UFPMP-Det: Toward Accurate and Efficient Object Detection on Drone Imagery

Yecheng Huang¹,², Jiaxin Chen², Di Huang¹,²*

¹ State Key Laboratory of Software Development Environment, Beihang University, Beijing, China
² School of Computer Science and Engineering, Beihang University, Beijing, China
{ychuang, jiaxinchen, dhuang}@buaa.edu.cn

Abstract
This paper proposes a novel approach to object detection on drone imagery, namely Multi-Proxy Detection Network with Unified Foreground Packing (UFPMP-Det). To deal with the numerous instances of very small scales, different from the common solution that divides the high-resolution input image into quite a number of chips with low foreground ratios to perform detection on them each, the Unified Foreground Packing (UFP) module is designed, where the sub-regions given by a coarse detector are initially merged through clustering to suppress background and the resulting ones are subsequently packed into a mosaic for a single inference, thus significantly reducing overall time cost. Furthermore, to address the more serious confusion between inter-class similarities and intra-class variations of instances, which deteriorates detection performance but is rarely discussed, the Multi-Proxy Detection Network (MP-Det) is presented to model object distributions in a fine-grained manner by employing multiple proxy learning, and the proxies are enforced to be diverse by minimizing a Bag-of-Instance-Words (BoIW) guided optimal transport loss. By such means, UFPMP-Det largely promotes both the detection accuracy and efficiency. Extensive experiments are carried out on the widely used VisDrone and UAVDT datasets, and UFPMP-Det reports new state-of-the-art scores at a much higher speed, highlighting its advantages. The code is available at https://github.com/PuAnysh/UFPMP-Det.

Introduction
Recently, the drone, also known as UAV, has become a popular equipment for its trade-off between mobility and stability in many applications, such as security surveillance, aerial photography, express delivery, and agricultural production, where drone image based object detection is the fundamental issue. Although object detection on natural images has been greatly developed by Convolutional Neural Networks (CNNs) during the past several years, that on drone imagery is still limited in terms of both accuracy and efficiency.

One major challenge to detect objects using drone images lies in that there exist a large amount of instances of very small sizes, and compared with the case in the benchmarks of general object detection, e.g., PASCAL VOC (Everingham et al. 2010) and MS COCO (Lin et al. 2014), the ratio of such instances is rather high, as Fig.1 (a) depicts. Meanwhile, computing resources are usually inadequate in drones, current time-consuming solutions, like image pyramid, which proves effective in general object detection, are no longer competent. Instead, a coarse-to-fine pipeline is often followed (Li et al. 2020a; Yang et al. 2019), where a coarse detector is launched to locate the large-scale instances and sub-regions that contain densely distributed small ones and a fine detector is further applied to those regions to find instances of small sizes. These methods show promising results; however the sub-regions delivered by the coarse detectors are relatively rough, with a large portion of backgrounds included, incurring inefficient computations. Furthermore, since they partition the input image into multiple chips, they have to individually process each sub-region, leading to several times of inferences for final decision. The two drawbacks hence suggest room for efficiency improvement.

Another considerable challenge is that some object categories defined in drone datasets, e.g. VisDrone (Zhu et al. 2018), are semantically close to each other, for instance,
pedestrian vs. person; tricycle vs. awning-tricycle, and the appearances of instances belonging to these categories are quite confusing, as displayed in Fig. 1 (b). Besides, due to more severe disturbances caused by flying altitude, viewing angle, and weather condition, the distances between features of the instances from the same category tend to be enlarged. The inter-class similarities and intra-class variations are thus more seriously intertwined than in general object detection, making the classification of instances even harder. Unfortunately, to the best of our knowledge, this problem is ignored in the previous literature, which leaves much space for accuracy amelioration.

