



Personal Product Dev Story

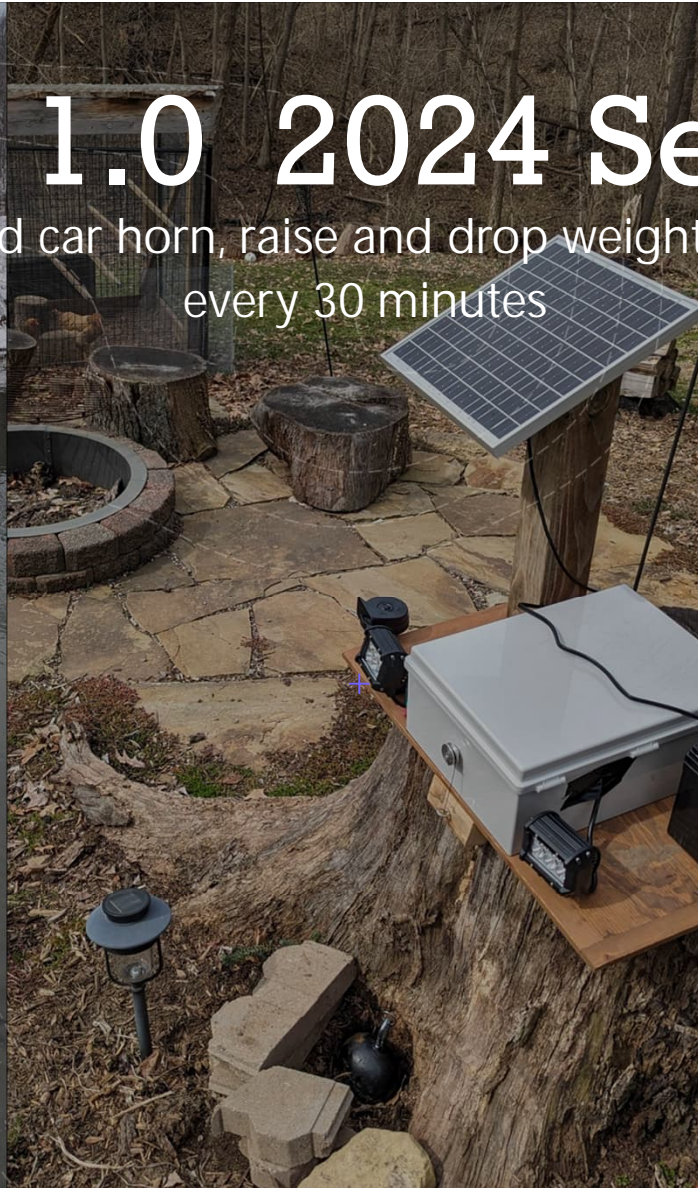
featuring AI & CVML
By Mark Cherney

Objective: Keep Otters and Beavers out of my pond

- + Why? Otters eat all the fish, Beaver make a mess
- + Mission: Make a heavy duty, solar powered vermin repeller
- + How to detect them when swimming in the water?
 - + Garage door sensor (IR) – poor range outdoors
 - + Broken laser beam – both emitter and receiver need to stay oriented with changes in water level.
 - + Ultrasonic sensor/emitter
 - + IR Camera with ML
 - + RGB Camera with ML

