ARUBA: An Architecture-Agnostic Balanced Loss for Aerial Object Detection

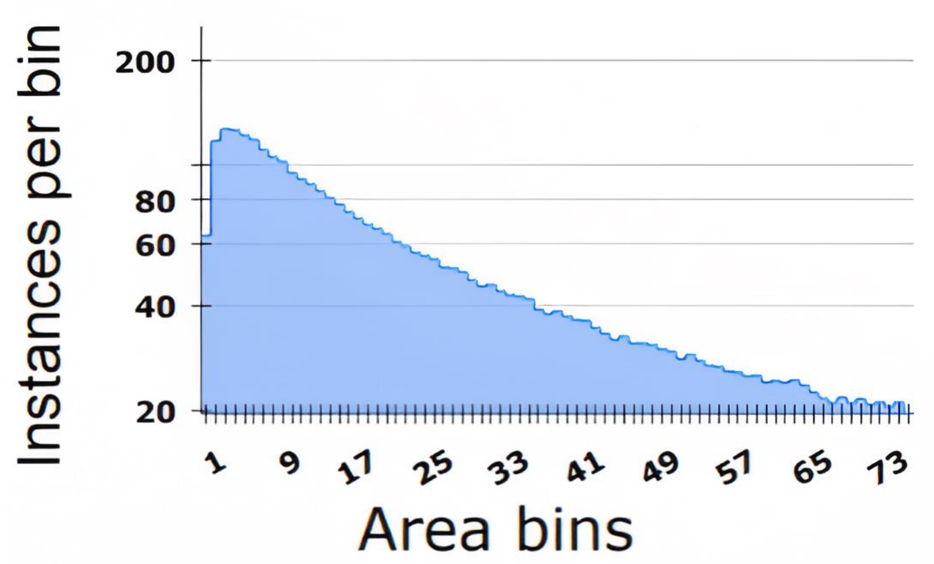
Rebbapragada V C Sairam, Monish Keswani, Uttaran Sinha, Nishit Shah, Vineeth N Balasubramanian
{ai20resch13001, monish.keswani, cs17mtech11003, cs18mtech11020, vineethnb}@iith.ac.in
Indian Institute of Technology, Hyderabad