To address the two challenges aforementioned, i.e., high percentage of small instances and low distinctiveness of similar categories, in this paper, we present a novel approach to object detection on drone images, namely Multi-Proxy Detection Network with Unified Foreground Packing (UFPMPDet). It substantially extends the coarse-to-fine framework by two specially designed modules. For the former, the Unified Foreground Packing (UFP) module is proposed. UFP operates in a two-stage manner, where the foreground sub-regions by the coarse detector are firstly merged through a clustering algorithm to suppress backgrounds and the resulting regions are then packed into a mosaic with adaptively enlarged scales. By this mean, the foreground ratio of small objects is increased, and the successive fine detector performs inference only once at the mean time. As a consequence, both the detection accuracy and speed can be promoted. For the latter, the Multi-Proxy Detection Network (MP-Det) module is proposed. In MP-Det, the multi-proxy learning scheme originally explored for the image retrieval task is adapted to object detection, aiming to boost the performance of the classification head by generating compound and flexible decision boundaries. In particular, to bypass the collapse phenomenon in training multiple proxies, Bag-of-Instance-Words (BoIW) guided Optimal Transport is introduced, in which BoIW well models the distribution of each category in the presence of the confusion between inter-class similarities and intra-class variations, thus facilitating feature-proxy matching by Sinkhorn optimization. We extensively evaluate the proposed approach on two public databases, i.e., VisDrone and UAVDT, and report the state-of-art scores with largely promoted efficiency, highlighting its effectiveness.

**Related Work**

**Generic Object Detection**

Generic Object Detection has been largely developed in recent years along with the success of CNNs within the community of artificial intelligence, especially computer vision. On whether using pre-defined sliding windows or proposals, the existing methods are divided into two main streams, i.e., anchor-based and anchor-free. The anchor-based detectors sample the box space into discrete bins and refine the boxes of objects accordingly, and anchors are taken as regression references and classification candidates to infer proposals in multi-stage detectors, such as R-CNN (Girshick et al. 2014), Faster-RCNN (Ren et al. 2017) and Cascade-RCNN (Cai and Vasconcelos 2018), or final bounding boxes in single-stage ones, e.g. SSD (Liu et al. 2016), YOLO (Redmon and Farhadi 2018), and RetinaNet (Lin et al. 2020). Compared to anchor-based ones, the anchor-free detectors avoid complicated computation related to anchor boxes and bypass the corresponding prior hyper-parameter setting, figuring out a promising alternative, where FCOS (Tian et al. 2019), FSAF (Zhu, He, and Savvides 2019), and GFL v1/v2 (Li et al. 2020b, 2021) are representatives.

**Object Detection on Drone Imagery**

Despite the progress achieved by object detection on natural images, e.g. PASCAL VOC and COCO, the performance on drone images is still far from satisfactory. As stated, the high percentage of small instances and low distinctiveness of similar categories make the issue more challenging. Inspired by the region search strategies (Singh, Najibi, and Davis 2018; Najibi, Singh, and Davis 2019) employed in general object detection to accelerate training and inference, all the current studies focus on the problem of small instances and address it by following a coarse-to-fine framework, which serially adopts a simple strategy or a coarse detector to roughly suggest the regions with densely distributed small instances and a fine detector to precisely localize the objects on them. For example, in (Unel, Özkalayci, and Çıgla 2019), the tiling based method makes even splits to produce sub-regions but it tends to break instances while truncating images; ClutDet (Yang et al. 2019) applies a sub-network to crop sub-regions from the raw input; and DMNet (Li et al. 2020a) estimates the object density in the original image and then separates sub-regions as minimal areas of connected possible blocks.

Such methods indeed advance drone image based object detection; however, the sub-region generated are not so decent with much background in them and the decision on an entire image requires multiple inferences on its sub-regions, both of which show room for efficiency. On the other side, the low distinctiveness of similar categories has not been discussed, resulting in limitation to improved accuracies.

**Small Object Detection**

Since MS COCO (Lin et al. 2014) was released, small object detection has become a critical topic and received increasing attention. FPN (Lin et al. 2017) is a major choice to handle scale changes through feature pyramid, and it is extended to a number of variants, including EFF-FPN (Gong et al. 2021), AugFPN (Guo et al. 2020), etc. Perceptual GAN (Li et al. 2017) utilizes an adversarial network to boost the detection performance by narrowing the representation difference between small and large objects, and a super-resolution feature generator is trained with proper high-resolution target features for supervision. A similar idea appears in (Noh et al. 2019) but considers the impacts of the receptive fields of various sizes as well. TinyPerson (Yu et al. 2020) claims that scale mismatch between the data for network pre-training and detector learning incurs degradation and designs Scale Match to align object scales between different datasets.