MVP 1.0 2024 Season

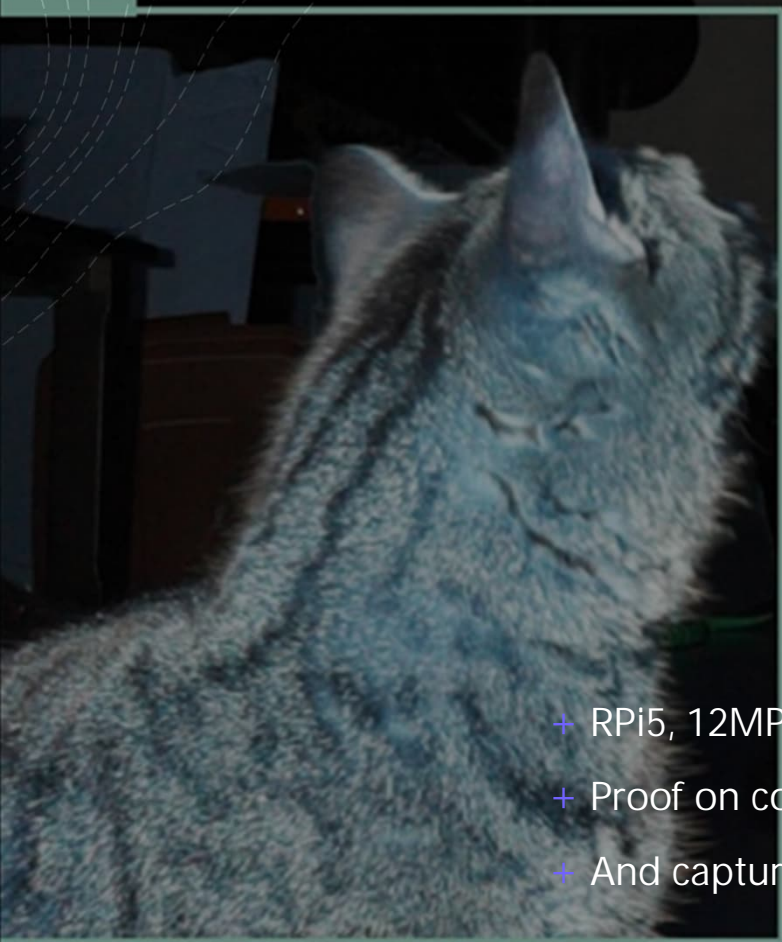
Strobe lights, sound car horn, raise and drop weight with stepper motor every 30 minutes



