Object Localization of Concentrated Animal Feeding Operations

Minnie Ho[1], Jorge Troncoso[2]

[1]minnie.ho@intel.com, Intel Corporation
[2]jatron@google.com, Google

Stanford
CS 230 Final Project

Objective

- Gather Statistics on Concentrated Animal Feeding Operations (CAFOs) from satellite (aerial) imagery.
- Classify and Localize CAFOs with a bounding box.

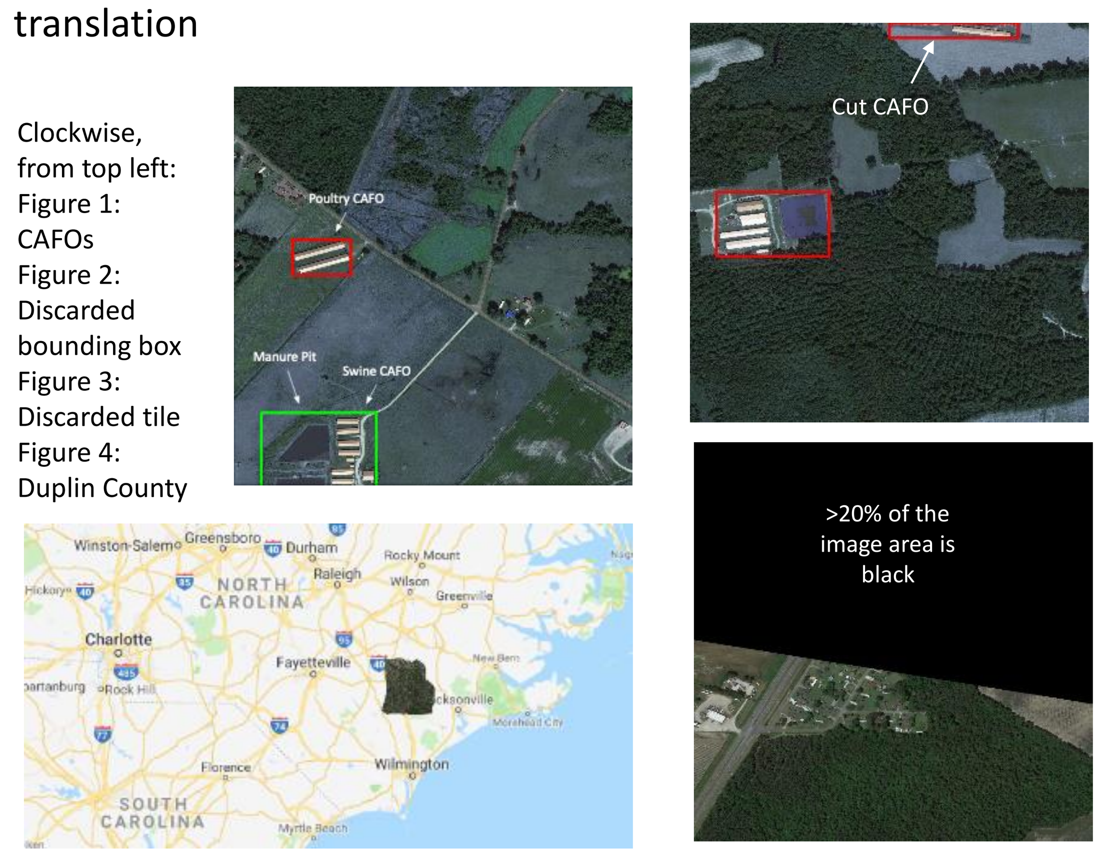
Motivation

- Estimated 60% of CAFOs within the US do not hold permits.
- CAFOs produce 40% of US livestock and generate 335 million tons of waste per year (potential ground-water contamination)

