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

Yecheng Huang[1, 2]**, Jiaxin Chen**[2]**, Di Huang**[1, 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.
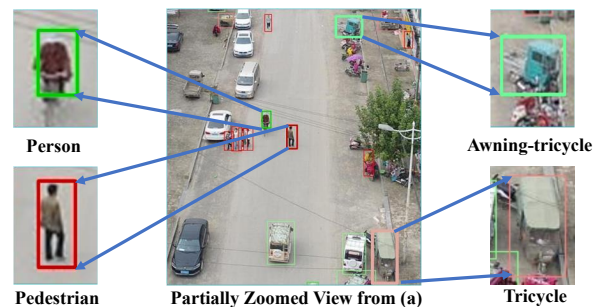
Introduction

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

(a) Instance scales in MS COCO (left) *vs.* VisDrone (right)

Person · **Awning-tricycle**
Pedestrian · **Partially Zoomed View from (a)** · **Tricycle**

(b) Semantically similar categories on VisDrone

Figure 1: Challenges in object detection on drone imagery.

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 (UFPMP-Det). 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.* Vis-Drone 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 Çigla 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-

(a) Image Pyramid

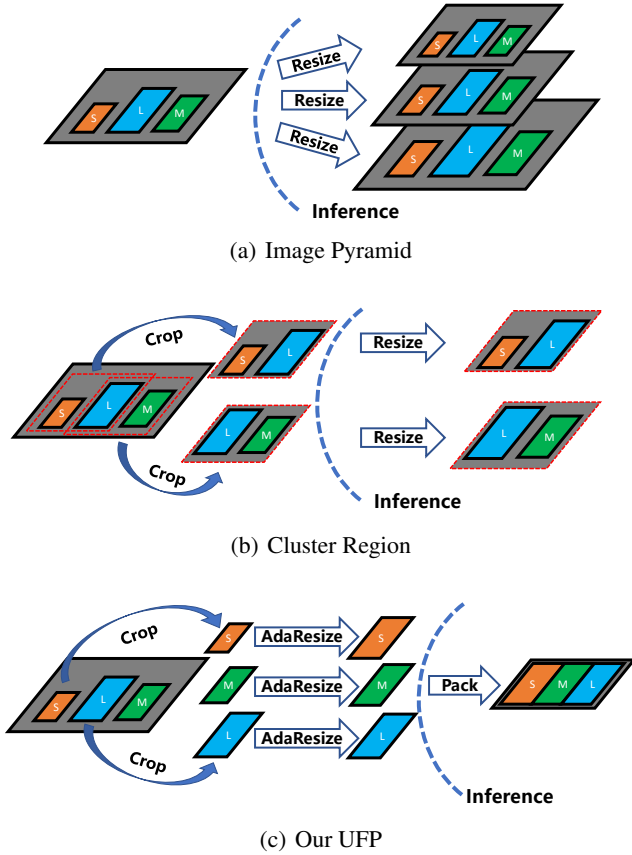(b) Cluster Region

(c) Our UFP

Figure 2: Comparison of different pipelines when handling scale changes. Orange, green and blue boxes indicate small, medium and large scale objects, respectively.

lenges, where the scale distribution of instances on drone images is quite different and the object categories share semantic similarity as shown in Fig. 1 (a) and (b), thereby making them problematic to the given issue.

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,

Algorithm 1: Foreground Region Generation

Input: Bounding boxes $\mathcal{B}_c$
Output: Merged regions $\mathcal{B}_r$

1: Initialize $\mathcal{B}_r = \emptyset$.
2: **while** $\mathcal{B}_c \neq \emptyset$ **do**
3:     $A = \mathrm{argmin}_{A' \in \mathcal{B}_c} |A'|$
4:     $\mathcal{B}_c := \mathcal{B}_c - \{A\}$
5:     **for all** $B \in \mathcal{B}_c$ **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:             $\mathcal{B}_c := \mathcal{B}_c - \{B\}$
10:         **end if**
11:     **end for**
12:     $\mathcal{B}_r := \mathcal{B}_r \cup \{A\}$
13: **end while**
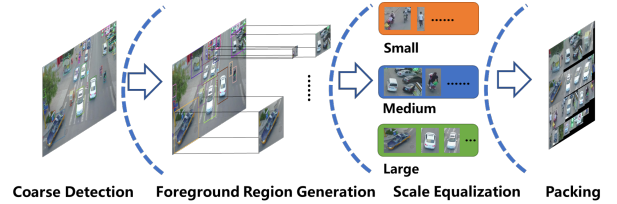14: **return** $\mathcal{B}_r$

Figure 3: Pipeline of Unified Foreground Packing (UFP).

