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

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

Abstract

*Deep neural networks tend to reciprocate the bias of their training dataset. In object detection, the bias exists in the form of various imbalances such as class, background-foreground, and object size. In this paper, we denote size of an object as the number of pixels it covers in an image and size imbalance as the over-representation of certain sizes of objects in a dataset. We aim to address the problem of size imbalance in drone-based aerial image datasets. Existing methods for solving size imbalance are based on architectural changes that utilize multiple scales of images or feature maps for detecting objects of different sizes. We, on the other hand, propose a novel **AR**chitect**U**re-agnostic **BA**lanced Loss (**ARUBA**) that can be applied as a plugin on top of any object detection model. It follows a neighborhood-driven approach inspired by the ordinality of object size. We evaluate the effectiveness of our approach through comprehensive experiments on aerial datasets such as HRSC2016, DOTAv1.0, DOTAv1.5 and VisDrone and obtain consistent improvement in performance.*

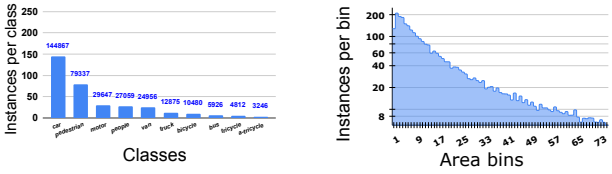
1. Introduction

In recent years, drones have shown immense potential in numerous disciplines. In military warfare, they can be used as target decoys for combat missions. In agriculture, drones provide farmers with real-time data to make informed harvesting decisions. For search-and-rescue, they can reach places where humans cannot. Alternatively, they are also used in fire-fighting, delivery of essentials and aerial photography. This increasing demand for drones in various domains has recently encouraged the computer vision community to work extensively on vision from drones [3].

Deep neural networks have led computer vision research and development for a decade now on multiple challenging problems such as semantic segmentation, object detection/tracking, as well as image classification. In object detection, methods like FasterRCNN [27], YOLO [25], RetinaNet [15] and its variants have achieved decent performance on many challenging datasets. With increased interest and creation of datasets in drone-based imagery, aerial object detection [34, 6] has gained a lot of interest from the research community. Although the aforementioned methods exhibit exceptional performance on popular general object detection datasets such as MSCOCO [16], aerial-object datasets [34, 6] pose more challenges, even to state-of-the-art object detection models.

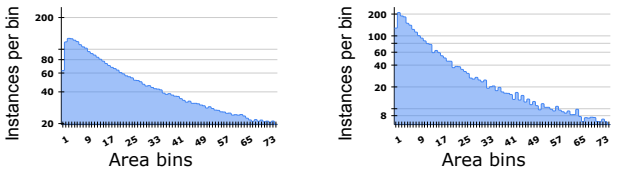
Figure 1: Predictions on an image from VisDrone dataset [6] with Focal loss [14] vs Ours. **Top:** Focal loss fails to detect many objects. **Bottom:** Ours is able to recognize additional objects, including small ones, because of our **AR**chitect**U**re-agnostic **BA**lanced (**ARUBA**) loss. Yellow boxes indicate objects additionally detected.

High variation in scale and orientation of objects in aerial datasets, especially from drone images, make detecting these objects quite challenging. Specialized methods [8, 37] have been proposed to capture the oriented bounding boxes efficiently. An added difficulty in aerial datasets [34, 6, 19] is that they are highly skewed in their object size distribution in addition to the class distribution (as shown in Figure 2. Note that in Figure 2b, x-axis shows the object area bins where the size of the objects increases from left to right and y-axis shows the number of object instances per an area bin). We also observe that size imbalance is severe in aerial object datasets when compared to more general-purpose object detection datasets (refer Figure 3), which motivates us to address this imbalance problem of drone-based aerial datasets in this work.



(a) Class Imbalance (b) Size Imbalance

Figure 2: Highly skewed class and size distributions in VisDrone dataset



(a) Size Imbalance-COCO (b) Size Imbalance-VisDrone

Figure 3: Comparison of size imbalance severity between general and drone-based aerial object datasets. Note that y-axis is log of frequency, hence the effect is exponential in terms of occurrence.

Size imbalance is a common problem in object detection datasets, and many methods have been proposed to mitigate this issue, as summarized in [20]. Existing methods [13, 18, 28] have largely proposed architectural modifications to enhance the model's ability to view objects at different scales. However, such multi-scale approaches arise from careful engineering of architectures to suit a specific domain or setting. In this work, we propose to address the size imbalance problem from an architecture-agnostic balanced loss perspective. One could also view our approach as a long-tailed perspective to a size balance problem, unlike the class imbalance setting that is typically studied in long-tailed detection/recognition problems.

Long-tailed object detection methods typically focus on datasets with skewed class distribution, to improve performance on detecting and classifying minority classes. Many methods [10, 2, 32, 31, 5, 30, 24] have been proposed to tackle this problem from a class imbalance perspective (summarized in Sec 2). We focus on the idea of using loss-reweighting [5, 30, 24] wherein higher weights are assigned to tail classes. Unlike class labels, size (when distinguished as large-to-small) is an ordinal variable making it non-trivial to apply existing solutions for class imbalance to size. Besides, as shown in Figure 2b, small-sized objects are dominant in drone-based aerial datasets and large-sized objects are sparse. Although large-sized objects are the tail, they have larger spatial support which can provide richer and more useful features compared to small objects which can make it helpful to detect them. On the other hand, learning small-sized objects, although the majority in such datasets, is challenging, even for state-of-the-art detection models [27, 25, 15]. The increasing use of drone images and the lack of a consistent method for detection of objects of different sizes in such datasets motivates us to solve the severe size imbalance in such aerial datasets. In summary, we address the long-tailed size imbalance issue in drone-based aerial datasets rather than the long-tailed class imbalance issue that is typically addressed in earlier related efforts.