The methods above are validated either on MS COCO for general object detection or on other detection tasks for special objects, i.e. persons or traffic signs. As we analyze before, drone image based object detection has its unique chal-
The Proposed UFPMP-Det Approach
To tackle the two challenges, i.e., high percentage of small instances and low distinctiveness of similar categories, a novel approach, namely Multi-Proxy Detection Network with Unified Foreground Packing (UFPMP-Det), is proposed. It contains two major stages, where in the first stage, the Unified Foreground Packing (UFP) module converts raw drone images into mosaics with higher foreground ratios, and in the second stage, the Multi-Proxy Detection Network (MP-Det) module infers on the mosaic images, which employs a multi-proxy learning scheme with Bag-of-Instance-Words (BoIW) guided Optimal Transport to model complex object distributions. They are introduced in detail in the following.

Unified Foreground Packing
The UFP module aims to convert original drone images into unified mosaic ones with significantly increased foreground ratios and enlarged sizes of small objects. As Fig. 3 shows, when foreground sub-areas are extracted from the drone image by a coarse detector, UFP introduces three successive steps: (1) foreground sub-areas are merged to several clustered ones; (2) small scale cluster regions are enlarged adaptively; and (3) adjusted cluster regions are packed into a unified mosaic.

Foreground Region Generation In order to mitigate severe biases and heavy overlaps of foreground sub-areas in coarse detection, we expand the width and height of each detected bounding box from the center with an expansion ratio β to roughly enclose its ground truth. Thereafter, we propose a greedy Foreground Region Generation (FRG) algorithm to merge the expanded results as summarized in Algorithm 1.

Specifically, FRG takes the expanded coarse detection results \(B_c\) as input and selects the box \(A\) with the minimal size as the generation starting point. For each box \(B\) in \(B_c - \{A\}\), FRG searches the smallest convex box \(C\) that encloses \(A\) and \(B\). If the sum of the areas \(|A| + |B|\) is larger than that of \(C\), we update \(A\) with \(C\) and remove \(B\) from \(B_c\). This process is repeated until there is no box \(B\) that satisfies the condition \(|A| + |B| \geq |C|\). In this case, \(A\) is collected as a cluster region in \(B_c\). We repeat the procedure above until \(B_c\) turns to an empty set, when we obtain the final merged region set \(B_r\).

Algorithm 1: Foreground Region Generation

<table>
<thead>
<tr>
<th>Input</th>
<th>Bounding boxes (B_c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>Merged regions (B_r)</td>
</tr>
</tbody>
</table>

1: Initialize \(B_r = \emptyset\).
2: while \(B_c \neq \emptyset\) do
3: \(A = \arg \min_{A' \in B_c} |A'|\)
4: \(B_c := B_c - \{A\}\)
5: for all \(B \in B_c\) do
6: For \(A\) and \(B\), find the smallest enclosing convex bounding box \(C\).
7: if \(|A| + |B| \geq |C|\) then
8: \(A = C\)
9: \(B_c := B_c - \{B\}\)
10: \end if
11: \end for
12: \(B_r := B_r \cup \{A\}\)
13: \end while
14: return \(B_r\)

Figure 3: Pipeline of Unified Foreground Packing (UFP).
we first estimate the averaged region scale from $B$, and subsequently enlarge the regions smaller than a fixed size (e.g. $96 \times 96$ in our work) by adjusting the average scale to the fixed one.

**Foreground Region Packing**  The current approaches individually perform fine-grained detection on each cluster region, which is extremely inefficient. To avoid this, we splice all the regions into a unified mosaic by using the PHSPPOG method (Zhang et al. 2016). By this mean, fine-grained detection operates only once, remarkably saving the time cost.

**Multi-Proxy Detection Network**
The MP-Det module targets on relieving the confusion between inter-class similarities and intra-class variations from complex object distributions in classification. It includes two main components: (1) the Multi-Proxy Classification Head and (2) the Bag-of-Instance-Words (BoIW) model.

