Machine Learning Research for Automating Dotting Process

Applied Practice Experience – The Water Institute of the Gulf

Liam Guest, MSPH Candidate (Biostatistics and Data Science)

PROJECT OVERVIEW

SMART Objective: By August 2025, complete research and pilot testing for a machine learning object detection model, generate training and validation metrics from at least two parameterized runs, and deliver a written summary of findings and recommendations to inform future model development.

Primary Goal: reduce manual workload while maintaining precision

Phase 1: research + pilot testing to assess feasibility, document hurdles, and outline recommendations

Ultimate Vision: seamless integration of automated symbol detection model into existing workflow, increasing efficiency, saving time, and furthering the mission

BACKGROUND: AVIAN MONITORING WORKFLOW

1. **Survey and Image Collecting**
    a. High-altitude overview imagery from fixed mount cameras for basemap context
    b. Low-altitude detailed imagery for fine-scale colony counting (dotting)
2. **Identify Images to Mosaic**
    a. Find and remove blurry images
3. **Create Mosaics**
    a. Using adobe workshop or python packages
    b. Save as tiff file
4. **Georeference Mosaics**
    a. Using the basemap, georeference images to their proper locations
    b. Save control points as coordinates
5. **Dotting Process**
    a. **The process we attempt to automate**

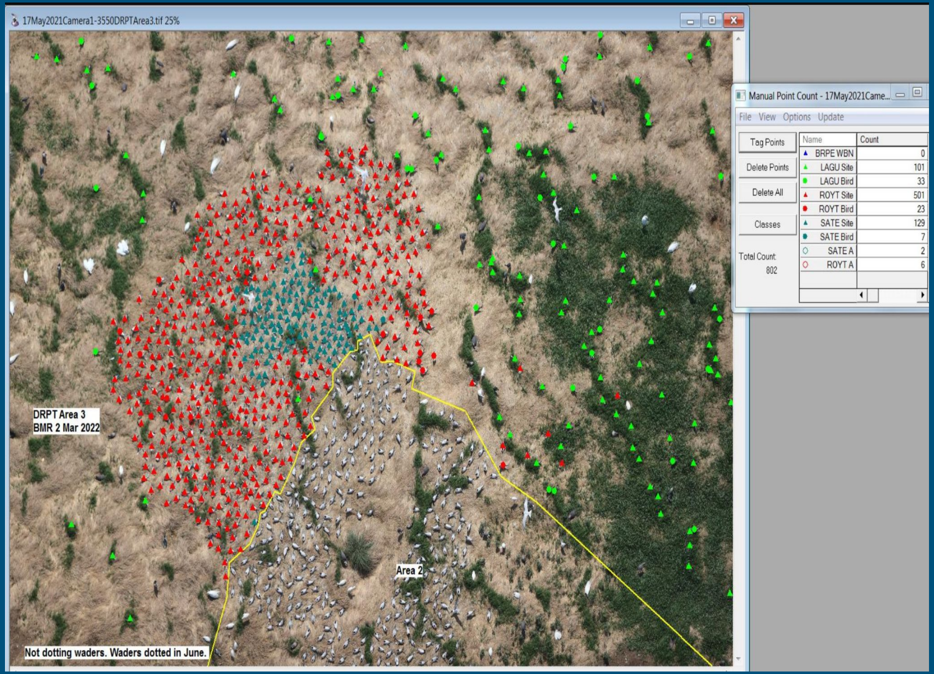
Dotting Process Automation

Manual Dotting Process in ArcGIS Pro:

- Color = species, shape = observation type (nest, bird, site)
- Dotters use “unique values” symbology by combining species code and observation type
- End products are used for colonial waterbird endpoint estimation, and ultimately conservation efforts

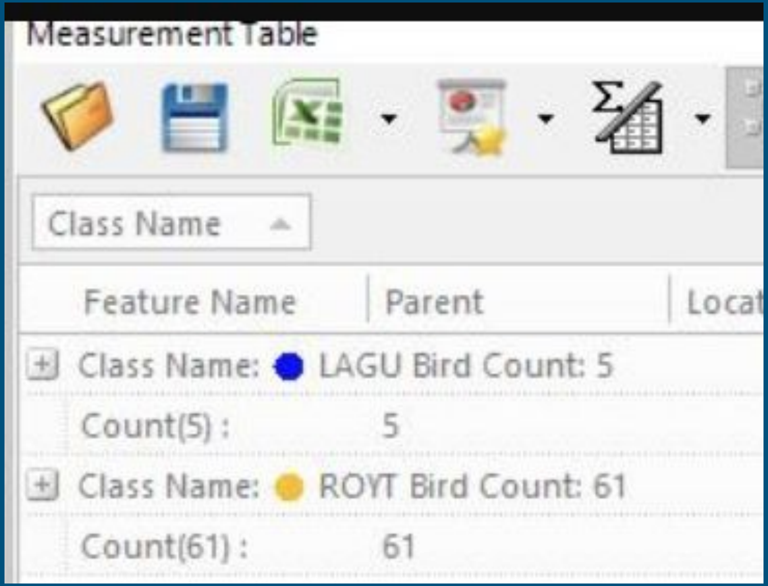
Why Automate?