Dataset

- High-resolution satellite images in geotiff format of Duplin County, North Carolina (300 poultry, 600 swine) and Lagrange County, Indiana (13 swine, 167 poultry, 29 cattle). County areas: ~2110 km^2 and ~1000 km^2. CAFO areas: range from 115x20 m^2 poultry to 355x290 m^2 swine.
- Tiled large images into 1024 x 1024 pixel, non-overlapping, consecutive "tiles" (1 m per pixel resolution). Discarded partial CAFO pieces (cut CAFO) with < 20% of CAFO area from training set. Discarded tiles with > 20% black pixels.
- Balanced tile "classes" (equal numbers of CAFO and no CAFO tiles) by under-sampling majority classes. 65/15/20 split for training/validation/test. Test data included same balance (CAFO:no CAFO) as seen in practice.
- Data augmentation using left-right flip, up-down flip, rotation, shear, scale, color (sat/value), and translation

Clockwise, from top left: Figure 1: CAFOs Figure 2: Discarded bounding box Figure 3: Discarded tile Figure 4: Duplin County



Selected References for Poster

[1] Redmon, J., & Farhardi, A. (n.d.). YOLOv3: An Incremental Improvement. arXiv:180 https://pjreddie.com/darknet/
[2] Pytorch Yolov3 software developed by Ultralytics LLC, https://github.com/ultralytics/yolov3
[3] Karen Simonyan, Andrea Vedaldi, and An-drew Zisserman.Deep Inside Convolutional Networks: Visualising Image Classi-fication Models and Saliency Maps. 2013.arXiv:1312.6034 [cs.CV]

Acknowledgments

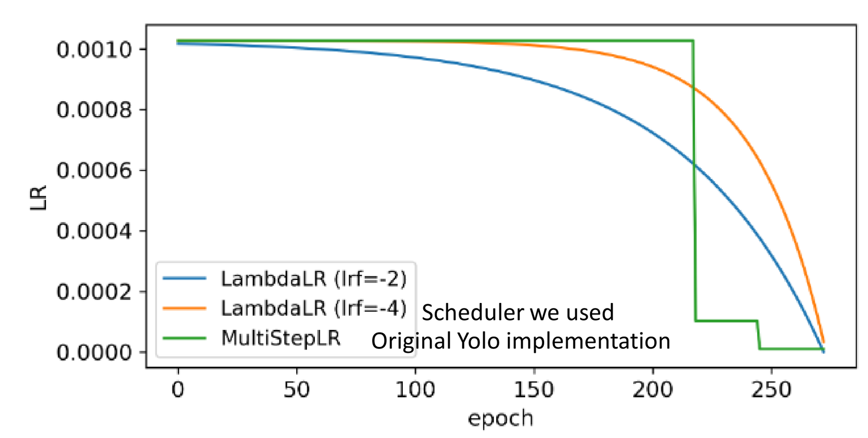
Professor Dan Ho, Brandon Anderson, and Ben Chugg from Stanford Law School. Mohamed El-Geish from the CS230 CA Staff.