To this end, we propose a novel architecture-agnostic loss-reweighting strategy which considers the ordinality of the size variable in its design. The performance of an object detection model on instances of a given size would have a contribution from object instances of neighboring sizes. For example, given a particular class, a model learned on object instances of area X is more likely to recognize an instance of area $X \pm \delta$ rather than $X \pm k\delta$, where k is a large integer. We hence apply a Gaussian amplification on the size distribution to consider the effect of such neighborhood instances (as detailed in Section 3). We subsequently use a clustering approach to assign weights to object instances based on their sizes. Finally, inspired by previous balanced loss work which focus on class imbalance [5], we reweight the loss based on size clusters to suit our problem. Unlike existing methods for long-tailed class imbalance which assign lower weights to head categories, our method assigns higher weights to the head categories (small-sized objects) ensuring that the model learns better on them. We show that the size-imbalance problem can be addressed using such a loss-based approach without the need for time-consuming architecture engineering. To summarize, our key contributions are as follows:

- We propose a novel architecture-agnostic loss-reweighting strategy to solve the severe size imbalance issue in drone-based aerial image datasets. We call this **AR**chitect**U**re-agnostic **BA**lanced Loss (**ARUBA**), which can be applied while training any object detection model.
- To the best of our knowledge, this is the first such loss-based approach to handle size imbalance in this domain. Our key observations around the ordinality of the considered categories and the connection of such ordering to a model's performance may be useful in other settings with ordinal categories (e.g. class labels of a disease with increasing severity levels).
- We propose a simple yet effective pipeline based on well-known modules to achieve the objectives using our loss-reweigting strategy. Our extensive experimental results corroborate the usefulness of this pipeline.
- We perform a comprehensive suite of experiments on multiple drone-based aerial image datasets including HRSC2016, DOTA-v1.0, DOTA-v1.5 and VisDrone to validate the effectiveness of our proposed approach. We also provide additional ablation studies and qualitative results to illustrate the usefulness of the proposed method to handle size imbalance in this domain.

2. Related Work

We describe prior work from different related perspectives individually below.

Aerial Object Detection. Compared to the general object detection [17], aerial object detection requires special attention because of the additional challenges like high variation in orientation of the objects. Specialized methods [35, 7, 8] have been designed for detecting oriented bounding boxes in such aerial image datasets. R3Det [35] proposed a feature refinement module for accurate features, thereby improving performance. S^2aNet addressed the issue of misalignment between anchor boxes and axis-aligned convolutional features by proposing Feature Alignment Module and Oriented Detection Module. Recently, ReDet [8] encoded rotation equivariance and rotation invariance by incorporating rotation-equivariant networks. However, all these methods use architecture-based approaches, as mentioned earlier. We instead propose a loss-based approach to address this problem. We, in fact, make use of the abovementioned methods as baselines and show that our loss re-weighting strategy achieves improvement in performance when applied on top of them.

Size Imbalance. There have been fewer efforts that have explicitly addressed size imbalance in object detection as summarized in [20]. These approaches typically depend on using multiple scales of images, feature maps or both to detect objects of different sizes. Methods like SSD [18] and Scale-aware Fast-RCNN [11] make predictions from multiple layers of feature maps and combine them. Feature Pyramid Networks [13] and its variants aggregates features from multiple layers before performing prediction. Image pyramid-based methods like SNIP [28] and SNIPER [29] use multiple scales of images rather than features for detecting objects of different sizes. [23, 12] combines the advantages of both feature pyramids and image pyramids. The idea behind these methods is to improve the performance by processing at multiple scales. We, on the other hand, exploit the long-tail imbalance of the size distribution by proposing a loss-reweighting strategy for this challenge.

Long-tailed Object Detection. Existing efforts on long-tailed imbalance generally focus on class imbalance and are divided into three categories: sampling-based, data generation and re-weighting based methods. We describe each of them below.

Sampling based methods: Sampling-based approaches rely on data manipulation techniques such as under-sampling and over-sampling. Works such as [10, 22, 2] utilize sampling-based methods to balance background-foreground and class labels in the dataset.

Data generation methods: These methods generate objects of minority classes synthetically using data generation methods such as Generative Adversarial Networks and data augmentation [32, 31]. Unlike oversampling, this approach does not repeat data samples and thus reduces over-fitting. However, the performance of these methods is contingent on quality of the samples generated.

Re-weighting based methods: Re-weighting methods formulate the training objective of a model based on the statistics of the class-imbalanced dataset. [5] balance the loss based on the effective number of instances per class. [30] ignore the discouraging gradients for the rare categories from majority categories. [24] alleviate the class-imbalance problem by posing it as a ranking problem.

These approaches tackle the long-tailed imbalance problem from a class perspective. However, we tackle this problem from a size perspective.

Regression Imbalance. One of the works closest the present work is DIR [36] which focuses on imbalance in continuous targets in general rather than categorical. We focus on continuous targets specific to object detection and propose a framework to mitigate the issue of object size imbalance, which is different from their focus.