WACV WAIKOLOA HAWAII JAN 3-7 • 2023

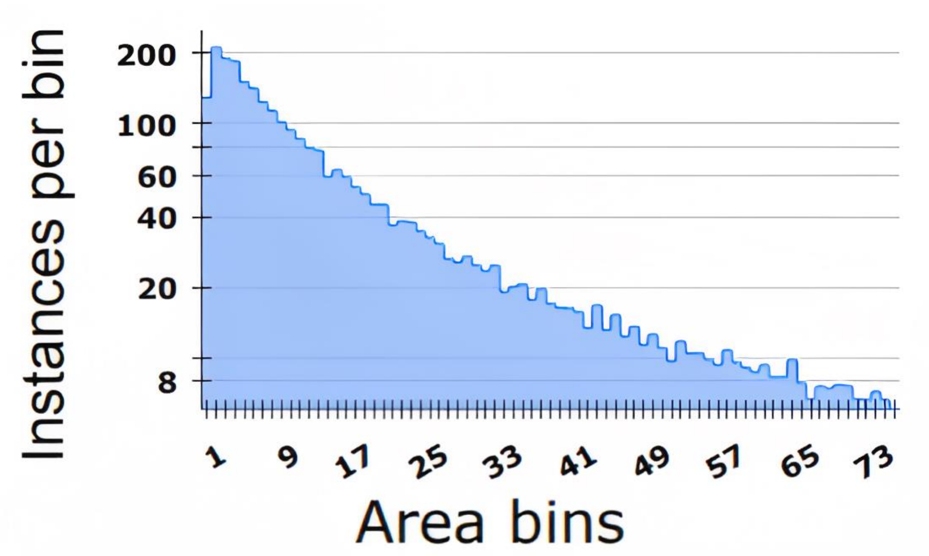
AIM

To solve the severe size-imbalance problem in drone-based aerial image datasets

Size-imbalance - natural vs aerial image datasets



Size imbalance - COCO

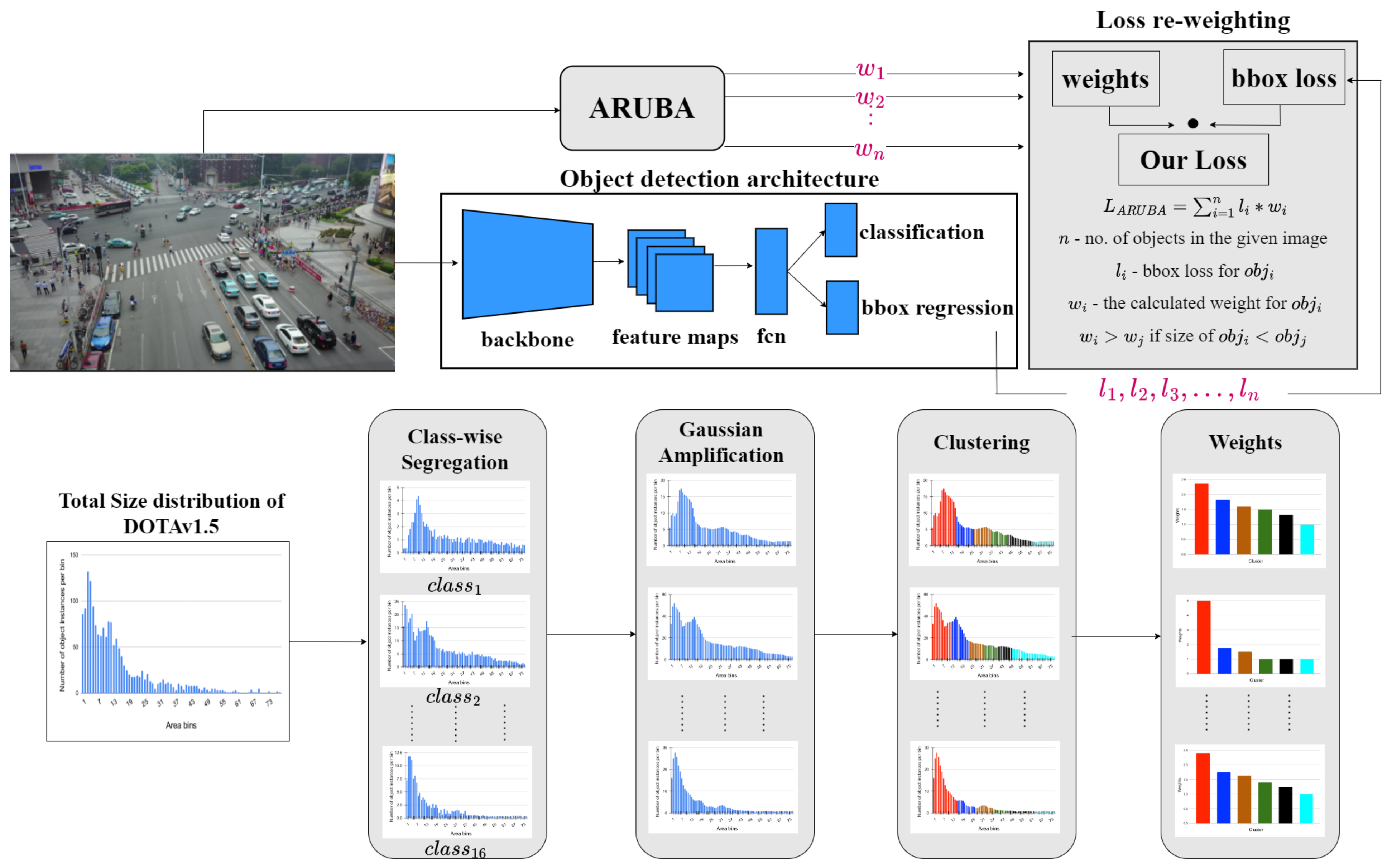


Size imbalance - VisDrone

Contributions

- Novel architecture-agnostic loss reweighting strategy
- First such loss based approach in this domain
- Simple yet effective pipeline based on well-known modules
- Key observations around the ordinality of object size - might be useful in other settings with ordinal categories
- Extensive experiments on multiple aerial image datasets

Proposed Methodology



Effect of neighbourhood

Trained on	Tested on		
	Small	Medium	Large
Small	**33.78**	26.87	1.81
Medium	7.01	**46.01**	15.26
Large	2.56	23.58	**38.91**

Performance of baseline on different size bins of HRSC2016 dataset

Datasets:
HRSC2016, DOTA_v1.0, DOTA_v1.5 & VisDrone

Evaluation metrics:

- For HRSC2016, VisDrone:
  mAP = mean(APs@[.5:.05:.9])
- For DOTA datasets:
  mAP = mean(class-wise APs@50)

