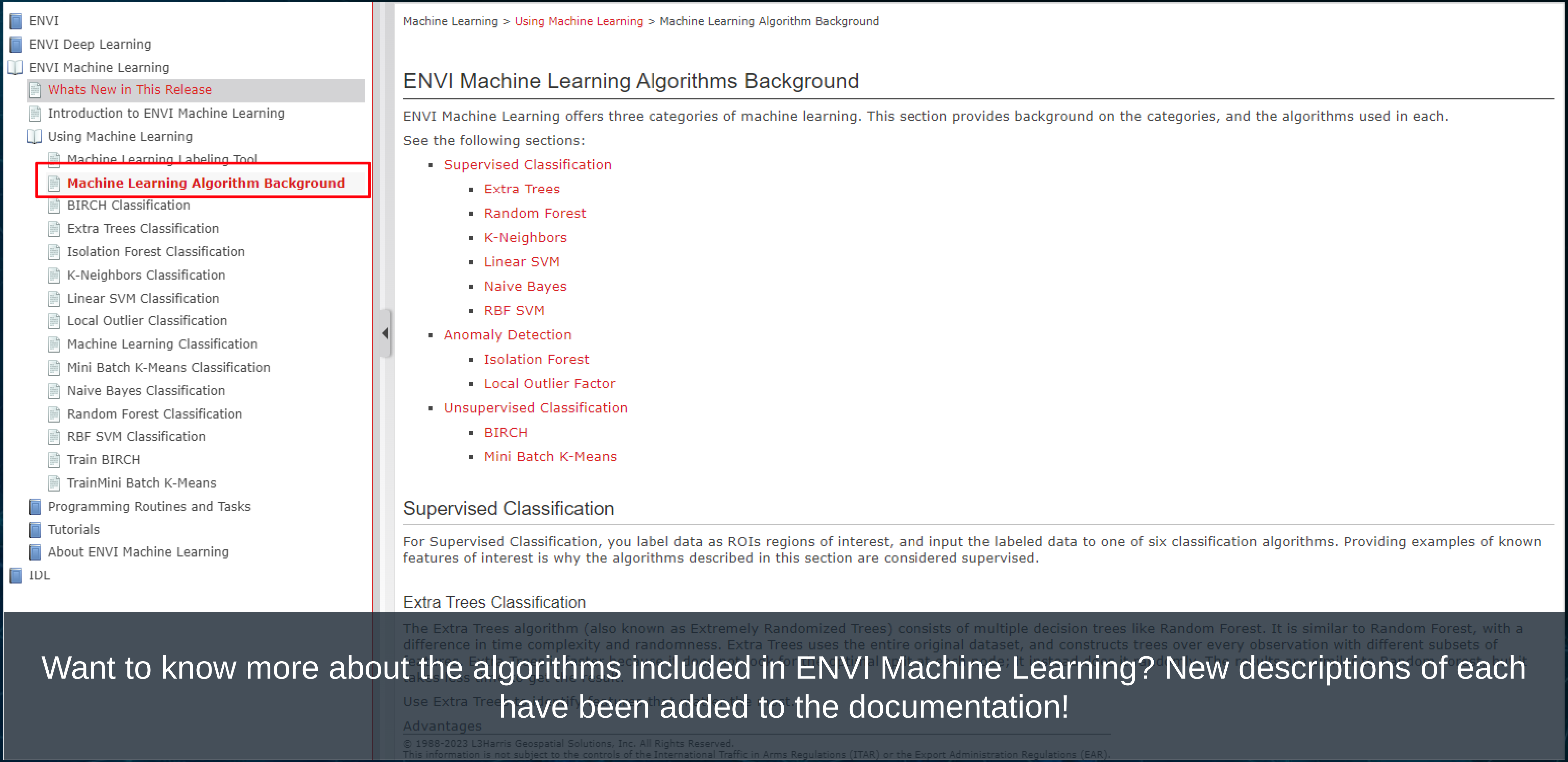
Edit Raster Color Slices: Raster Color Slice