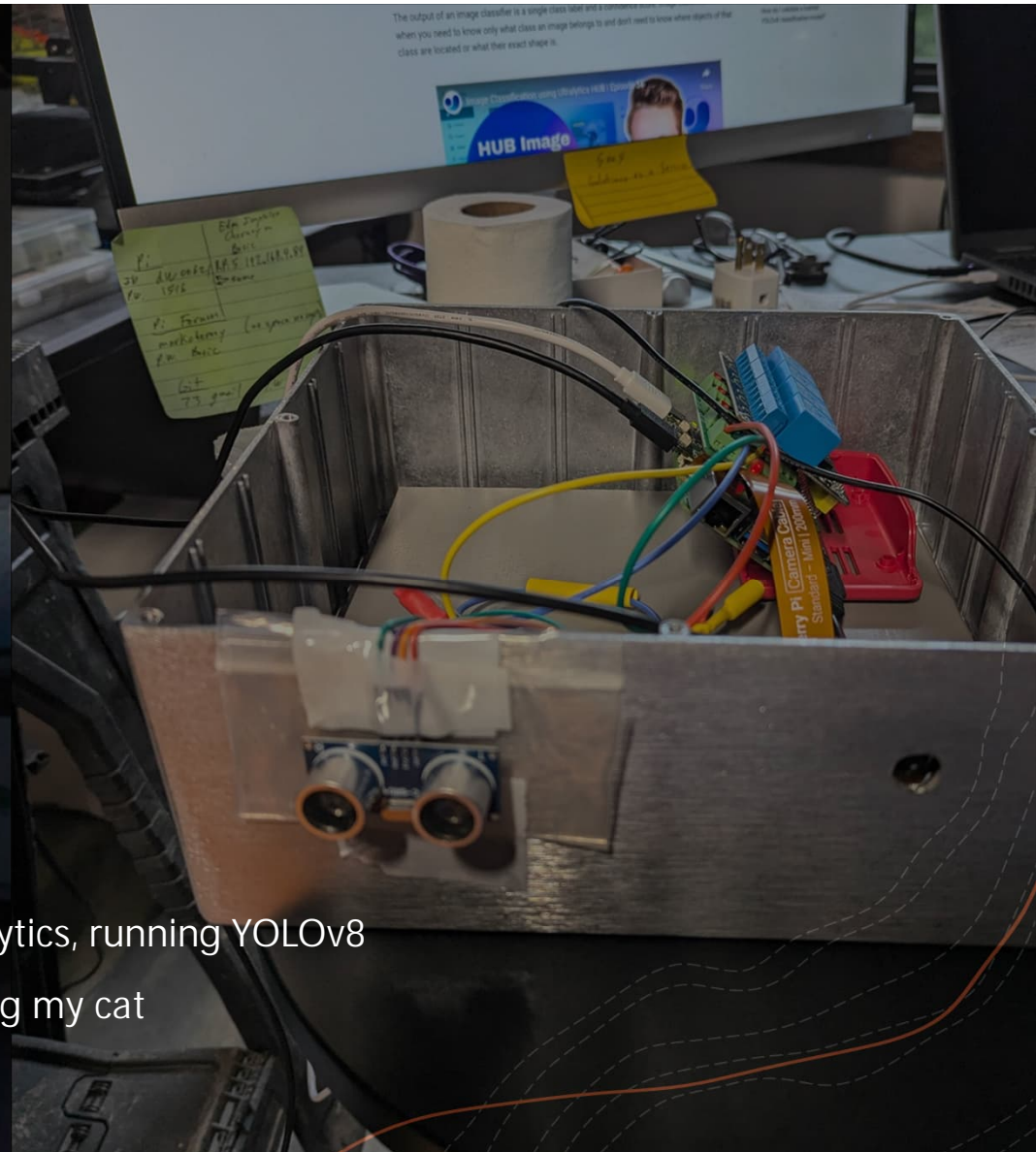
MVP 1.1 2025 Season



Meanwhile Getting Ready for 2.0

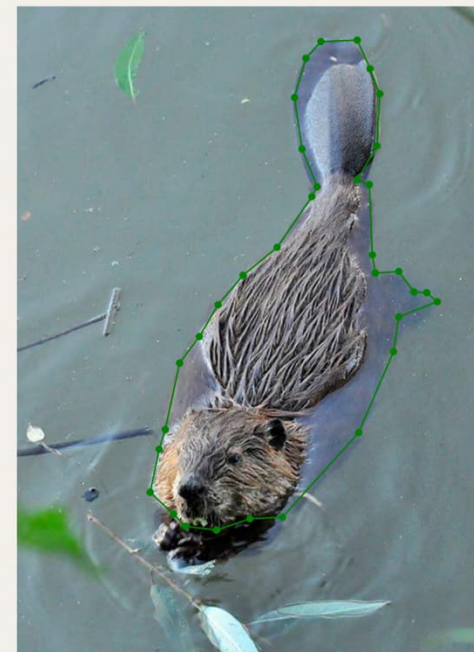


- + RPi5, 12MP camera, Ultralytics, running YOLOv8
- + Proof on concept detecting my cat
- + And captured images



Beavers and Otters Data Set

- +Manual data curation from internet search
- +Manual data labels (segments) with LabelMe app
- +AI provided augmentations with labels
- +AI provided additional legal and “manual” image sources and deduplications script



Examples AI Generated Synthetic Augmentation



8be78ef4-702a-4752-a66d-56d59fe27f
04_aug_000.jpg



8be78ef4-702a-4752-a66d-56d59fe27f
04_aug_001.jpg



8be78ef4-702a-4752-a66d-56d59fe27f
04_aug_002.jpg



8be78ef4-702a-4752-a66d-56d59fe27f
04_aug_003.jpg



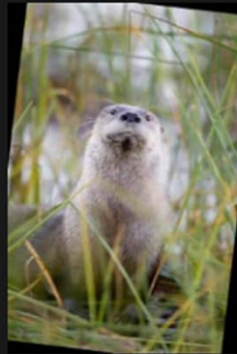
8be78ef4-702a-4752-a66d-56d59fe27f
04_aug_004.jpg