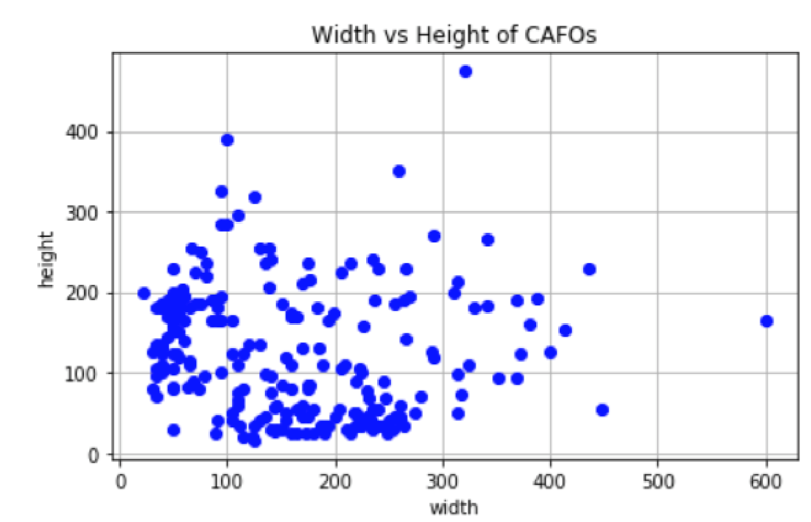
Model and modifications

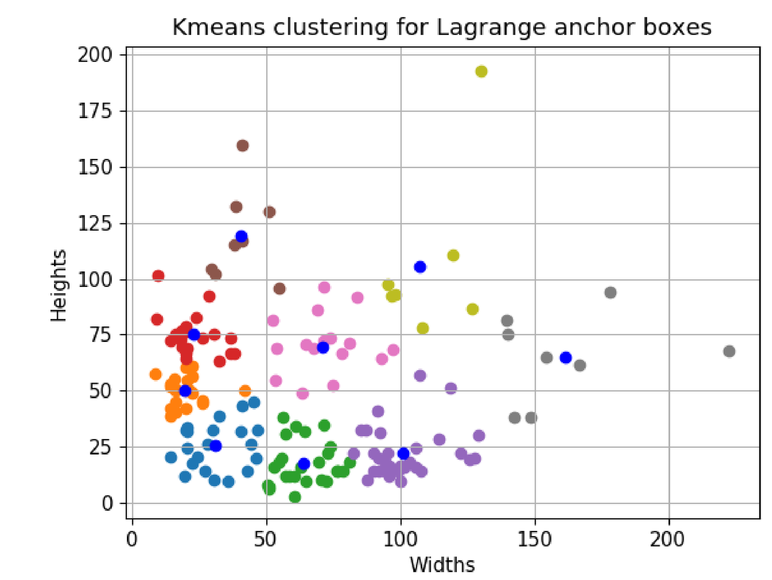
- Yolov3 as baseline algorithm [1], [2], initialized with ImageNet Yolo weights.
- K-means clustering from CAFO training data to determine 9 anchor box widths and heights
- Optimizer: SGD with Nesterov momentum, momentum value = 0.9, staircase scheduler for learning rate
- L2 regularization, λ=0.0005
- Total Loss = Loss (localization) + Loss (classification) = w_{giou} (1-GIOU) + w_{cls} nn.BCELogitsLoss
- MAP computed by integrating area under Precision-Recall curve (101 points)

BCEWithLogitsLoss :

$$l_n = -w_n \left[t_n \cdot \log \sigma(x_n) + (1 - t_n) \cdot \log(1 - \sigma(x_n)) \right]$$







Quantitative Results

- For each experiment (new K-means anchor box vs Yolo boxes, different Loss weights, different IOU thresholds), the following statistics were generated. See below for one such example. We refer to our project report and supplemental material for all experimental results.
- We find that we can significantly improve the test MAP by tuning hyper-parameters. We achieve 0.91 MAP at IOU_t=0.1.
- Saliency maps show that network trained on CAFO images picks up less features than network trained on COCO dataset