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 $\mathcal{B}_c$ as input and selects the box A with the minimal size as the generation starting point. For each box B in $\mathcal{B}_c - \{A\}$, FRG searches the smallest convex box C that encloses A and B. If the sum of the areas *w.r.t.* A and B, *i.e.* $|A| + |B|$, is larger than that of C, we update A with C and remove B from $\mathcal{B}_c$. 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 $\mathcal{B}_r$. We repeat the procedure above until $\mathcal{B}_c$ turns to an empty set, when we obtain the final merged region set $\mathcal{B}_r$.

Foreground Region Scale Equalization   After FRG, each image is represented as several cluster regions with different scales. To equalize their scales, especially the small ones,
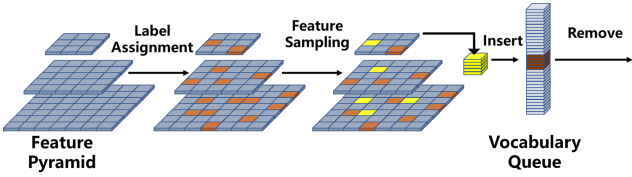
Figure 4: Illustration of updating the vocabulary queue. The orange cells indicate positive samples after label assignments, and the yellow ones are randomly selected to update the oldest elements in the vocabulary queue.

we first estimate the averaged region scale from $\mathcal{B}_r$ and subsequently enlarge the regions smaller than a fixed size (*e.g.* 96×96 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 $\boldsymbol{w}_i$ to the i-th category. Suppose that $\boldsymbol{x}$ is the feature extracted by the backbone network, its conditional probability corresponding to the i-th category is formulated as:

$$Pr(Y = y_i|\boldsymbol{x}) = \text{Sigmoid}(\boldsymbol{w}_i^{\text{T}}\boldsymbol{x}), \quad (1)$$

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 $\boldsymbol{w}_i/\|\boldsymbol{w}_i\|$.

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 $[\boldsymbol{w}_i^1, \boldsymbol{w}_i^2, ..., \boldsymbol{w}_i^K]$. Inspired by SoftTriple (Qian et al. 2019), the multi-proxy conditional probability is formulated as:

$$Pr(Y = y_i|\boldsymbol{x}) = \text{Sigmoid}\left(\gamma \sum_k \frac{\exp(s_i^k)s_i^k}{\sum_{l=1}^{K}\exp(s_i^l)}\right), \quad (2)$$

where $s_i^k = \frac{\boldsymbol{w}_i^{k\text{T}}\boldsymbol{x}}{\|\boldsymbol{w}_i^k\|\|\boldsymbol{x}\|}$ and γ is a scaling factor. Accordingly, the weights $[\boldsymbol{w}_i^1, \boldsymbol{w}_i^2, ..., \boldsymbol{w}_i^K]$ can be optimized by minimizing the cross-entropy or the focal loss *w.r.t.* $Pr(Y = y_i|x)$.

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 $\{\boldsymbol{f}_i^j\}_{j=1}^{N_i}$ of N_i positive instances and K proxies $\{\boldsymbol{w}_i^k\}_{k=1}^{K}$ for the i-th class, the cost matrix is computed as $\boldsymbol{C}_i = \frac{\mathbf{1}-\boldsymbol{S}}{2}$, where $\boldsymbol{S} \in \mathbb{R}^{N_i \times K}$ refers to the cosine similarity matrix between $\{\boldsymbol{f}_i^j\}_{j=1}^{N_i}$ and $\{\boldsymbol{w}_i^k\}_{k=1}^{K}$ and $\mathbf{1}$ is an all-ones matrix. The transportation plan $\boldsymbol{P}_i^* \in \mathbb{R}^{N_i \times K}$ is computed as below:

$$\boldsymbol{P}_i^* = \arg\min_{\boldsymbol{P}} \ \text{tr}\left(\boldsymbol{C}_i^T \boldsymbol{P}\right)$$
$$s.t. \ \sum_{j=1}^{N_i}\boldsymbol{P}(j,k) = \boldsymbol{p}_i(k), \sum_{k=1}^{K}\boldsymbol{P}(j,k) = \boldsymbol{q}_i(j), \quad (3)$$

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

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

where N_c is the number of classes.

