Small Object Detection in Satellite Imagery

Stanford University

Claire Huang, Derek Huang, Edward Lee

Motivation

- State-of-the-art object detectors perform with half the average precision on small objects
- Existing methods for addressing small objects, class imbalance increase training overhead
- We focus on satellite images in the xView dataset (over 1 million objects, 60% < 1024 px^2)

Baselines

- 2 general categories exist: single- and double-stage detectors
- We chose to experiment with one model from each: YOLOv3 and Faster R-CNN

YOLOv3 [1]

- 1 stage: Darknet feature extractor, fixed grid detectors at 3 scales
- Anchor boxes help each detector specialize
- Worse accuracy, fast detection

Faster R-CNN [2]

- 2 stages: RPN (Region Proposal Network) and RoI (Fast R-CNN-like)
- Each stage has 2 losses: classification and bounding box
- Better accuracy, slower performance

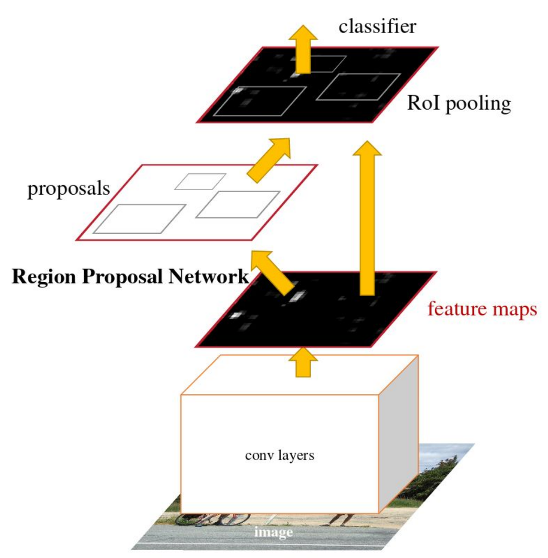

Figure 1: Faster R-CNN structure

[1] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018.
[2] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

Methods

- Make low-overhead changes to existing models to improve performance on small objects
- Set dataset split, augmentations, training schedule for all experiments (per baseline)

Loss Function Changes

- Change weights of different tasks of the multi-task loss functions
  - tasks: bounding box, classification, confidence
- Change weights for each data point
  - focal loss, reduced focal loss, area-based weights

$$\text{AW-d}(c, A) = \begin{cases} 1 & A \leq t_a \\ \frac{1}{c} & A > t_a \end{cases} \qquad \text{AW-p}(c, A) = \begin{cases} 1 & A \leq t_a \\ \left(\frac{t_a}{A}\right)^c & A > t_a \end{cases} \qquad \text{AW-c}(c, A) = \left(\frac{t_a}{A}\right)^c$$

Figure 2: Different area-based weights we tried.

Anchor Box Changes

- Compute new "small anchors" with k-means clustering: 20 anchors < 1024 px^2 and 10 anchors > 1024 px^2

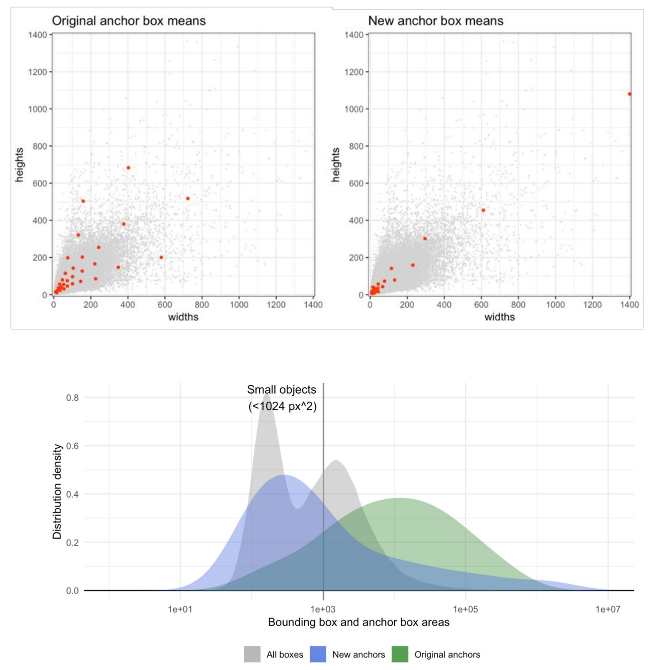

Figure 3: The distribution of **small YOLOv3 anchor box areas** (blue) matches all bounding box areas (gray) better than the original anchor boxes (green).