	Color	Slice Min	Slice Max
1	🟪	0.039987	0.097981
2	🟦	0.097981	0.155976
3	🟦	0.155976	0.213971
4	🟦	0.213971	0.271966
5	🟦	0.271966	0.329960
6	🟦	0.329960	0.387955
7	🟩	0.387955	0.445950
8	🟩	0.445950	0.503945
9	🟩	0.503945	0.561939
10	🟩	0.561939	0.619934
11	🟩	0.619934	0.677929
12	🟨	0.677929	0.735923
13	🟨	0.735923	0.793918
14	🟧	0.793918	0.851913
15	🟧	0.851913	0.909908
16	🟥	0.909908	0.967902

3. Click **OK** in the Edit Raster Color Slices dialog to accept the default categories and colors.
4. In the Layer Manager, uncheck the **ClassActivationContainers.dat** layer to hide it.
5. In the **Raster Color Slice** layer in the Layer Manager, uncheck the purple color slice range with the lowest data values. This is the background. The remaining colors (blue to red) identify pixels with increasing probabilities of matching shipping containers, according to the training data you provided. The following image

Legacy tutorial updated to use new architecture!

Machine Learning: Algorithm Background



ENVI
ENVI Deep Learning
ENVI Machine Learning
Whats New in This Release
Introduction to ENVI Machine Learning
Using Machine Learning
Machine Learning Labeling Tool
Machine Learning Algorithm Background
BIRCH Classification
Extra Trees Classification
Isolation Forest Classification
K-Neighbors Classification
Linear SVM Classification
Local Outlier Classification
Machine Learning Classification
Mini Batch K-Means Classification
Naive Bayes Classification
Random Forest Classification
RBF SVM Classification
Train BIRCH
TrainMini Batch K-Means
Programming Routines and Tasks
Tutorials
About ENVI Machine Learning
IDL

Machine Learning > Using Machine Learning > Machine Learning Algorithm Background

ENVI Machine Learning Algorithms Background

ENVI Machine Learning offers three categories of machine learning. This section provides background on the categories, and the algorithms used in each.

See the following sections:

- Supervised Classification
  - Extra Trees
  - Random Forest
  - K-Neighbors
  - Linear SVM
  - Naive Bayes
  - RBF SVM
- Anomaly Detection
  - Isolation Forest
  - Local Outlier Factor
- Unsupervised Classification
  - BIRCH
  - Mini Batch K-Means

Supervised Classification

For Supervised Classification, you label data as ROIs regions of interest, and input the labeled data to one of six classification algorithms. Providing examples of known features of interest is why the algorithms described in this section are considered supervised.

Extra Trees Classification

The Extra Trees algorithm (also known as Extremely Randomized Trees) consists of multiple decision trees like Random Forest. It is similar to Random Forest, with a difference in time complexity and randomness. Extra Trees uses the entire original dataset, and constructs trees over every observation with different subsets of features. Extra Trees is faster because it does not look for the optimal split at each node; it instead does it randomly. The results are similar to Random Forest, but it takes less time to get the result.