**Multi-Proxy Classification Head**  The conventional classification head assigns a single weight vector $w_i$ to the $i$-th category. Suppose that $x$ is the feature extracted by the backbone network, its conditional probability corresponding to the $i$-th category is formulated as:

$$ P_r(Y = y_i | x) = \text{Sigmoid}(w_i^T x), \quad (1) $$

where $\text{Sigmoid}(\cdot)$ is the sigmoid function. Eq. (1) can be considered as a single proxy classifier, implicitly assuming that each category has only one class center at $w_i / \|w_i\|$. By contrast, in drone images, intra-class variance is rather large due to great size and view changes of instances. In this case, data belonging to a single class may span around multiple centers. Therefore, we adopt the multi-proxy classification head, assuming that the $i$-th category has $K$ ($K > 1$) proxies denoted by $[w_i^1, w_i^2, ..., w_i^K]$. Inspired by SoftTriple (Qian et al. 2019), the multi-proxy conditional probability is formulated as:

$$ P_r(Y = y_i | x) = \text{Sigmoid} \left( \gamma \sum_k \frac{\exp(s_k^i) s_k^i}{\sum_{l=1}^K \exp(s_l^i)} \right), \quad (2) $$

where $s_k^i = w_i^k x / \|w_i^k\| / \|x\|$ and $\gamma$ is a scaling factor. Accordingly, the weights $[w_i^1, w_i^2, ..., w_i^K]$ can be optimized by minimizing the cross-entropy or the focal loss w.r.t. $P_r(Y = y_i | x)$.

As in Eq. (2), the decision boundary for the $i$-th class is represented by multiple centers (proxies), thus being more flexible and accurate than that by a single one.

**Bag-of-Instance-Words guided Optimal Transport**  The multiple proxies tend to collapse into a similar one during training, caused by extremely imbalanced cluster assignments. To overcome this dilemma, an optimal transport based procedure is adopted as in (Liu et al. 2021a). Specifically, given the feature vectors $\{f_i^j\}_{j=1}^{N_i}$ of $N_i$ positive instances and $K$ proxies $\{w_i^k\}_{k=1}^K$ for the $i$-th class, the cost matrix is computed as $C_i = \frac{1}{2} S_i$, where $S_i \in \mathbb{R}^{N_i \times K}$ refers to the cosine similarity matrix between $\{f_i^j\}_{j=1}^{N_i}$ and $\{w_i^k\}_{k=1}^K$ and $1$ is an all-ones matrix. The transportation plan $P_i^* \in \mathbb{R}^{N_i \times K}$ is computed as below:

$$ P_i^* = \arg \min_{P} \; \text{tr} \left( C_i^T P \right) \quad \text{s.t.} \quad \sum_{j=1}^{N_i} P(j, k) = p_i(k), \sum_{k=1}^K P(j, k) = q_i(j), \quad (3) $$

where $\text{tr}(\cdot)$ is the matrix trace. $p_i = [p_i(1), \cdots, p_i(K)]$ and $q_i = [q_i(1), \cdots, q_i(N_i)]$ indicate the marginal probability distributions of $\{w_i^k\}_{k=1}^K$ and $\{f_i^j\}_{j=1}^{N_i}$, respectively. Based on Eq. (3), the instance-proxy matching loss w.r.t. $\{f_i^j\}_{j=1}^{N_i}$ and $\{w_i^k\}_{k=1}^K$ is defined as

$$ \mathcal{L}_{ot} = \frac{1}{N_c} \sum_{i=1}^{N_c} \text{tr} \left( C_i^T P_i^* \right), \quad (4) $$

where $N_c$ is the number of classes.

In Eq. (3), the uniform distributions, i.e. $p_i(k) = \frac{1}{K}$ and $q_i(j) = \frac{1}{N_i}$, are usually used as priors, but this may be different from the ones of real data, thus leading to performance drop. To address this issue, we develop the BoIW model to estimate the intra-class distribution of $\{w_i^k\}_{k=1}^K$.

Concretely, BoIW firstly constructs a queue-based vocabulary $V_i$ of size $N$, as the representative feature set for the $i$-th class. As shown in Fig. 4, $V_i$ is updated in each minibatch, where $m$ positive instances are selected, and their features are inserted into $V_i$ by removing the $m$ oldest ones at the mean time. For a two-stage detector, we extract the flattened instance-level features after the RoI layer. As to the one-stage detector, we additionally employ a convolutional layer to extract $C$-dimensional instance features.