Loss re-weighting strategy

Size-balanced loss based on the size of the objects in each class:

$$L_{ours} = w_{ys} * L_{reg}(\acute{b}, b)$$

w_{ys} - weight for an object of size s and class y

Gaussian Amplification: To add the context of size neighbourhood

$$B^c = (b_1^c, b_2^c, \ldots, b_m^c)$$

$$K_w = (K_{-\frac{w}{2}}, \ldots, K_{-1}, K_0, K_1, \ldots, K_{\frac{w}{2}})$$

B^c - size distribution of class c, K_w - Discrete gaussian kernel of width w

$$GA(b_k^c) = \sum_{i=-w/2}^{w/2} k_i * b_{k+i}^c$$

b_k^c - size bin in consideration, k_i - corresponding kernel entry

Effective number of objects of size s and class y: $E_{ys} = \dfrac{1 - \beta^{\left(\sqrt[n]{GA(n_{ys})}\right)}}{1 - \beta}$

Our loss function is given by : $L_{ours} = L_C + w_{ys} * L_R$

where $w_{ys} = 1 - 1/E_{ys}$

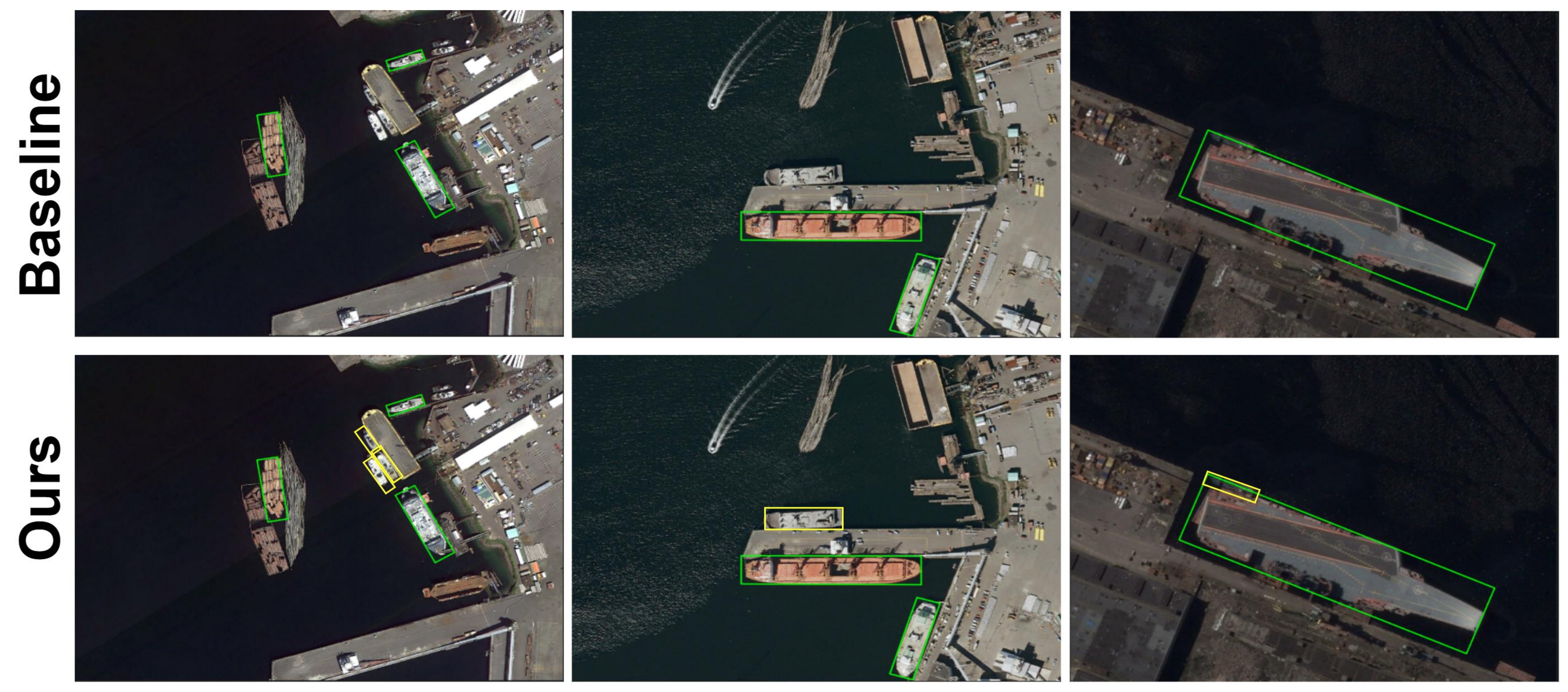
Results

Trained on	Method	mAP
HRSC2016	ReDet	70.41
	Ours + ReDet	**72.42**
DOTA_v1.0	ReDet	76.15
	Ours + ReDet	**77.14**
DOTA_v1.5	ReDet	66.86
	Ours + ReDet	**68.71**
VisDrone	ReDet	18.80
	Ours + ReDet	**20.32**