Results

- Evaluated with AP (VOC-style), $AP_{C,S}$ (on "small" classes), $AP_{A,S}$ (on objects < 1024 px^2)

YOLOv3

- Varied task weights in loss function do not target small object performance
- Small anchor boxes perform well
  - We see small object performance of ~4.5x on small objects
  - Effects diminished when combined with AW-c
- Area-based weights on classification loss perform best
  - Reduced small object false positives by 8.1%
- Combined small anchor boxes + area-based weights

Model	AP	$AP_{C,S}$	$AP_{A,S}$
YOLOv3	0.0118	0.0065	0.00255
small anchors	0.0396	0.0283	0.0111
AW-d	0.0397	0.0134	0.0099
AW-p	0.0663	**0.0324**	0.0108
AW-c	**0.0689**	0.0298	**0.0112**
small anchors + AW-c	0.0514	0.0130	0.0080

Model	AP	$AP_{C,S}$	$AP_{A,S}$
YOLOv3	0.0118	0.0065	0.00255
+ small anchors	0.0396	0.0283	0.0111
+ AW-c	0.0514	0.0130	0.0080

Figure 6: All YOLOv3 results: experiments (left) and ablation (right)

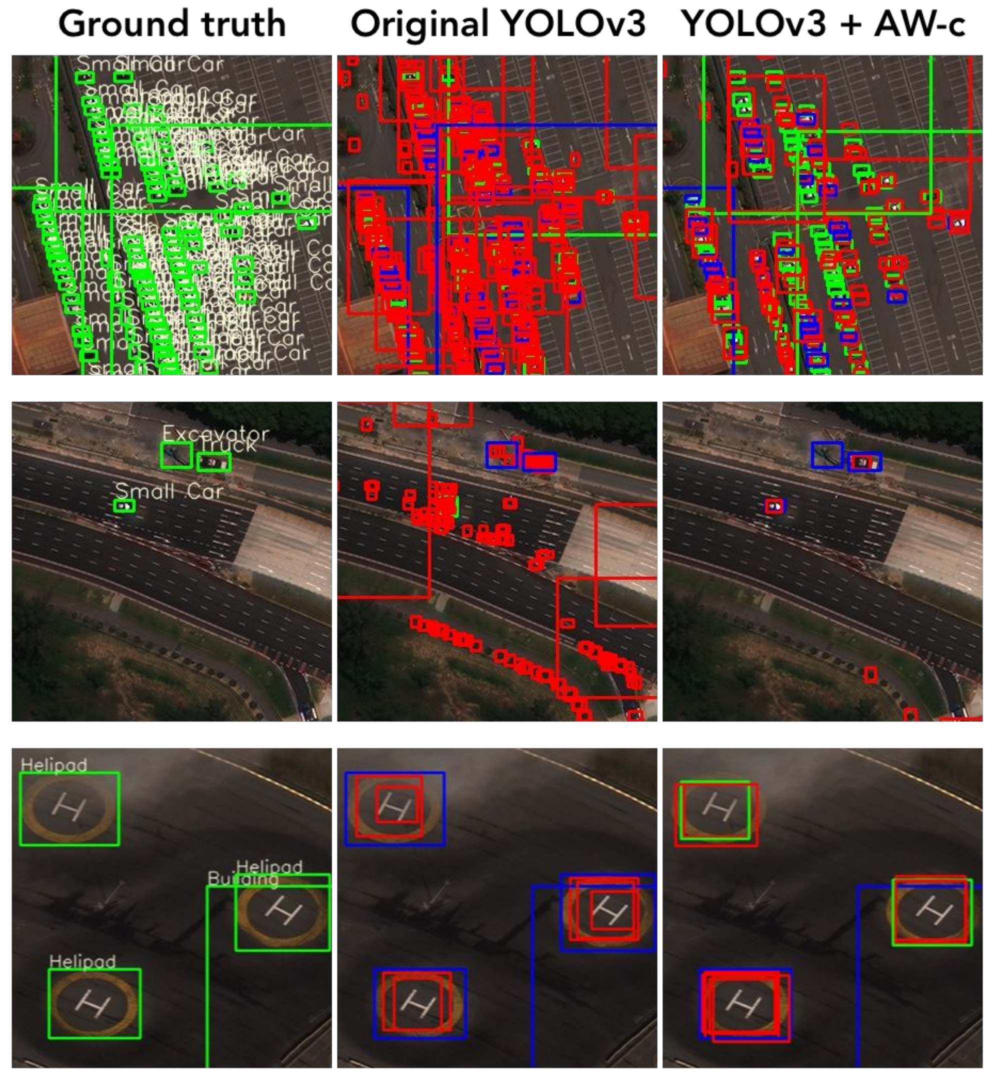

Figure 4: YOLOv3 improvements. Performance on**: (top) dense small objects in** parking lots, (mid) false negatives on roads, (bottom) rare small classes.

RPN	RoI	AP	$AP_{C,S}$	$AP_{A,S}$
CE	CE	**0.2065**	0.0874	0.0716
FL	CE	0.2036	**0.0890**	0.0671
FL	FL(sig)	0.1285	0.0573	0.0359
RFL	CE	0.2064	0.0889	**0.0744**
RFL	RFL(sig)	0.1364	0.0656	0.0407
RFL	RFL(sft)	0.1533	0.0764	0.0485

RPN	RoI	β	AP	$AP_{C,S}$	$AP_{A,S}$
CE	CE	0.5	**0.2065**	0.0874	0.0716
ACE	CE	0.5	0.2039	0.0941	0.0718
ACE	ACE	0.5	0.1856	**0.0970**	**0.0786**
ARFL	ACE	0.5	0.1827	0.0817	0.0693
ARFL	ARFL	0.5	0.1255	0.0683	0.0453