In Eq. (3), the uniform distributions, *i.e.* $\boldsymbol{p}_i(k) = \frac{1}{K}$ and $\boldsymbol{q}_i(j) = \frac{1}{N_i}$, 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 $\{\boldsymbol{w}_i^k\}_{k=1}^{K}$.

Concretely, BoIW firstly constructs a queue-based vocabulary $\boldsymbol{V}_i$ of size N, as the representative feature set for the i-th class. As shown in Fig. 4, $\boldsymbol{V}_i$ is updated in each mini-batch, where m positive instances are selected, and their features are inserted into $\boldsymbol{V}_i$ 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 $\boldsymbol{V}_i$ and K clusters $\{\boldsymbol{c}_i^k\}$ are obtained. The marginal distribution $\boldsymbol{p}_i$ is therefore estimated as $\boldsymbol{p}_i(k) = \#|\boldsymbol{c}_i^k|/\#|\boldsymbol{V}_i|$, where $\#|\cdot|$ denotes the number of elements. Considering that the clusters $\{\boldsymbol{c}_i^k\}$ vary in different steps, we sort $\boldsymbol{p}_i$ in the descending order to ensure that $\boldsymbol{w}_i^k$ always corresponds to the cluster with the k-th highest probability. By taking $\boldsymbol{p}_i(k)$ back to Eqs. (3) and (4), we have the BoIW induced optimal transport loss.

Note that the vocabulary $\boldsymbol{V}_i$ is the representative feature set for the i-th class. To further enhance their representativeness, we adopt another contrastive learning loss *w.r.t.* $\boldsymbol{V}_i$ and the positive instance feature $\boldsymbol{x}_i$:

$$\mathcal{L}_{cl} = -\frac{1}{N}\sum_i \log\left(\frac{\sum_{\boldsymbol{v}\in\boldsymbol{V}_i}\exp(\boldsymbol{v}^{\text{T}}\boldsymbol{x}_i)}{\sum_{\boldsymbol{u}\in\boldsymbol{V}}\exp(\boldsymbol{u}^{\text{T}}\boldsymbol{x}_i)}\right), \quad (5)$$

Method	References	ResNet-50			ResNet-101			ResNeXt-101		
		AP	AP50	AP75	AP	AP50	AP75	AP	AP50	AP75
Faster-RCNN	(Ren et al. 2017)	21.4	40.7	19.9	21.4	40.7	20.3	21.8	41.8	20.1
ClusDet	(Yang et al. 2019)	26.7	50.6	24.4	26.7	50.4	25.2	32.4	56.2	31.6
DMNet	(Li et al. 2020a)	28.2	47.6	28.9	28.5	48.1	29.4	29.4	49.3	30.6
GLSAN	(Deng et al. 2021)	30.7	55.4	30.0	30.7	55.6	29.9	-	-	-
SAIC-FPN	(Zhou et al. 2019)	-	-	-	-	-	-	35.7	62.3	35.1
AMRNet	(Wei et al. 2020)	31.7	52.7	33.1	31.7	52.6	33.0	32.1	53.0	33.2
HRDNet	(Liu et al. 2021b)	-	-	-	31.4	53.3	31.6	35.5	62.0	35.1
UFPMP-Det	Ours	36.6	62.4	36.7	37.5	63.2	38.3	39.2	65.3	40.2
UFPMP-Det+MS	Ours	**37.4**	**63.7**	**37.7**	**38.7**	**65.1**	**39.4**	**40.1**	**66.8**	**41.3**

Table 1: Comparison of different approaches in AP/AP50/AP75 (%) on the validation set of VisDrone. MS refers to the multi-scale trick during inference and '-' indicates that the result is not reported.

where $V = \cup_{i=1}^{N_c} V_i$.

By minimizing $\mathcal{L}_{cl}$ 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, 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.

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

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

During training, we perform BoIW to regularly estimate p_i (for every 2,000 iterations in our case). When optimizing $\mathcal{L}_{ot}$, we employ Sinkhorn-Knopp (Cuturi 2013) to compute the transportation plan P_i^*, 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 (2000×1500) 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 1080×540 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.