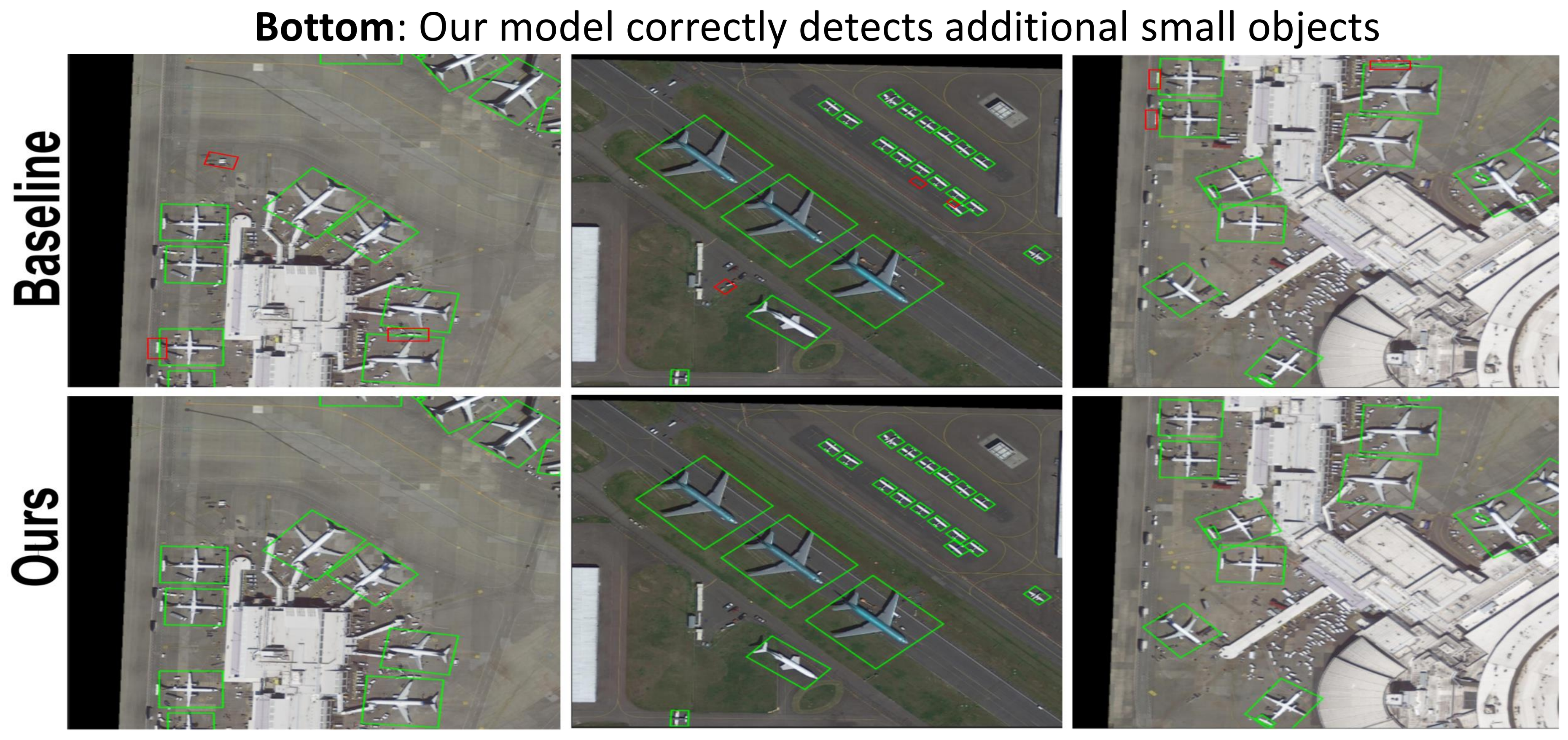
Performance comparison – Baseline vs our model on four different aerial object datasets

Trained on	Method	Tested on		
		small	med	Large
HRSC2016	ReDet	17.93	29.58	38.91
	Ours + ReDet	**20.79**	**29.97**	38.01
DOTA_1.0	ReDet	09.74	23.48	52.44
	Ours + ReDet	**11.81**	23.34	52.24
DOTA_1.5	ReDet	08.32	24.85	43.56
	Ours + ReDet	**10.65**	24.76	43.52
DOTA_1.0	S2aNet	10.64	24.93	47.43
	Ours + S2aNet	**12.48**	**25.57**	**47.85**

Performance comparison – Baseline vs our model on small, medium and large sized objects



Top: Baseline fails to detect few objects
Bottom: Our model correctly detects additional small objects



Top: Baseline predicts many false positives
Bottom: Our model reduces FPs because of the effective ARUBA loss