Subsequently, $K$-means is applied to $V_i$ and $K$ clusters $\{c_i^k\}$ are obtained. The marginal distribution $p_i$ is therefore estimated as $p_i(k) = \#|c_i^k| / \#|V_i|$, where $\#|\cdot|$ denotes the number of elements. Considering that the clusters $\{c_i^k\}$ vary in different steps, we sort $p_i$ in the descending order to ensure that $w_i^k$ always corresponds to the cluster with the $k$-th highest probability. By taking $p_i(k)$ back to Eqs. (3) and (4), we have the BoIW induced optimal transport loss.

Note that the vocabulary $V_i$ is the representative feature set for the $i$-th class. To further enhance their representativeness, we adopt another contrastive learning loss w.r.t. $V_i$ and the positive instance feature $x_i$:

$$ \mathcal{L}_{ct} = -\frac{1}{N} \sum_{i=1}^{N} \log \left( \frac{\sum_{u \in V_i} \exp(u^T x_i)}{\sum_{u \in V} \exp(u^T x_i)} \right), \quad (5) $$
where $V = \bigcup_{i=1}^{N} V_i$.

By minimizing $L_{cl}$ in Eq. (5), the correlations between the intra-class/inter-class features are increased/decreased.

### Adaptive $K$-Proxy Estimation

The number of proxies within each class, i.e., $K_i$, is important to MP-Det. Instead of the trivial solution to set it manually, we propose an adaptive way for estimation. To be specific, we extract instance features by the vanilla GFL v1 (Li et al. 2020b) and perform clustering by DBSCAN (Ester et al. 1996) to determine $K_i$.

### Overall Optimization

The overall training loss of MP-Det combines the conventional detection loss $L_{det}$, the BoIW induced optimal transportation loss $L_{ot}$ in Eq. (3) and the contrastive learning loss $L_{cl}$ in Eq. (5), formulated as:

$$L = L_{det} + L_{ot} + L_{cl}. \quad (6)$$

During training, we perform BoIW to regularly estimate $p_t$ (for every 2,000 iterations in our case). When optimizing $L_{ot}$, we employ Sinkhorn-Knopp (Cuturi 2013) to compute the transportation plan $P^*_t$, which only slightly increases the training time (27.5h vs. 33h) without any extra cost at test.

## Experimental Results and Analysis

UFPMP-Det is evaluated on the widely-used VisDrone (Zhu et al. 2018) and UAVDT (Du et al. 2018) benchmarks and extensive experiments are carried out.

### Datasets and Protocols

VisDrone consists of 10,209 high resolution images ($2000 \times 1500$) with 10 object categories, captured by various drone-mounted cameras from different areas (urban and country) and scenes (sparse and crowded). 6,471 images are used for training, 548 for validation and 3,190 for test. Since the test set is not publicly available, we follow ClutDet (Yang et al. 2019) and DMNet (Li et al. 2020a) to report scores on the validation set. UAVDT includes 23,258 images for training and 15,069 images for test. All the images are captured from urban areas by a UAV at low altitudes with a 1080 $\times$ 540 resolution. Three kinds of vehicles (car, bus, and truck) are manually labeled. Similar to the protocols for general object detection (Lin et al. 2014), we adopt Average Precision (AP) and APs at the IoU thresholds of 0.5 (AP50) and 0.75 (AP75) as the metrics on both datasets.

### Comparison with the State-of-the-arts

We compare UFPMP-Det with the state-of-the-art counterparts, including Faster-RCNN (Ren et al. 2017), ClusDet.
Table 4: Comparison of packing methods in AP/AP50/AP75 (%) and number of packed images (#img) based on Faster-RCNN with ResNet-50.

<table>
<thead>
<tr>
<th>Method</th>
<th>Reference</th>
<th>#img</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
</tr>
</thead>
<tbody>
<tr>
<td>EIP</td>
<td>(Yang et al. 2019)</td>
<td>3288</td>
<td>21.1</td>
<td>44.0</td>
<td>18.1</td>
</tr>
<tr>
<td>ClusDet</td>
<td>(Yang et al. 2019)</td>
<td>2716</td>
<td>26.7</td>
<td>50.6</td>
<td>24.7</td>
</tr>
<tr>
<td>DMNet</td>
<td>(Li et al. 2020a)</td>
<td>2736</td>
<td>28.2</td>
<td>47.6</td>
<td>28.9</td>
</tr>
<tr>
<td>UFP</td>
<td>Ours</td>
<td>1096</td>
<td>30.6</td>
<td>52.5</td>
<td>31.0</td>
</tr>
</tbody>
</table>