9 Amazing Facts About River
Otters_aug_000.jpg



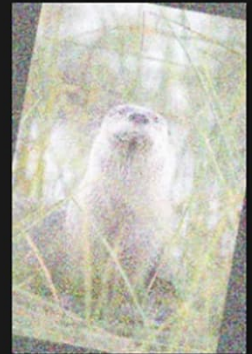
9 Amazing Facts About River
Otters_aug_001.jpg



9 Amazing Facts About River
Otters_aug_002.jpg



9 Amazing Facts About River
Otters_aug_003.jpg



9 Amazing Facts About River
Otters_aug_004.jpg

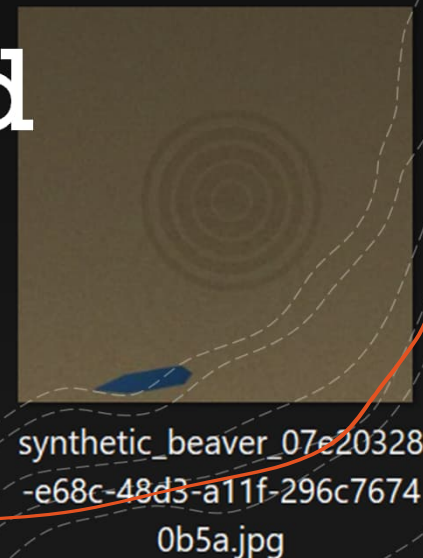