3. Architecture-Agnostic Balanced Loss

As stated earlier, the proposed **ARUBA** (**AR**chitect**U**re-agnostic **BA**lanced) loss is designed to address the problem of severe size imbalance in drone-based aerial object datasets. To formulate **ARUBA**, we begin by discussing the loss re-weighting strategy used in general long-tailed class imbalance methods [14, 5, 30]:

$$CB(\mathbf{p}, y) = w_y * \mathcal{L}_{cls}(\mathbf{p}, y) \tag{1}$$

where w_y is the weight for a class y, $\mathbf{p}$ is the predicted class probability and $\mathcal{L}_{cls}$ is the classification loss. We instead propose herein a size-balanced loss based on the size of the objects within a class as follows:

$$\mathcal{L}_{ours} = w_{ys} * \mathcal{L}_{reg}(b', b) \tag{2}$$

where w_{ys} is the weight for an object of size s belonging to a class y; b' and b are the predicted and ground-truth bounding boxes, and $\mathcal{L}_{reg}$ is the regression loss. The idea of our re-weighting strategy is to assign higher weights to small-sized objects because they have poor spatial support making it difficult to detect them. Note that size being an ordinal variable does not have strict partitions like class categories. Learning to detect objects of a given size does imbue a model with the capability to detect objects of similar sizes (at least partially, as also shown in Table 1 and explained in the next paragraph). Also, objects in a dataset may have a large variety of sizes, unlike a fixed number of classes. These differences between categorical and ordinal

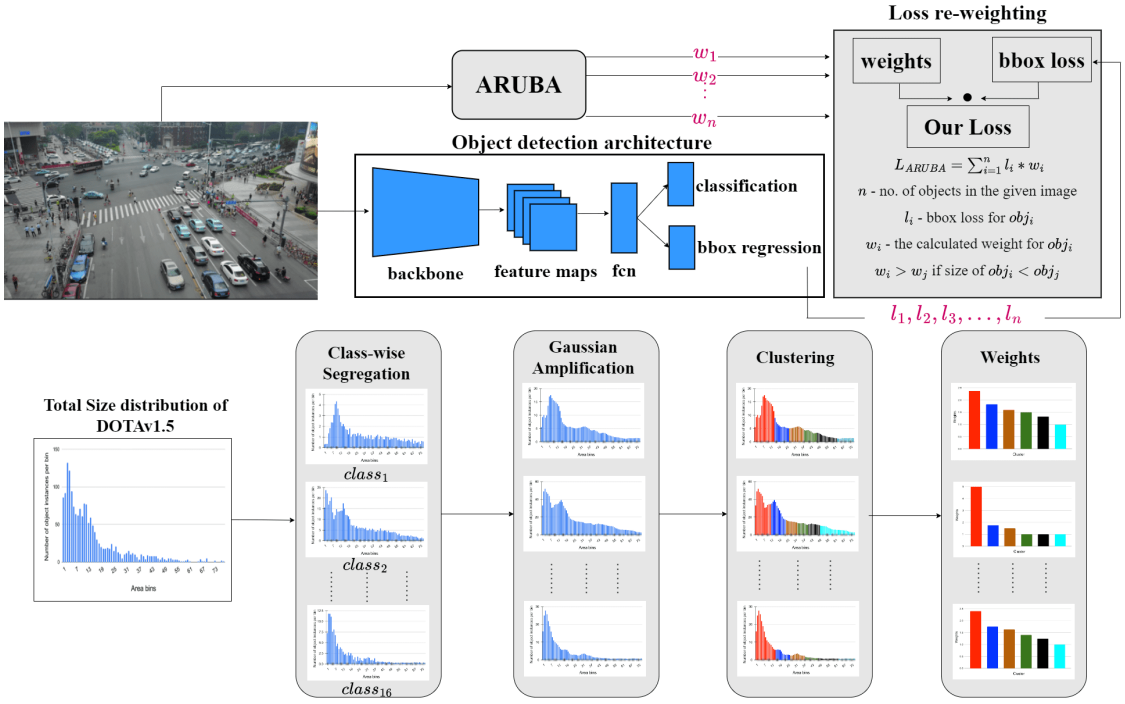


Figure 4: **Overview of proposed method: (a)** The top figure shows how our method is architecture-agnostic. Independent of the object detection architecture, ARUBA calculates weights for objects based on their sizes. **(b)** The bottom figure details the ARUBA pipeline comprised of four stages. We use the size distribution of the DOTA_v1.5 dataset for visualization.

Trained on	Tested on		
	Small	Medium	Large
Small	**33.78**	26.87	1.81
Medium	7.01	**46.01**	15.26
Large	2.56	23.53	**49.21**
All	17.93	29.58	38.91

Table 1: Performance of baseline on different size bins of HRSC2016 dataset. The train and test sets are divided into three bins - Small, Medium, and Large. ALL bin means we consider the entire train data.

variables make it non-trivial to directly apply long-tailed class re-weighting strategies to solve size imbalance. To address these differences, we propose a pipeline of steps, which are simple and well-known, to address size imbalance: class-wise segregation, followed by Gaussian amplification and then clustering. The details of each of these modules are provided in subsequent sections. Figure 4 outlines our overall pipeline.

Effect of neighborhood. Before describing each of the components in our pipeline, we first show the effect of the ordinality of the size variable through a study. In particular, we discuss the effect of neighborhood on adjacent size bins by experimenting on the HRSC2016 dataset which only has a single class, Ship. We divide both train and test data into three kinds: *small*, *medium* and *large*, based on the object sizes. Table 1 summarizes the results (Average Precision values) of a recent aerial object detection method, ReDet [8], trained and tested on these categories of objects. We note that the model trained on the *small* train bin (bin containing small-sized objects) performs well on the small test bin, and its performance reduces as we move from small test bin to large. Similarly, the performance of a model trained on the large train bin decreases as we move from large test bin to small. The influence of ordinality of the size categories on model performance is evident in these results. We leverage this neighborhood effect by using a Gaussian amplification process that we describe later in this section.

Class-wise segregation. As shown in Figure 4, the first stage of our overall pipeline is the class-wise segregation. Our empirical studies presented in the supplementary section suggest that the effect of neighborhood should be considered within a class rather than across-classes. We hence segregate the size distribution class-wise and deal with size imbalance within each class separately (shown as the first stage in Figure 4b).