Method	Reference	AP	AP50	AP75
Faster-RCNN	(Ren et al. 2017)	11.0	23.4	8.4
ClusDet	(Yang et al. 2019)	13.7	26.5	12.5
DMNet	(Li et al. 2020a)	14.7	24.6	16.3
GLSAN	(Deng et al. 2021)	17.0	28.1	18.8
DREN	(Zhang et al. 2019)	15.1	-	-
ARMNet	(Wei et al. 2020)	18.2	30.4	19.8
UFPMP-Det	Ours	**24.6**	**38.7**	**28.0**

Table 2: Comparison of different approaches with ResNet-50 in AP/AP50/AP75 (%) on UAVDT. '-' indicates that the result is not reported.

Method	Reference	#img	Inference Time
ClusDet	(Yang et al. 2019)	2716	0.273
DMNet	(Li et al. 2020a)	2736	0.290
UFPMP-Det	Ours	**1096**	**0.152**

Table 3: Comparison of different methods in efficiency *w.r.t.* the number of packed images (#img) and the inference time cost (in seconds) on VisDrone.

Implementation Details

We implement the proposed approach using the open-source *MMDetection* toolbox[1]. GFL (Li et al. 2020b) is employed as the baseline detector with the model pre-trained on ImageNet. UFPMP-Det is trained for 60 epochs in total by the SGD optimizer. The momentum and weight decay are fixed as 0.9 and 0.0001, respectively. The initial learning rate is set at 0.01 with a linear warm-up, which decreases by the factor of 10 after 40 and 55 epochs. As for BoIW, it is individually updated in the first 10 epochs without performing the optimal transport and contrastive learning, and jointly optimized for all the components afterwords, which is empirically stable during training in our experiments. The sizes of the input images for our detector are set to 1333×800 on VisDrone and 1000×600 on UAVDT, respectively.

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

---

[1]https://github.com/open-mmlab/mmdetection

Method	Reference	#img	AP	AP50	AP75
EIP	(Yang et al. 2019)	3288	21.1	44.0	18.1
ClusDet	(Yang et al. 2019)	2716	26.7	50.6	24.7
DMNet	(Li et al. 2020a)	2736	28.2	47.6	28.9
UFP	Ours	**1096**	**30.6**	**52.5**	**31.0**

Table 4: Comparison of packing methods in AP/AP50/AP75 (%) and number of packed images (#img) based on Faster-RCNN with ResNet-50.

Dataset	UFP	FR	Small	Medium	Large
VisDrone		10.2	68.56	28.68	2.76
VisDrone	✓	24.5	6.96	63.35	29.69
UAVDT		5.11	74.87	23.01	2.12
UAVDT	✓	22.98	0.64	71.60	27.76

Table 5: Ablation study on UPF *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.

(Yang et al. 2019), DMNet (Li et al. 2020a), GLSAN (Deng et al. 2021), and DREN (Zhang et al. 2019).

Results on VisDrone. Existing approaches deliver results by using different base networks on VisDrone, and we therefore report the performance of UFPMP-Det with various typical backbones, *i.e.* ResNet-50, ResNet-101, and ResNeXt-101, for more comprehensive validation. As summarized in Table 1, all these methods generally achieve higher accuracies through stronger networks. When using the same backbone, UFPMP-Det consistently outperforms such counterparts by large margins, improving APs of the second best by 4.9%, 5.8% and 3.5% with ResNet-50, ResNet-101 and ResNeXt-101, respectively. It is worth noting that the performance of UFPMP-Det with ResNet-50 is even superior to that with much deeper networks (*e.g.* ResNeXt-101), reaching the new state-of-the-art. Besides, the multi-scale technique during inference further promotes the accuracy.

Results on UAVDT. Most of detectors utilize the ResNet-50 backbone for evaluation on UAVDT, and we follow this setting for fair comparison. As illustrated in Table 2, UFPMP-Det largely boosts the performance of other detectors, and it improves the AP, AP50 and AP75 of the second best ARM-Net by 6.4%, 8.3% and 8.2%, respectively.

Overall Complexity. To analyze the efficiency of UFPMP-Det, we show the number of packed images as well as the inference time cost, in comparison to ClusDet (Yang et al. 2019) and DMNet (Li et al. 2020a). All the experiments are conducted on one GTX 1080TI GPU. As Table 3 displays, UFPMP-Det generates less than half of the packed images by ClusDet and DMNet and thus infers significantly faster, highlighting its advantage.