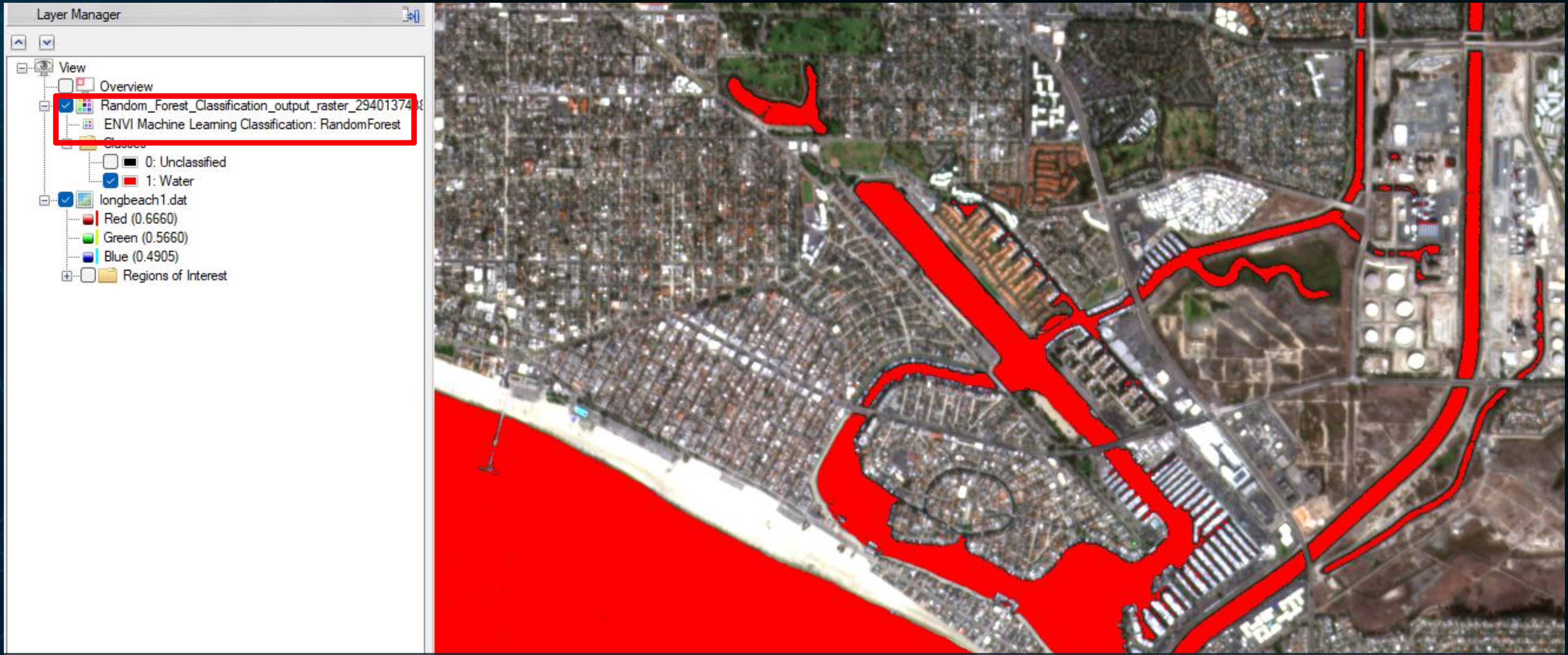
Use Extra Trees to identify features [illegible].

Advantages

© 1988-2023 L3Harris Geospatial Solutions, Inc. All Rights Reserved.
This information is not subject to the controls of the International Traffic in Arms Regulations (ITAR) or the Export Administration Regulations (EAR).

Want to know more about the algorithms included in ENVI Machine Learning? New descriptions of each have been added to the documentation!

User Experience: Band Names

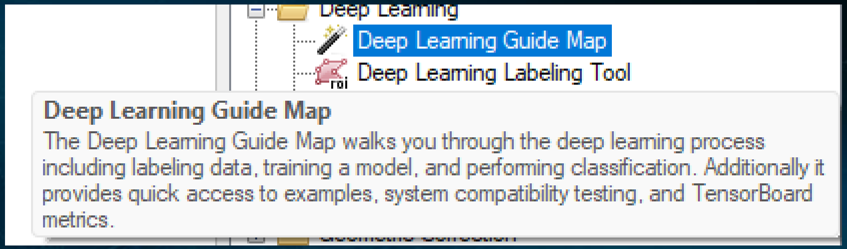


Layer Manager

- View
  - ☐ Overview
  - ☑ Random_Forest_Classification_output_raster_29401374
    - ENVI Machine Learning Classification: RandomForest
    - Classes
      - ☐ 0: Unclassified
      - ☑ 1: Water
  - ☑ longbeach1.dat
    - Red (0.6660)
    - Green (0.5660)
    - Blue (0.4905)
    - ☐ Regions of Interest