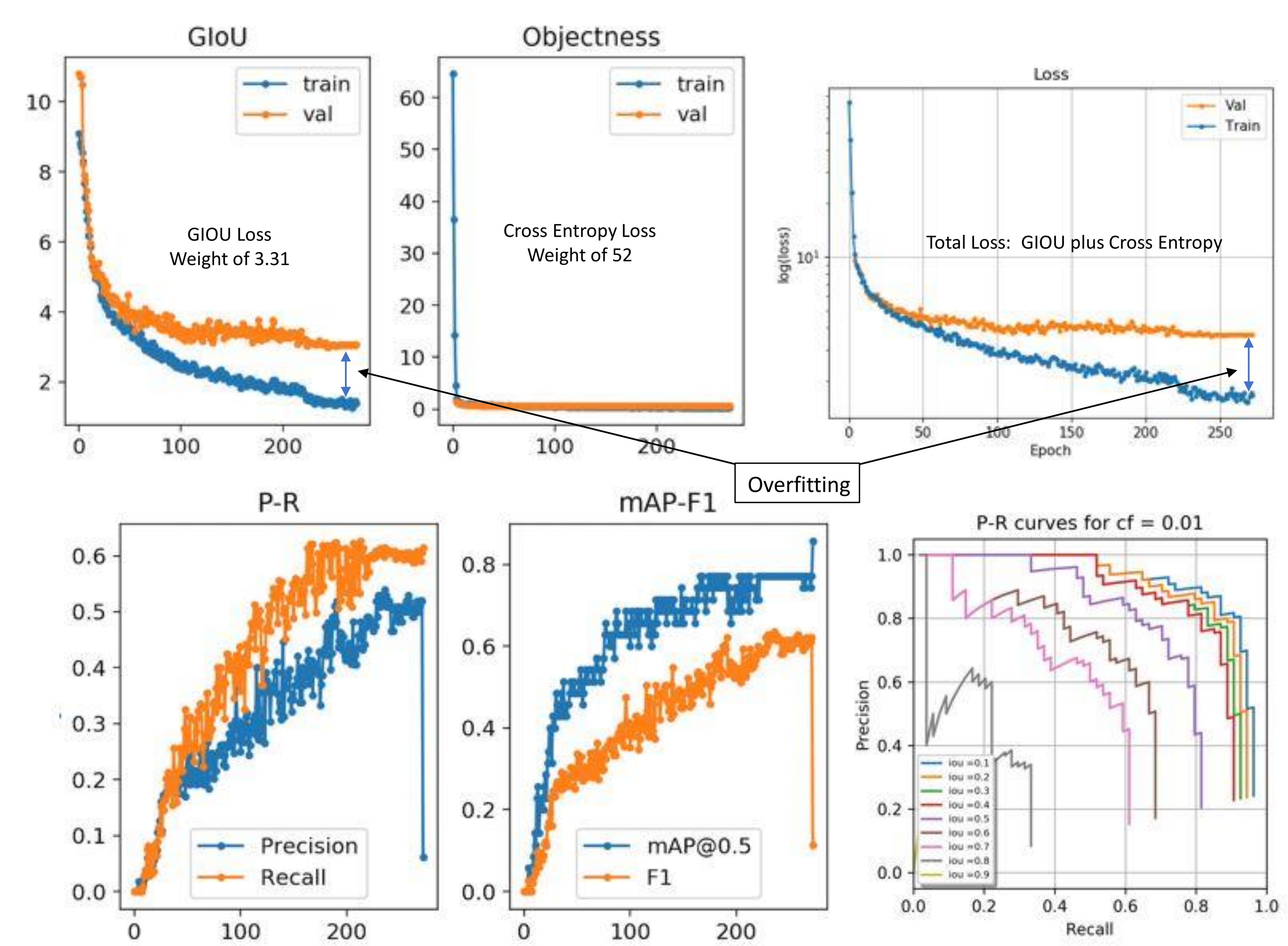
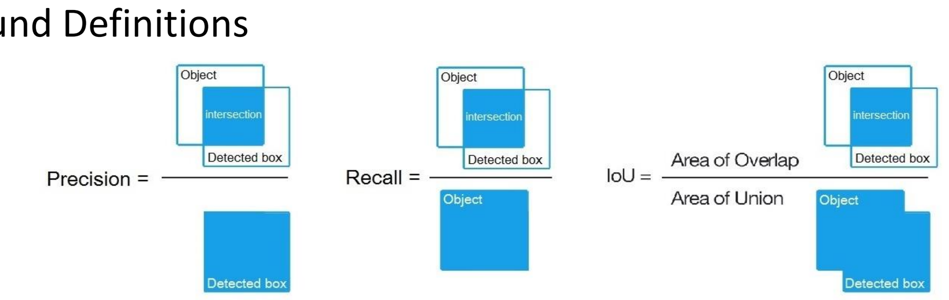