RPN	RoI	β	AP	$AP_{C,S}$	$AP_{A,S}$
ACE	ACE	0.0	0.2065	0.0874	0.0716
ACE	ACE	0.125	**0.2091**	0.0946	0.0806
ACE	ACE	0.25	0.1993	0.0931	**0.0849**
ACE	ACE	0.5	0.1856	**0.0970**	0.0786
ACE	ACE	1.0	0.1574	0.0879	0.0708

model	AP	$AP_{C,S}$	$AP_{A,S}$
baseline	0.2065	0.0874	0.0716
+ RPN ACE	0.2039	0.0941	0.0718
+ RoI ACE	0.1856	0.0970	0.0786
+ $\beta = 0.25$	0.1993	0.0931	0.0849
+ multi-task weights	0.2023	0.0966	0.0869

Figure 6: All Faster R-CNN Results:
- Focal Loss Variants (1st)
- Area-based weights (2nd)
- **β-selection (3rd)**
- **Final ablation (4th)**

Faster R-CNN

- Adding focal loss and/or reduced focal loss doesn't provide major improvements, even decreasing performance when added to RoI
- Area-based weights do seem to improve performance on cross-entropy, but not reduced focal loss
- β=0.25 with AW-p seems to work well
- Best single-model improves small object AP by 1.5%, while decreasing overall AP by 0.4%
- Errors mostly due to misclassifications and too few objects detected

Figure 7: Common error modes in Faster R-CNN (left to right)
1. ground truth (top) vs. predictions (bottom)
2. object detected as multiple classes
3. too few detections

Conclusions & Future Work

- Due to time and compute constraints, hyperparameters optimized for a short train session that might not have converged fully
- Even still, we are able to improve small object detection significantly over baseline
- Future work could be to using GANs to upscale small image features, or other more significant changes to the architecture