- Current efforts are accurate but time consuming
- Hundreds, sometimes thousands of dots per image
- Frees up human time for more complex processes related to the saem mission

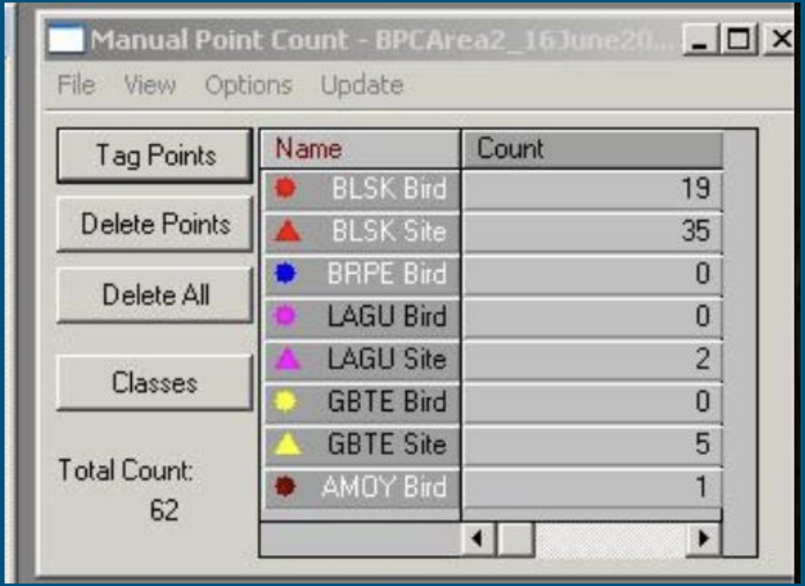


Hurdles: Inconsistent Symbology

Biloxi North: Martin Island



Biloxi North: Bayou Pintou



Solution: Inconsistent Symbology

Not only were the legends themselves inconsistent, the nature of the object detection machine learning model we were attempting to train struggled to inform meaning from a green square vs a green circle consistently. This led to two key informed takeaways:

1. We will not create a model of value by training on multiple symbol classes (green squares, green circles, red squares), instead, let's focus on one symbol (green circle)
2. Symbology informs different meanings almost every time, so, we either needed to build a model to extract the legend and illicit meaning or we need a human-in-the-loop to make those decisions each time.

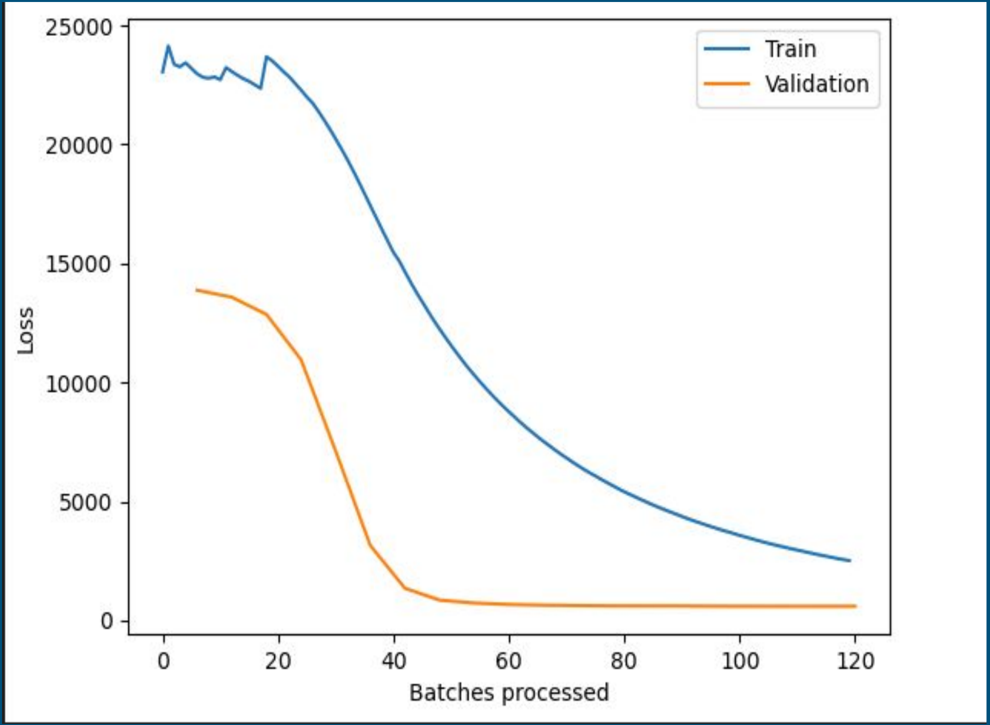
RESEARCH APPROACH

1. Data Preparation:
   a. Merge May/June dotting feature classes
   b. Create buffer bounding boxes
   c. Organize images into a mosaic dataset
2. Model Training (ArcGIS Deep Learning Tools)
   a. Create label dataset for validation
3. Evaluation
   a. Monitor average precision (AP)
   b. Track detection accuracy
4. Documentation
   a. Record effective practices, and specific measures in OneNote