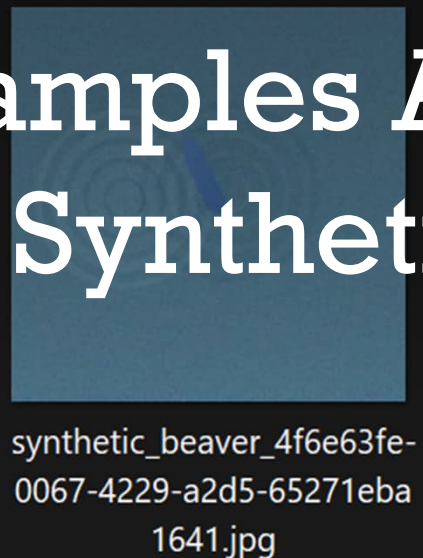
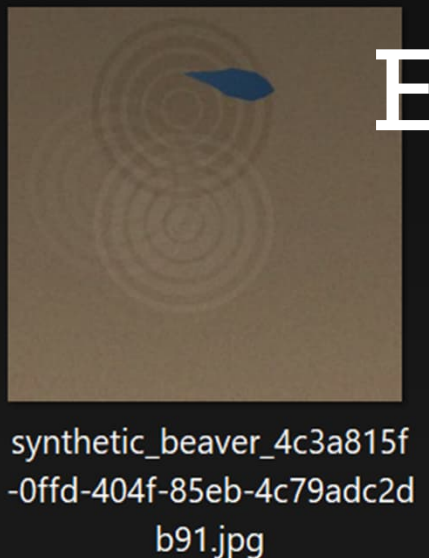
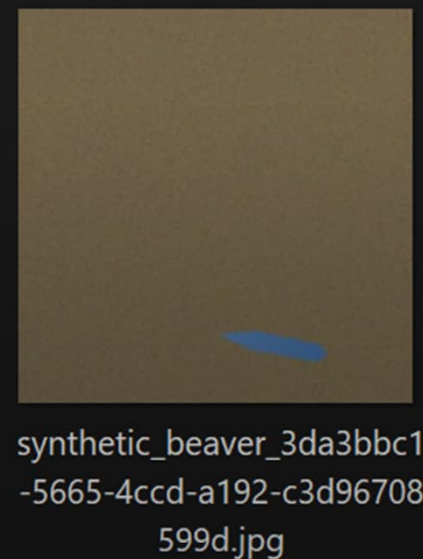
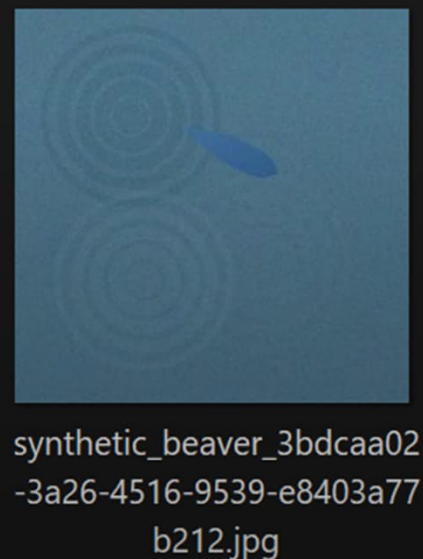
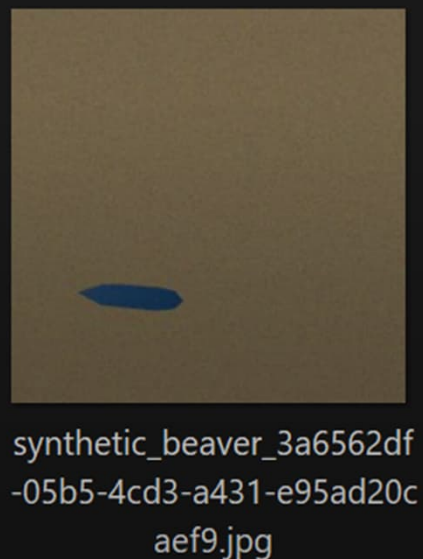
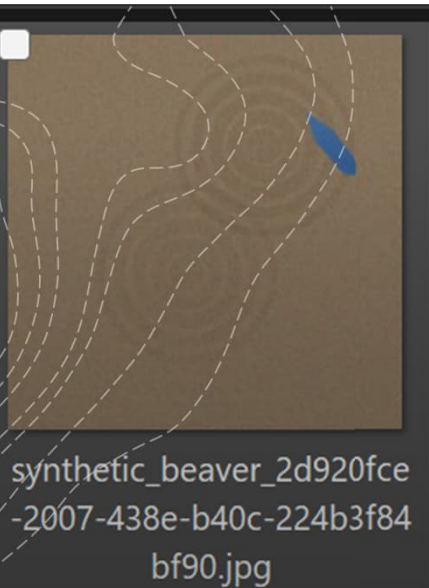
Training a YOLOv8 model to detect Beavers and Otters