Gaussian Amplification. We apply a Gaussian amplification on the size distribution of each class to add the context of the size neighborhood. Similar to Label Distribution Smoothing in DIR [36], we use kernel density estimation to achieve our objective. For each class, we convolve a 1D-Gaussian kernel with the size distribution to obtain a

smoothed and amplified distribution. We denote the size distribution of class c as B^c and a discrete Gaussian kernel with window size w as K_w. They are defined as follows:

$$B^c = (b_1^c, b_2^c, ...b_m^c) \tag{3}$$

$$K_w = (k_{-w/2}, ...k_{-1}, k_0, k_{+1}, ...k_{+w/2}) \tag{4}$$

We design the above discrete Gaussian kernel with certain properties: **1.** It is an odd-symmetric kernel. **2.** The peak of the kernel, k_0, is always one. We divide the kernel by its maximum value to achieve this. **3.** It has two hyperparameters, namely, window size w and variance σ. w is the width of the Gaussian kernel, i.e. it specifies the number of bins ($b_{-w/2}$ to $b_{w/2}$) that we want to consider from the neighborhood. σ specifies the importance that we give to each bin while considering the neighborhood. By increasing σ, we increase the weight given to each neighboring bin.

We thus define Gaussian Amplification, GA, as follows:

$$GA(b_k) = \sum_{i=-w/2}^{w/2} k_i * b_{k+i} \tag{5}$$

where b_k refers to the size bin in consideration and k_i is the corresponding entry of the Gaussian kernel. For the extremities, the convolution is zero-padded accordingly. Unlike Gaussian smoothing, which can result in reduction of the bin values at times, our procedure always results in amplification by design of the Gaussian filter. We hence call it Gaussian Amplification. For better understanding of its functioning, please go through the example provided in the supplementary section.

Clustering. Objects in a dataset usually can be of a wide variety of sizes. One way of categorizing the size distribution is to simply consider each size as a different category. However, this may result in too many size categories. We divide the objects into multiple equal-sized bins before applying Gaussian amplification to accommodate the effect of neighborhood. However, owing to the large number of object instances in aerial object datasets [34, 6], this results in a large number of bins. This makes the step of weighting the loss terms of each bin tedious. In order to make the loss reweighting step more feasible, we cluster the instance area distribution (after Gaussian amplification) into a fixed number of clusters, which we can then reweight. In this work, we use a simple k-means approach for clustering the distribution into k clusters. Figure 4b shows an illustration of the size distribution after clustering the data. As we can observe, objects are grouped as per their sizes. Small-sized objects are clustered together and large-sized objects are clustered together with some intermediate-sized clusters in the middle. We found in our empirical studies that this step provided significant control over the reweighting strategy than merely using equal-sized bins.

Loss function. We now describe the actual size-balanced loss itself. As explained earlier, the differences between ordinal and categorical variables make it non-trivial to apply the existing loss re-weighting strategies used for long-tailed class imbalance in solving size imbalance. We bridge this gap by considering the effect of neighboring sized object instances and forming object clusters based on their sizes. This allows us to obtain weights based on size cluster frequencies. Inspired by [5], we define effective number of object instances of a class y belonging to a size cluster s as:

$$E_{ys} = \frac{1 - \beta^{GA(n_{ys})}}{1 - \beta} \tag{6}$$

where GA refers to the *Gaussian Amplification* process as in Eqn 5, n_{ys} is the number of objects of class y in size cluster s, and $\beta \in [0, 1)$ is a hyperparameter as defined in [5], it controls how fast E_{ys} grows as cluster size s increases. Depending on the size of the dataset, the value of $GA(n_{ys})$ could be very large, which is indeed the case for aerial datasets. A large value causes numerical instability, which is the drawback of [5]. We mitigate this issue by using an n^{th} root as follows:

$$\tilde{E_{ys}} = \frac{1 - \beta^{\sqrt[n]{GA(n_{ys})}}}{1 - \beta} \tag{7}$$

Using n^{th} root stabilizes the effective numbers without changing the way they ($\tilde{E_{ys}}$) are calculated.

In our overall object detection framework, for an object instance belonging to a class y and size cluster s, our loss function $\mathcal{L}_{ours}$ is thus given by:

$$\mathcal{L}_{ours} = \mathcal{L}_C + w_{ys} * \mathcal{L}_R \tag{8}$$

where $\mathcal{L}_C$ and $\mathcal{L}_R$ represent the classification and regression loss terms respectively, and w_{ys} is the re-weighting factor based on $\tilde{E_{ys}}$ as given below:

$$w_{ys} = 1 - \frac{1}{\tilde{E_{ys}}} \tag{9}$$

Adding the above weights w_{ys} to the object detection loss is the only implementation step required in our framework for any object detection architecture, thus making our approach easy to implement and effective.

4. Experiments and Results

Datasets: We perform extensive experiments on several popular drone-based aerial image datasets, namely, DOTA-v1.0 [34], DOTA-v1.5 [1], HRSC2016 [19] and VisDrone [6]. The details of these datasets are shared below.

DOTA-v1.0 [34]: This is one of the largest aerial image datasets released in 2018 containing 2,806 images and 188,282 object instances. The dataset is divided into train, val and test in $1/2$, $1/6$ and $1/3$ ratios respectively. The

aerial images are widespread in 15 different categories namely Plane (PL), Baseball-Diamond (BD), Bridge (BR), Ground-Track-Field (GTF), Small-vehicle (SV), Large-vehicle (LV), Ship (SH), Tennis-Court (TC), Baseball-Court (BC), Storage-Tank (ST), Soccer-Ball-Field (SBF), Roundabout (RA), Harbor (HA), Swimming-Pool (SP) and Helicopter (HC). Small-vehicle class is the majority class while Ground-Track-Field (GTF) is the minority class.

DOTA-v1.5: [1]: This was released in 2019 as the succeeding version of the DOTA-v1.0 with an additional category Container-Crane (CC) added to it. Although it is made from the same images as DOTA-v1.0, many additional annotations of very small object instances (less than 10 pixels) are added. It has a total of 403,318 object instances which is more than double the instances present in DOTA-v1.0, making it very distinct. Object detection on DOTA-v1.5 is more challenging than on v1.0 because of the newly added very small instances. Small-vehicle class is the majority class while the newly added Container-Crane class is the minority class.