Table 5: Ablation study on UFP w.r.t. Foreground Ratio (FR) (%) and proportion (%) of object instances in small, medium and large sizes (using the COCO metric) on VisDrone and UAVDT.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>UFP</th>
<th>FR</th>
<th>Small</th>
<th>Medium</th>
<th>Large</th>
</tr>
</thead>
<tbody>
<tr>
<td>VisDrone</td>
<td>✓</td>
<td>10.2</td>
<td>68.56</td>
<td>28.68</td>
<td>2.76</td>
</tr>
<tr>
<td>VisDrone</td>
<td></td>
<td>24.5</td>
<td>6.96</td>
<td>63.35</td>
<td>29.69</td>
</tr>
<tr>
<td>UAVDT</td>
<td>✓</td>
<td>5.11</td>
<td>74.87</td>
<td>23.01</td>
<td>2.12</td>
</tr>
<tr>
<td>UAVDT</td>
<td></td>
<td>22.98</td>
<td>0.64</td>
<td>71.60</td>
<td>27.76</td>
</tr>
</tbody>
</table>

Table 6: Validation of different components in MP-Det with ResNet-101 on VisDrone in terms of AP/AP50/AP75 (%).

<table>
<thead>
<tr>
<th>Method</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>29.6</td>
<td>49.8</td>
<td>30.3</td>
</tr>
<tr>
<td>Baseline+UFP</td>
<td>36.6</td>
<td>62.3</td>
<td>36.8</td>
</tr>
<tr>
<td>Baseline+UFP+MP-Head</td>
<td>37.0</td>
<td>62.5</td>
<td>37.6</td>
</tr>
<tr>
<td>Baseline+UFP+MP-Head+BoIW</td>
<td>37.5</td>
<td>63.2</td>
<td>38.3</td>
</tr>
</tbody>
</table>

On UFP. We evaluate UFP and compare it to three counterparts: i.e. evenly image partition (EIP) (Yang et al. 2019), ClusDet (Yang et al. 2019), and DMNet (Li et al. 2020a). For fair comparison, all the methods utilize Faster-RCNN with FPN as the base detector and ResNet-50 as the backbone. As shown in Table 4, UFP outputs less packed images by performing unified adaptive packing, whilst achieving the best accuracy by increasing the Foreground Ratio (FR). To show the advantage of UFP in increasing FR and decreasing the number of small objects, we summarize the FR as well as the percentages of small/medium/large objects on VisDrone and UAVDT in Table 5. It is worth noting that we utilize the metric used in MS COCO to determine whether an object is small, medium or large. As demonstrated, UFP clearly promotes FR and significantly reduces the percentage of small objects, thereby facilitating successive detection.

We visualize the intermediate and the final outputs of UFP in Fig. 5. The first row shows the input images; the second row depicts the FRG clusters from the object regions densely extracted by the foreground detector; and the last row displays the packed unified mosaics.
<table>
<thead>
<tr>
<th>Method</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K = 10$ (Manual)</td>
<td>37.1</td>
<td>62.8</td>
<td>37.8</td>
</tr>
<tr>
<td>$K = 20$ (Manual)</td>
<td>37.1</td>
<td>62.6</td>
<td>37.7</td>
</tr>
<tr>
<td>MP-Det</td>
<td>37.5</td>
<td>63.2</td>
<td>38.3</td>
</tr>
</tbody>
</table>

Table 7: Comparison of AP/AP50/AP75 (%) on VisDrone by using different methods to set the number of proxies $K$: Manual vs. MP-Det (Ours).

<table>
<thead>
<tr>
<th>Size of BoIW ($N$)</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>22.3</td>
<td>36.1</td>
<td>25.2</td>
</tr>
<tr>
<td>100</td>
<td>24.0</td>
<td>37.6</td>
<td>27.1</td>
</tr>
<tr>
<td>200</td>
<td>24.6</td>
<td>38.7</td>
<td>28.0</td>
</tr>
</tbody>
</table>

Table 8: The impact of the size of BoIW ($N$) on the performance of MP-Det w.r.t. AP/AP50/AP75 (%) on UAVDT.