PARAMETER RESEARCH

Parameter	Think of it like...	Good Value
Chip Size	Puzzle piece size	256 or 512
Stride	Jump distance between pieces	128 or 256
Model Type	Brain type	FasterRCNN
Backbone	Memory size	ResNet50
Batch Size	How many at once	2–4
Epochs	How many times to review	30–50
Validation %	Test questions	20
Monitor Metric	Scoreboard	mAP

First Run:

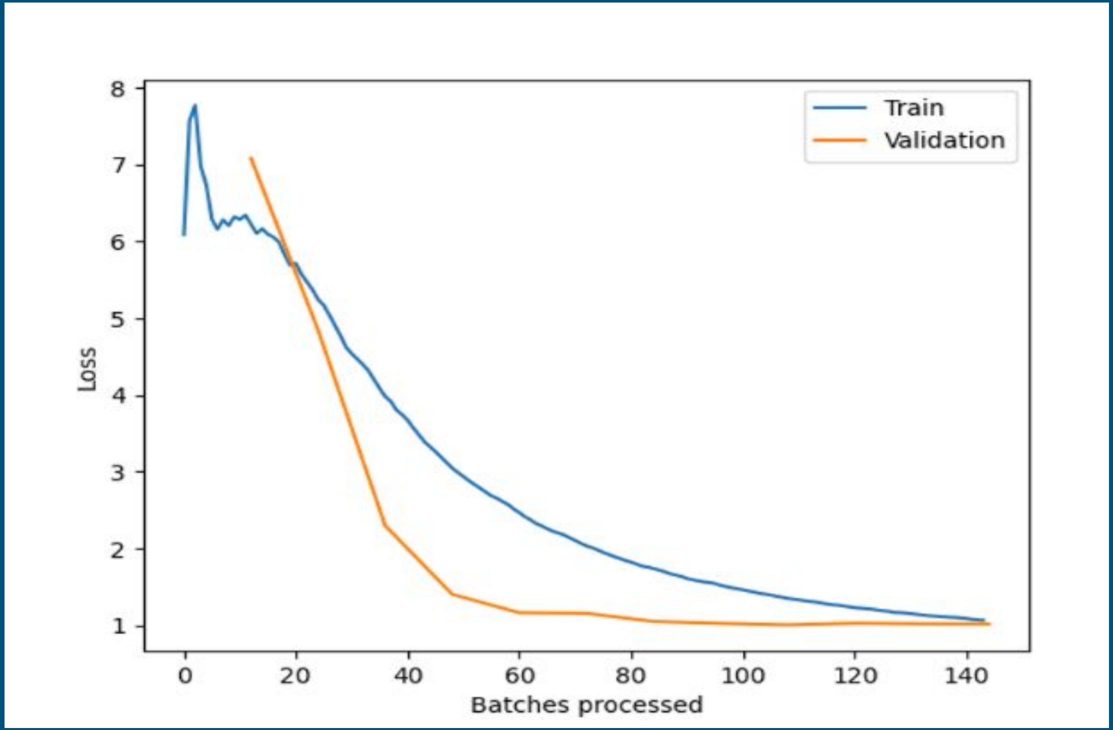


Blue line: shows how well the model is doing on data it is learning from.

Orange line: shows how well the model is doing on new data is hasn’t seen before

This is a good sign, the model did learn, but according to the statistical output, not nearly well enough to be integrated into the workflow.

Most Recent Run:



Significant changes to the y-axis. This is result of changes to the parameters on the backend.

However, the most important thing to note is still the level of the curve. The validation (orange) curve reaches 0 by batch 50, which indicates the model reached its best performance quickly. In other words, it learned quickly, but can not go very far.

Model Log

Model Name	Date	Classes	Epochs	AP Scores	Chip Size	Learning rate	Notes
Model_1_YOLOv3	7/10	All	10	All 0.0	256	?	Fail
Model_2_RCNN	7/10	All	11 (early stop)/40	Only '61' = 0.19	512	0.0001	Some Learning
Model_3_RCNN_Focused	7/15	GREG, DCCO	50	???	512	0.0009	

Future Recommendations

Measure	Recommendation	Reasoning
Model Type	FasterRCNN	Outperformed other tested model types
Learning Rate	.0001	Stable convergence
Validation %	15-20%	Recommendation I found through research.
Symbology	Train on one symbol that can be used to inform multiple meanings	A model will be more valid and precise if trained on one symbol
Chip Size	256	Trained smoothly