HRSC2016 [19]: This is an aerial image dataset that focuses on ship detection. It is comparatively smaller in number, but has variation in object sizes. It has 1061 images divided into 436, 181 and 444 images for training, validation and testing respectively.

VisDrone [6]: This dataset is released as a part of the VisDrone Object detection challenge in 2019. It contains a total of 10209 images divided into 6471, 548 and 3190 for training, validation and testing respectively. Train and validation sets have a combined total of nearly 382,000 object instances that are spread across 10 different categories.

Evaluation Metrics: For HRSC2016 and VisDrone datasets, we present the results in the standard COCO format, mAP as the mean of APs@$[.5 : .05 : .95]$. For DOTA-v1.0 and DOTA-v1.5, following [33, 35, 7, 8], we present class wise AP@50 and mAP as the mean of class-wise APs.

Implementation Details: Our method is architecture-agnostic and can be applied on top of any architecture proposed for object detection. As we aim to solve the size imbalance issues in drone-based aerial object datasets, for our experiments, we chose two recent state-of-the-art aerial object detection architectures [7, 8] as our baselines. We implement our method using the `mmdetection` repository. For purposes of fair comparison, we use the same backbone, training schedules, optimizer, learning rate, momentum, weight decay, number of epochs and dataset preparation strategy as used in the baseline methods [7, 8]. For training, we use 4 GTX 1080 Ti GPUs and for inference, we use a single GTX 1080 Ti GPU.

Results:

HRSC2016. For our experiments, we use ReResNet50 as the backbone and ReFPN as the neck which were proposed in [8]. We crop all images in HRSC2016 dataset to $800*512$ and perform horizontal flip augmentation.Table 2 shows our results. Our method obtains a notable performance improvement of **2.01%** mAP over the baseline method [8].

Method	mAP
ReDet [8]	70.41
Ours + ReDet [8]	72.42

Table 2: Comparisons with the baseline on HRSC2016.

DOTAv1.0. For both DOTA-v1.0 and v1.5, the images were cropped to $1024*1024$ and augmented with horizontal flips. Table 3 summarizes the results of state-of-the-art methods on DOTA-v1.0 OBB task. We apply our method on top of two baselines S^2aNet [7] and ReDet [8]. As observed, our method obtains improvement on top of both baselines, showing the architecture-agnostic nature of our approach. ReDet obtains a performance of 76.15% mAP and our method obtains 77.14% mAP. Our model performs better than all existing state-of-the-art methods. Compared to ReDet, our method improves performance on 12 out of 15 classes, which contain a good mix of both small and large-sized objects. Specifically, on classes 'Roundabout (RA)' and 'Helicopter (HC)', our method achieves an improvement of **4.49%** and **3.76%** in AP respectively. We observed that these classes have severe size imbalance which shows the efficacy of our approach.

DOTAv1.5. Table 4 provides a comparison with state-of-the-art results on DOTA-v1.5 OBB task. ReDet [8] obtains a performance of 66.86% mAP, while our method obtains 68.71% mAP. Our method achieves a gain of **1.85%** mAP. We also obtain improvement for most of the classes on this dataset. Specifically, for the class 'Basketball Court', which has severe imbalance in object sizes, we obtain an improvement of **5.97%** in AP. Despite the fact that DOTA-v1.5 contains a lot of newly added small instances when compared to DOTA-v1.0, our methods achieves better results on DOTA-v1.5, which supports our claim that our method improves performance on small objects.

VisDrone. We use the same image cropping and augmentation techniques as used for the DOTA datasets. A performance comparison between the state-of-the-art models and our model is given in Table 5. As the evaluation server for this challenge dataset is closed, we present our model's performance on the validation set, and do the same for the baseline models for fairness of comparison. Our model achieves a performance gain of **1.5%** mAP over the baseline.

Results on small, medium and large objects. Table 6 shows a comparison in the performance of the baselines [7, 8] and our model on different sized objects. For these experiments, we use the test set of HRSC2016 and the validation set of DOTA-v1.0 and DOTA-v1.5. Note that the ground truth

Method	backbone	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	mAP
DRN [21]	H-104	88.91	80.22	43.52	63.35	73.48	70.69	84.94	90.14	83.85	84.11	50.12	58.41	67.62	68.60	52.50	70.70
CenterMap [33]	R50-FPN	88.88	81.24	53.15	60.65	78.62	66.55	78.10	88.83	77.80	83.61	49.36	66.19	72.10	**72.36**	58.70	71.74
R^3Det [35]	R50-FPN	88.92	77.70	46.49	71.24	72.70	77.81	79.75	90.86	81.46	83.96	57.53	59.10	65.24	70.59	51.38	71.63
S^2aNet [7]	R50-FPN	89.00	80.77	51.77	70.91	78.52	78.01	87.19	90.86	84.99	84.64	58.45	63.60	66.39	67.90	57.92	74.06
Ours + S^2aNet [7]	R50-FPN	89.23	81.07	51.92	70.91	**78.68**	78.97	87.33	90.89	86.07	**85.41**	63.20	66.22	66.90	69.82	59.81	75.20
ReDet [8]	ReR50-ReFPN	89.34	83.03	53.83	74.35	77.45	83.41	87.86	90.87	87.77	85.06	62.89	62.10	75.76	70.58	57.93	76.15
Ours + ReDet [8]	ReR50-ReFPN	**89.34**	**83.17**	**54.16**	**76.24**	78.22	**83.42**	**87.97**	**90.90**	**87.86**	85.35	**65.39**	**66.59**	**76.17**	70.63	**61.69**	**77.14**

Table 3: Comparison of our method with the state-of-the-art methods on DOTA-v1.0 OBB Task. The results in bold specify the best result of each column.