<table>
<thead>
<tr>
<th>$\beta$</th>
<th>FR</th>
<th>AP</th>
<th>AP50</th>
<th>AP75</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.3</td>
<td>32.46</td>
<td>33.9</td>
<td>61.7</td>
<td>33.5</td>
</tr>
<tr>
<td>1.5</td>
<td>24.53</td>
<td>36.6</td>
<td>62.4</td>
<td>36.7</td>
</tr>
<tr>
<td>1.7</td>
<td>21.07</td>
<td>35.6</td>
<td>61.3</td>
<td>35.8</td>
</tr>
</tbody>
</table>

Table 9: The impact of $\beta$ on UFP w.r.t. Foreground Ratio (FR) and AP/AP50/AP75 (%) on VisDrone.

**On MP-Det.** MP-Det has two major components: i.e., MP-Head and BoIW, and we validate them on a stronger baseline with ResNet-101. As shown in Table 6, MP-Head first improves the baseline accuracy (AP) by 0.4% and BoIW further increases it by 0.5%, highlighting their credits.

We further evaluate the effects of adaptive $K$−proxy estimation and the size of BoIW on the performance of MP-Det. As aforementioned, MP-Det adaptively estimates the number of proxies for each class by performing DBSCAN on the instance features extracted from pre-trained models. For comparison, we choose the way to set it manually as the baseline. As shown in Table 7, manually setting is not as good as adaptively setting when used in MP-Det.

As to the size of BoIW, we report the results of MP-Det by using different values, e.g., $N = 50/100/200$ on UAVDT in Table 8. As summarized, MP-Det achieves the highest score with $N = 200$, which is therefore adopted in our work.

We qualitatively demonstrate the impact of optimal transport (OT) based feature-proxy matching, by visualizing the features of object instances as well as the proxies on UAVDT via $t$-SNE. As shown in Fig. 6, without OT, the features of instances tend to gather to non-proxy points and are apart from the proxies, and the proxies from different classes are not discriminative enough. In contrast, when OT is employed, the learned features locate in the small neighborhoods of multiple proxies in a more uniformly distributed way. Besides, the inter-class distances between proxies are clearly enlarged, making it easier for object classification.

**On Hyper-parameter $\beta$.** As described, $\beta$ affects the number of clustered regions and the average recall of raw images. In Table 9, we report the detection accuracies for different values of $\beta$. The results indicate that FR decreases as $\beta$ increases, and UFP reaches the highest performance when $\beta = 1.5$, which is therefore used in our experiments.

**Conclusion**

In this paper, we propose a novel approach, namely UFPMP-Det, to object detection on drone imagery. It first introduces the UFP module to address the instances of very small scales by generating single mosaics of input images with largely increased foreground ratios, substantially improving both the accuracy and efficiency. The MP-Det module is further presented to model complex object distributions through multiple proxy learning, where the proxies are enforced to be diverse by minimizing a Bag-of-Instance-Words guided optimal transport loss. Extensive experiments are conducted on two benchmarks, and UFPMP-Det reaches the new state-of-the-art, highlighting its effectiveness.

**Acknowledgments**

This work is partly supported by the National Natural Science Foundation of China (No. 62022011), the Research Program of State Key Laboratory of Software Development Environment (SKLSDE-2021ZX-04), and the Fundamental Research Funds for the Central Universities.
References
Gong, Y.; Yu, X.; Ding, Y.; Peng, X.; Zhao, J.; and Han, Z. 2021. Effective Fusion Factor in FPN for Tiny Object Detection. In WACV.
Li, J.; Liang, X.; Wei, Y.; Xu, T.; Feng, J.; and Yan, S. 2017. Perceptual Generative Adversarial Networks for Small Object Detection. In CVPR.
Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S. E.; Fu, C.; and Berg, A. C. 2016. SSD: Single Shot MultiBox Detector. In ECCV.
Liu, Z.; Gao, G.; Sun, L.; and Fang, Z. 2021b. HRDNet: high-resolution detection network for small objects. In ICME.
Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In ICCV.
Yu, X.; Gong, Y.; Jiang, N.; Ye, Q.; and Han, Z. 2020. Scale Match for Tiny Person Detection. In WACV.