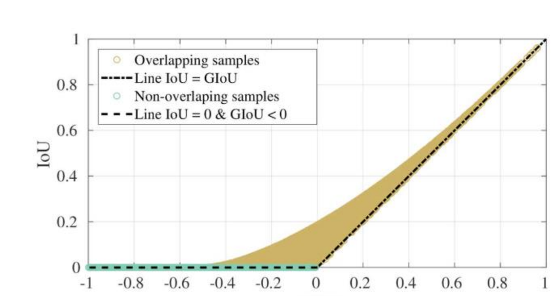
Table of Techniques Tried

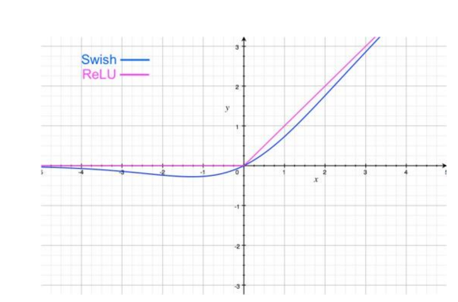
Method (and other parameters)	Baseline MAP	New MAP
IOU Threshold	(IOU_t, MAP) = (0.5:0.259)	(IOU_t, MAP) = (0.213:0.548) (IOU_t, MAP) = (0.1:0.651)
Multi-Scale Training (IOU_t=0.213)	0.548	0.91
Kmeans anchor boxes (IOU_t=0.1)	0.651	0.757
Weighting of sub-losses (IOU_t=0.1, Kmeans anchors)	$(w_{glo}:w_{obj}:MAP)$ = (3.31:52.0:0.757)	$(w_{glo}:w_{obj}:MAP)$ = (0.5:10.0:0.856) (0.5:52.0:0.912)
Swish vs. RELU (IOU_t=0.1, Kmeans anchors, weighting of sub-losses)	RELU MAP = 0.912	Swish MAP = 0.856