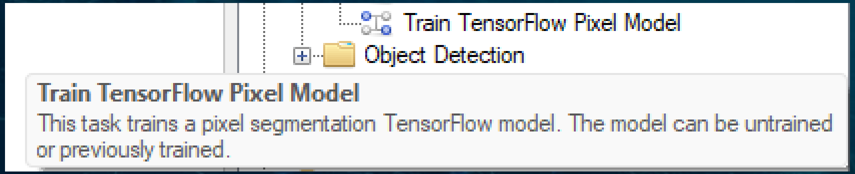
Images classified with ENVI Deep Learning or ENVI Machine Learning save the model type in the band names, making it easy to compare results from different models without losing track of the source.

Usability: New Tooltips



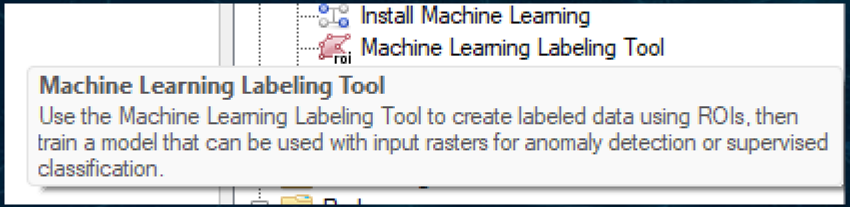
Deep Learning Guide Map

The Deep Learning Guide Map walks you through the deep learning process including labeling data, training a model, and performing classification. Additionally it provides quick access to examples, system compatibility testing, and TensorBoard metrics.



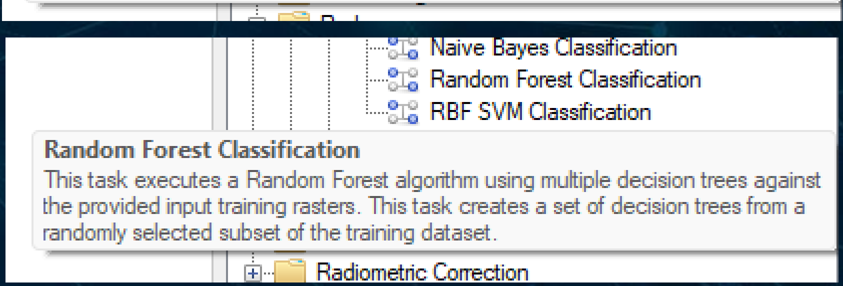
Train TensorFlow Pixel Model

This task trains a pixel segmentation TensorFlow model. The model can be untrained or previously trained.



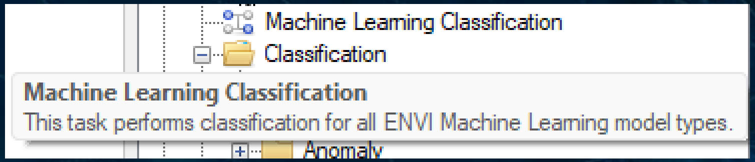
Machine Learning Labeling Tool

Use the Machine Learning Labeling Tool to create labeled data using ROIs, then train a model that can be used with input rasters for anomaly detection or supervised classification.



Random Forest Classification

This task executes a Random Forest algorithm using multiple decision trees against the provided input training rasters. This task creates a set of decision trees from a randomly selected subset of the training dataset.



Machine Learning Classification

This task performs classification for all ENVI Machine Learning model types.

Like ENVI, the machine learning and deep learning toolbox entries now have tooltips with basic descriptions of each tool

Questions and Discussion

Bill Okubo

Product Manager

bill.okubo@nv5.com

Zach Norman

Product Manager

zachary.norman@nv5.com

NV5 Geospatial

www.NV5GeospatialSoftware.com
geospatialinfo@nv5.com
303-786-9900