Test data set is 18 images: 7 instances of beavers, 8 of otters.

Run ---->	Bounding boxes 100ep		Seg different test data set		Seg different 50 ep		Seg same test set 100ep		Seg, full AI Aug 100 ep		Seg, Geo AI Aug 100 ep	
	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
Beavers	0.946	0.817	0.856	0.734	0.704	0.513	0.374	0.282	0.424	0.333	0.319	0.232
Otter	0.796	0.482	0.61	0.402	0.513	0.421	0.0709	0.0222	0.373	0.178	0.212	0.165

- Bounding boxes training yields better results
- Test set needs to be the same
- 100 epochs is about right
- AI generated augmentation helps
- AI recommendation for sources for expanding training data set was helpful

Examples AI Generated Synthetic Images



Next Steps

- +Reload app in RPi5 due to corruption
- +Retrained with added images and corresponding augmentations
- +Focus on bounding box training
- +Deploy model in data collection kit in ODD
 - +Overfitting doesn't matter
 - +Need to turn lights on for night data collection, need more power
- +Investigate Pytest

24Dec2025 Bbox training, Full Aug, larger data set, 100 ep

```
Epoch      GPU_mem    box_loss    cls_loss    dfl_loss    Instances    Size
100/100      0G         0.353       0.2761      0.8386
              Class    Images    Instances    Box(P
              all      36        33          0.305      0.322      0.258      0.174

100 epochs completed in 23.550 hours.
Optimizer stripped from runs/detect/train6/weights/last.pt, 6.2MB
Optimizer stripped from runs/detect/train6/weights/best.pt, 6.2MB

Validating runs/detect/train6/weights/best.pt...
Ultralytics 8.3.183 🚀 Python-3.10.12 torch-2.8.0+cu128 CPU (12th Gen Intel Core(TM) i7-1265U)
Model summary (fused): 72 layers, 3,006,233 parameters, 0 gradients, 8.1 GFLOPs
              Class    Images    Instances    Box(P
              all      36        33          0.738      0.462      0.657      0.321
              beaver    13        15          0.741      0.2       0.501      0.397
              otter     10        16           1       0.186      0.475      0.251
              muskrat    2         2          0.472      1         0.995      0.316

Speed: 1.5ms preprocess, 85.9ms inference, 0.0ms loss, 0.6ms postprocess per image
Results saved to runs/detect/train6
💡 Learn more at https://docs.ultralytics.com/modes/train
VS Code: view Ultralytics VS Code Extension ⚡ at https://docs.ultralytics.com/integrations/vscode
dw00621@WDX7873BS3:~/ml$
```

14July'25 First ever Bbox training, 100 ep

```
Epoch  GPU_mem  box_loss  cls_loss  dfl_loss  Instances  Size
100/100    0G      0.3356    0.8937    0.9605      5      640: 100%|██████████| 4/4 [00:14<00:00, 3.65s/it]
          Class    Images  Instances  Box(P      R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:02<00:00, 2.12s/it]
          all       18       17      0.985    0.417    0.836    0.683

00 epochs completed in 0.506 hours.
optimizer stripped from runs/detect/train2/weights/last.pt, 6.3MB
optimizer stripped from runs/detect/train2/weights/best.pt, 6.3MB

Validating runs/detect/train2/weights/best.pt...
Ultralytics 8.3.165 Python-3.10.12 torch-2.7.1+cu126 CPU (12th Gen Intel Core(TM) i7-1265U)
Model summary (fused): 72 layers, 3,006,233 parameters, 0 gradients, 8.1 GFLOPs
          Class    Images  Instances  Box(P      R      mAP50  mAP50-95): 100%|██████████| 1/1 [00:01<00:00, 1.78s/it]
          all       18       17      0.852    0.579    0.912    0.731
        beaver        7        8      0.725    0.988    0.946    0.817
         otter        5        8      0.832    0.75    0.796    0.482
        muskrat        1        1        1        0    0.995    0.895

Speed: 1.0ms preprocess, 79.9ms inference, 0.0ms loss, 1.6ms postprocess per image
Results saved to runs/detect/train2
Learn more at https://docs.ultralytics.com/modes/train
(myvenv) dw00621@WDX7873BS3:~/ml$
```


Quantizing Model Comparison Based on Original BBox model from July'25 model

Model	mAP50	mAP50-95	Inference (ms)		Size
PyTorch (best.pt)	0.418	0.253	53.7	5.9 MB	
Float32 TFLite	0.403	0.240	88.5	11.7 MB	
INT8 TFLite	0.383	0.217	50.1	~3 MB	

Different test data sets

Model	Task	Box mAP50	Box mAP50-95	Mask mAP50	Mask mAP50-95	Inference (ms)
train6 (yolov8n.pt)	Detect	0.418	0.253	N/A	N/A	53.7
train4 (yolov8n-seg.pt)	Segment	0.527	0.352	0.481	0.367	179.2

Using segmentation test data set (larger one)

Model	Task	Box mAP50	Box mAP50-95	Mask mAP50	Inference (ms)
Detection (train6)	detect	0.620	0.288	N/A	132.7
Segmentation (train4)	seg	0.527	0.352	0.481	179.2