Method	PL	BD	BR	GTF	SV	LV	SH	TC	BC	ST	SBF	RA	HA	SP	HC	CC	mAP
RetinaNet-O [14]	71.43	77.64	42.12	64.65	44.53	56.79	73.31	90.84	76.02	59.96	46.95	69.24	59.65	64.52	48.06	0.83	59.16
FR-O [26]	71.89	74.47	44.45	59.87	51.28	68.98	79.37	90.78	77.38	67.50	47.75	69.72	61.22	65.28	60.47	1.54	62.00
Mask R-CNN [9]	76.84	73.51	49.90	57.80	51.31	71.34	79.75	90.46	74.21	66.07	46.21	70.61	63.07	64.46	57.81	9.42	62.67
HTC [4]	77.80	73.67	51.40	63.99	51.54	73.31	80.31	90.48	75.12	67.34	48.51	70.63	64.84	64.48	55.87	5.15	63.40
ReDet [8]	79.20	82.81	51.92	71.41	**52.38**	75.73	80.92	90.83	75.81	68.64	49.29	72.03	73.36	**70.55**	63.33	11.53	66.86
Ours + ReDet [8]	**79.85**	**83.02**	**52.86**	**72.73**	52.35	**75.74**	**87.18**	**90.87**	**81.78**	**68.68**	**56.90**	**73.16**	**73.41**	70.49	**65.96**	**14.34**	**68.71**

Table 4: Comparison of our method with the state-of-the-art methods on DOTA-v1.5 test set OBB Task.

Method	Backbone	AP@50	AP@75	mAP
RetinaNet [14]	R50	27.7	12.7	13.9
DSHNet [38]	R50	30.2	15.5	16.1
ReDet [8]	ReR50-ReFPN	30.86	19.50	18.80
Ours + ReDet [8]	ReR50-ReFPN	**32.84**	**21.6**	**20.32**

Table 5: Comparison of our method with the state-of-the-art methods on VisDrone validation set.

Trained on	Method	Tested on		
		Small	Medium	Large
HRSC2016	ReDet	17.93	29.58	38.91
	Ours + ReDet	**20.79**	**29.97**	38.01
DOTA-v1.0	ReDet	09.74	23.48	52.44
	Ours + ReDet	**11.81**	23.34	52.24
DOTA-v1.5	ReDet	8.32	24.85	43.56
	Ours + ReDet	**10.65**	24.76	43.52
DOTA-v1.0	S^2aNet	10.64	24.93	47.43
	Ours + S^2aNet	**12.48**	**25.57**	**47.85**

Table 6: Comparison between the performance of our model and the baseline model on small, medium and large sized objects.

annotations of the test set for DOTA dataset are not publicly available, hence, we use the validation set. We follow the same evaluation metrics as mentioned in Section 4. Our method when applied on top of ReDet [8], achieves an improvement of **2.86%**, **2.07%** and **2.33%** mAP on the small sized objects of HRSC2016, DOTA-v1.0 and DOTA-v1.5 datasets respectively (first three rows of Table 6). We also provide the performance gain of our method applied on top of a different architecture [7] (last row of Table 6). On all the datasets, our model maintains the performance on medium and large sized objects as well which shows the efficacy of our approach.

Method	FPN	AP@50	AP@75	mAP
S^2aNet [7]	✗	56.27	23.72	27.85
Ours + S^2aNet [7]	✗	57.68	25.22	28.78
S^2aNet [7]	✓	74.06	36.88	40.28
Ours + S^2aNet [7]	✓	75.20	38.76	41.04

Table 7: Performance of our model with and without FPN.

5. Discussion and Analysis

5.1. Ablation studies

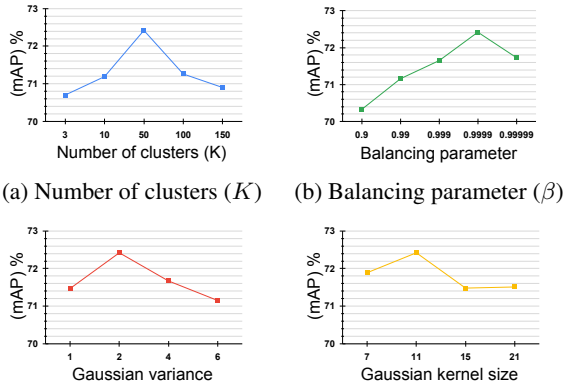
To clearly evaluate the effectiveness of our proposed approach, we perform ablation studies on the DOTA-v1.0 dataset using two baselines S^2aNet [7] and ReDet [8]. We use the ResNet50-FPN and ReResNet50-ReFPN backbones for experiments on the baseline methods [7] and [8] respectively. Table 8 shows the results, and indicates consistent improvement over the baseline methods.

Method	GA	AP@50	AP@75	mAP
S^2aNet [7]	✗	74.06	36.88	40.28
Ours + S^2aNet [7]	✗	74.32	37.12	40.35
Ours + S^2aNet [7]	✓	**75.20**	**38.76**	**41.04**
ReDet [8]	✗	76.15	50.75	47.05
Ours + ReDet [8]	✗	76.47	51.15	47.42
Ours + ReDet [8]	✓	**77.14**	**52.93**	**48.13**

Table 8: Performance comparison of our model with and without Gaussian Amplification. GA refers to Gaussian Amplification.

Effect of Gaussian Amplification: Table 8 shows the performance of our model with and without employing the Gaussian amplification step. As observed, when Gaussian amplification is not applied, results show minimal improvement on both baselines [7, 8] in contrast to a healthy improvement when it is applied. This shows the importance of

considering the effect of neighborhood when dealing with ordinal variables like size of objects.