Background Definitions

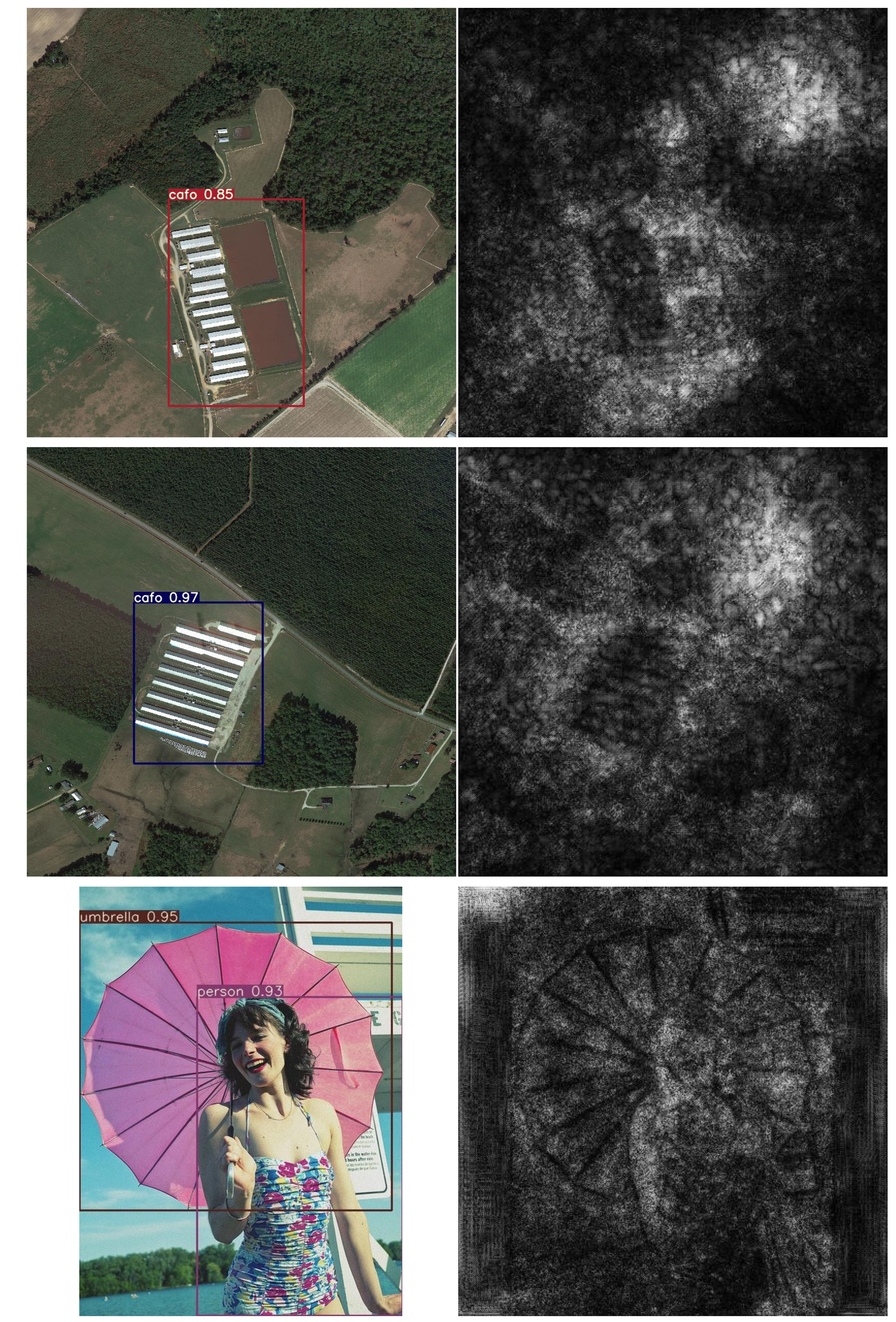


$$F1 = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$



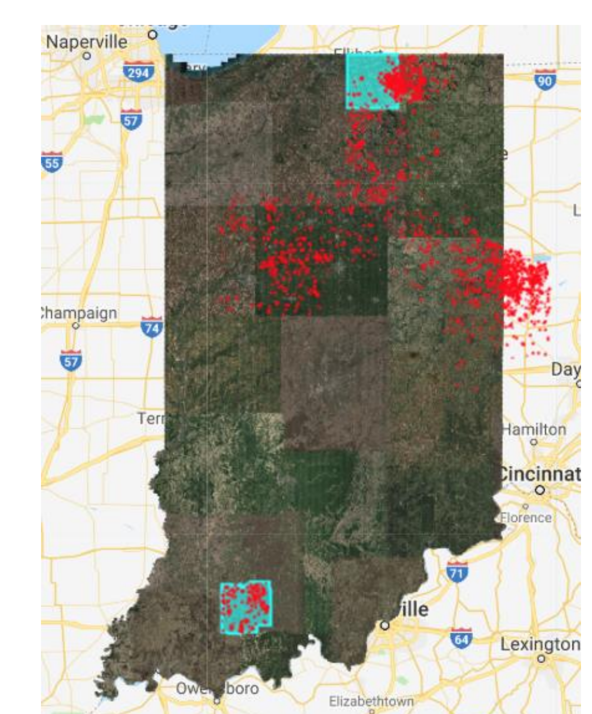


Qualitative Results



Conclusions

- MAP@0.1 improved from 0.259 to 0.912 with tuning.
- Techniques most helpful (discard bad tiles and partial CAFOs, K-means clustering for anchor boxes, decreasing IOU threshold, changing weights of loss functions).



Future Work

- Using full data set over multiple classes, counties, and states.
- Branch-and-search random hyper-parameter tuning
- Other object detectors (RetinaNet, mask-RCNN) to improve performance