(a) Number of clusters (K) (b) Balancing parameter (β)

(c) Gaussian variance σ (d) Gaussian width/kernel size

Figure 5: Effect of various hyperparameters on model performance

5.2. Hyperparameter Sensitivity Analysis

Analysis on k: k is the hyperparameter associated with k-Means clustering. It determines the number of clusters. We experimented with different k values on HRSC2016 and $k = 50$ gives the best results as evident from Figure 5a. We noticed this to be consistent with other datasets too.

Analysis on β: β is the hyper-parameter used in the calculation of effective weights. To determine the best value of β, we vary β in the range $[0.9, 0.99, 0.999, 0.9999, 0.99999]$. Figure 5b shows the performance on HRSC2016 dataset. We observed that by increasing the value of β, starting from 0.9, the performance increases until $\beta = 0.9999$ which achieves the best result. Lower values of β, eg. 0.5, 0.6, and 0.7 make the effective weights uniform. This issues is also cited in [5]. Upon experimentation on smaller βs, we observed that the results were close to the baseline.

Analysis on σ: σ specifies the level of importance to be given to neighboring bins when applying Gaussian amplification to a given bin. It can also be regarded as an amplification factor. Figure 5c shows the performance of our model as we change the σ value. We obtained the best results when σ is set to 2.

Analysis on w: This hyperparameter specifies the number of neighboring bins to consider while applying Gaussian amplification to a given bin. We experimented on HRSC2016 dataset by varying w, refer Figure 5d. We found that a width of 11 gives the best results.

We note that these hyperparameter values performed well across all the four datasets considered, without the need to fine tune the model separately on different datasets.

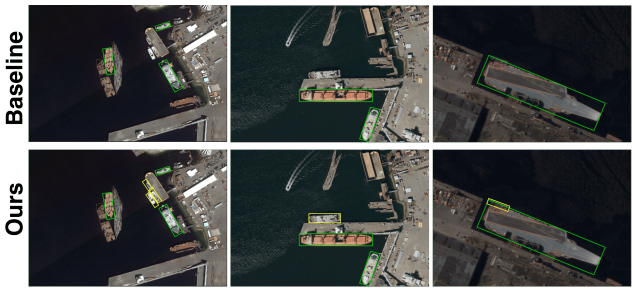


Figure 6: Predictions on images from HRSC2016 dataset [19] - ReDet [8] vs Ours. **Top:** The baseline method fails to detect small sized objects. **Bottom:** Ours is able to recognize additional small objects. Yellow boxes indicate objects additionally detected.

5.3. Further Analysis

Feature Pyramid Network [13] is one of the primary methods developed to solve the problem of high variation in sizes of instances in object detection datasets. Many methods [23, 12] have been proposed based on the idea of FPN. We perform experiments on DOTA_v1.0 dataset to show that our approach improves detection performance when applied on top of FPN. Table 7 summarizes the results of these experiments. As observed, when FPN is not applied, our model improves mAP from 27.85 to 28.78. Also, when FPN is used, our model achieves the best performance by improving mAP from 40.28 to **41.04**. FPN enhances the model's capability to process at multiple scales while our method addresses the severe long-tail in the size imbalance. Hence, best results are obtained when our method is applied on top of such architecture-engineered methods.

6. Conclusions and Future Work

In this work, we presented a framework to alleviate imbalance in the object size distribution. We proposed a novel simple-to-implement architecture-agnostic loss reweighting method for drone-based aerial object detection. We dealt with the ordinality of size by taking into consideration the effect of neighborhood instances on prediction and by clustering the object instances based on their size. We showed the need to increase the contribution of small objects despite them belonging to the head of the long-tail size distribution. We showed that our method improves performance on popular datasets like HRSC2016, DOTAv1.0 DOTAv1.5 and VisDrone. In future, we plan to extend this work by mitigating the imbalance of class and size together.

Acknowledgements. We are grateful to the Ministry of Electronics and Information Technology and Ministry of Education, Govt of India, as well as IIT-Hyderabad through its MoE-DRDO fellowship program for their support of this project. We also thank the anonymous reviewers and Area Chairs for their valuable feedback in improving the presentation of this paper.

References

[1] Dota1.5 dataset: Object detection in aerial images. https://captain-whu.github.io/DOAI2019/dataset.html. 5, 6

[2] Y. Cao, K. Chen, Chen Change Loy, and D. Lin. Prime sample attention in object detection. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11580–11588, 2020. 2, 3

[3] Dario Cazzato, Claudio Cimarelli, Jose Luis Sanchez-Lopez, Holger Voos, and Marco Leo. A survey of computer vision methods for 2d object detection from unmanned aerial vehicles. *Journal of Imaging*, 6(8):78, 2020. 1

[4] Kai Chen, Jiangmiao Pang, Jiaqi Wang, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jianping Shi, Wanli Ouyang, et al. Hybrid task cascade for instance segmentation. pages 4974–4983, 2019. 7

[5] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge J. Belongie. Class-balanced loss based on effective number of samples. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9260–9269, 2019. 2, 3, 5, 8

[6] Dawei Du, Pengfei Zhu, Longyin Wen, Xiao Bian, Haibin Lin, Qinghua Hu, Tao Peng, Jiayu Zheng, Xinyao Wang, Yue Zhang, Liefeng Bo, Hailin Shi, Rui Zhu, Aashish Kumar, Aijin Li, Almaz Zinollayev, Anuar Askergaliyev, Arne Schumann, Binjie Mao, Byeongwon Lee, Chang Liu, Changrui Chen, Chunhong Pan, Chunlei Huo, Da Yu, DeChun Cong, Dening Zeng, Dheeraj Reddy Pailla, Di Li, Dong Wang, Donghyeon Cho, Dongyu Zhang, Furui Bai, George Jose, Guangyu Gao, Guizhong Liu, Haitao Xiong, Hao Qi, Haoran Wang, Heqian Qiu, HongLiang Li, Huchuan Lu, Ildoo Kim, Jaekyum Kim, Jane Shen, Jihoon Lee, Jing Ge, Jingjing Xu, Jingkai Zhou, Jonas Meier, Jun Won Choi, Junhao Hu, Junyi Zhang, Junying Huang, Kaiqi Huang, Keyang Wang, Lars Sommer, Lei Jin, and Lei. Zhang. Visdrone-det2019: The vision meets drone object detection in image challenge results. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) Workshops*, Oct 2019. 1, 5, 6

[7] J. Han, J. Ding, J. Li, and G. S. Xia. Align deep features for oriented object detection. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–11, 2021. 3, 6, 7

[8] Jiaming Han, Jian Ding, Nan Xue, and Gui-Song Xia. Redet: A rotation-equivariant detector for aerial object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2786–2795, 2021. 1, 3, 4, 6, 7, 8

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 7

[10] Buyu Li, Y. Liu, and X. Wang. Gradient harmonized single-stage detector. *ArXiv*, abs/1811.05181, 2019. 2, 3

[11] Jianan Li, Xiaodan Liang, ShengMei Shen, Tingfa Xu, Jiashi Feng, and Shuicheng Yan. Scale-aware fast r-cnn for pedestrian detection. *IEEE transactions on Multimedia*, 20(4):985–996, 2017. 3

[12] Yanghao Li, Yuntao Chen, Naiyan Wang, and Zhaoxiang Zhang. Scale-aware trident networks for object detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 6054–6063, 2019. 3, 8

[13] Tsung-Yi Lin, P. Dollár, Ross B. Girshick, Kaiming He, Bharath Hariharan, and Serge J. Belongie. Feature pyramid networks for object detection. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 936–944, 2017. 2, 3, 8

[14] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. pages 2980–2988, 2017. 1, 3, 7

[15] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and P. Dollár. Focal loss for dense object detection. *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2999–3007, 2017. 1, 2

[16] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014. 1

[17] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *Int. J. Comput. Vision*, 128(2):261–318, Feb 2020. 3

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016. 2, 3

[19] Zikun Liu, Liu Yuan, Lubin Weng, and Yiping Yang. A high resolution optical satellite image dataset for ship recognition and some new baselines. In *International conference on pattern recognition applications and methods*, volume 2, pages 324–331. SCITEPRESS, 2017. 1, 5, 6, 8

[20] Kemal Oksuz, Baris Can Cam, Sinan Kalkan, and Emre Akbas. Imbalance problems in object detection: A review. *IEEE transactions on pattern analysis and machine intelligence*, 43(10):3388–3415, 2020. 2, 3

[21] Xingjia Pan, Yuqiang Ren, Kekai Sheng, Weiming Dong, Haolei Yuan, Xiaowei Guo, Chongyang Ma, and Changsheng Xu. Dynamic refinement network for oriented and densely packed object detection. June 2020. 7

[22] Jiangmiao Pang, K. Chen, J. Shi, H. Feng, Wanli Ouyang, and D. Lin. Libra r-cnn: Towards balanced learning for object detection. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 821–830, 2019. 3

[23] Yanwei Pang, Tiancai Wang, Rao Muhammad Anwer, Fahad Shahbaz Khan, and Ling Shao. Efficient featurized image pyramid network for single shot detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7336–7344, 2019. 3, 8

[24] Qi Qian, L. Chen, H. Li, and Rong Jin. Dr loss: Improving object detection by distributional ranking. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12161–12169, 2020. 2, 3

[25] Joseph Redmon, S. Divvala, Ross B. Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016. 1, 2

[26] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. pages 1137–1149, 2017. 7

[27] Shaoqing Ren, Kaiming He, Ross B. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39:1137–1149, 2015. 1, 2

[28] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3578–3587, 2018. 2, 3

[29] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. *Advances in neural information processing systems*, 31, 2018. 3

[30] Jingru Tan, Changbao Wang, Buyu Li, Quanquan Li, Wanli Ouyang, C. Yin, and J. Yan. Equalization loss for long-tailed object recognition. *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11659–11668, 2020. 2, 3

[31] Shashank Tripathi, S. Chandra, Amit Agrawal, Ambrish Tyagi, James M. Rehg, and V. Chari. Learning to generate synthetic data via compositing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 461–470, 2019. 2, 3

[32] Hao Wang, Qilong Wang, F. Yang, Weiqi Zhang, and W. Zuo. Data augmentation for object detection via progressive and selective instance-switching. *ArXiv*, abs/1906.00358, 2019. 2, 3

[33] Jinwang Wang, Wen Yang, Heng-Chao Li, Haijian Zhang, and Gui-Song Xia. Learning center probability map for detecting objects in aerial images. *IEEE Transactions on Geoscience and Remote Sensing*, 2020. 6, 7

[34] Gui-Song Xia, X. Bai, J. Ding, Zhen Zhu, Serge J. Belongie, Jiebo Luo, M. Datcu, M. Pelillo, and L. Zhang. Dota: A large-scale dataset for object detection in aerial images. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3974–3983, 2018. 1, 5

[35] Xue Yang, Qingqing Liu, Junchi Yan, Ang Li, Zhiqiang Zhang, and Gang Yu. R3det: Refined single-stage detector with feature refinement for rotating object. *arXiv preprint arXiv:1908.05612*, 2019. 3, 6, 7

[36] Yuzhe Yang, Kaiwen Zha, Ying-Cong Chen, Hao Wang, and Dina Katabi. Delving into deep imbalanced regression. *arXiv preprint arXiv:2102.09554*, 2021. 3, 4

[37] Jingru Yi, Pengxiang Wu, Bo Liu, Qiaoying Huang, Hui Qu, and Dimitris Metaxas. Oriented object detection in aerial images with box boundary-aware vectors. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2150–2159, 2021. 1

[38] Weiping Yu, Taojiannan Yang, and Chen Chen. Towards resolving the challenge of long-tail distribution in uav images for object detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, pages 3258–3267, January 2021. 7