Ablation Study

We detailedly validate the major components, *i.e.*, UFP and MP-Det, as well as several hyper-parameters on UFPMP-Det.

Figure 5: Visualization of UFP. Top: input images; middle: clustered regions highlighted by green bounding boxes; and bottom: packed mosaics.

Method	AP	AP50	AP75
Baseline	29.6	49.8	30.3
Baseline+UFP	36.6	62.3	36.8
Baseline+UFP+MP-Head	37.0	62.5	37.6
Baseline+UFP+MP-Head+BoIW	37.5	63.2	38.3

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

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 UPF 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.

Method	AP	AP50	AP75
$K = 10$ (Manual)	37.1	62.8	37.8
$K = 20$ (Manual)	37.1	62.6	37.7
MP-Det	37.5	63.2	38.3

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).

Size of BoIW (N)	AP	AP50	AP75
50	22.3	36.1	25.2
100	24.0	37.6	27.1
200	24.6	38.7	28.0

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

β	FR	AP	AP50	AP75
1.3	32.46	33.9	61.7	33.5
1.5	24.53	36.6	62.4	36.7
1.7	21.07	35.6	61.3	35.8

Table 9: The impact of β 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 β. As described, β 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 β. The results indicate that FR decreases as
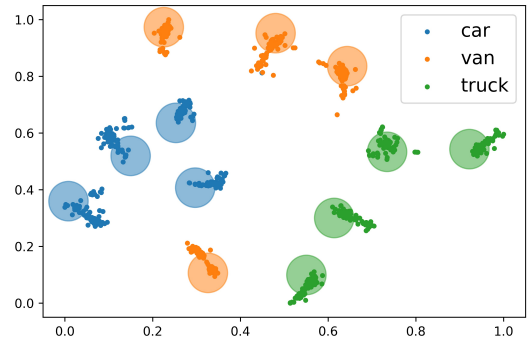
(a) with OT
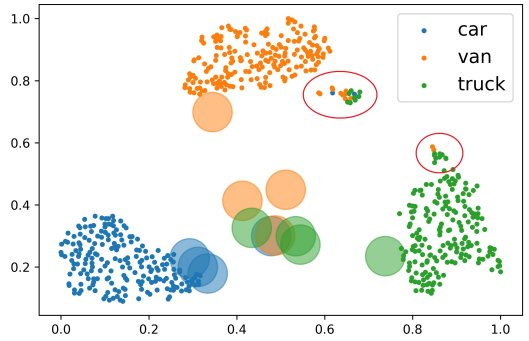
(b) w/o OT

Figure 6: Visualization of multi-proxies learned on UAVDT via t-SNE. Solid dots denote the features of object instances and translucent circles denote the proxies. Different colors indicate distinct classes.

β 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

References

Cai, Z.; and Vasconcelos, N. 2018. Cascade R-CNN: Delving Into High Quality Object Detection. In *CVPR*.

Cuturi, M. 2013. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In *NeurIPS*.

Deng, S.; Li, S.; Xie, K.; Song, W.; Liao, X.; Hao, A.; and Qin, H. 2021. A Global-Local Self-Adaptive Network for Drone-View Object Detection. *TIP*, 30: 1556–1569.

Du, D.; Qi, Y.; Yu, H.; Yang, Y.; Duan, K.; Li, G.; Zhang, W.; Huang, Q.; and Tian, Q. 2018. The Unmanned Aerial Vehicle Benchmark: Object Detection and Tracking. In *ECCV*.

Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X.; et al. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. In *KDD*.

Everingham, M.; Gool, L. V.; Williams, C. K. I.; Winn, J. M.; and Zisserman, A. 2010. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2): 303–338.

Girshick, R. B.; Donahue, J.; Darrell, T.; and Malik, J. 2014. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. In *CVPR*.

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*.

Guo, C.; Fan, B.; Zhang, Q.; Xiang, S.; and Pan, C. 2020. AugFPN: Improving Multi-Scale Feature Learning for Object Detection. In *CVPR*.

Li, C.; Yang, T.; Zhu, S.; Chen, C.; and Guan, S. 2020a. Density Map Guided Object Detection in Aerial Images. In *CVPRW*.

Li, J.; Liang, X.; Wei, Y.; Xu, T.; Feng, J.; and Yan, S. 2017. Perceptual Generative Adversarial Networks for Small Object Detection. In *CVPR*.

Li, X.; Wang, W.; Hu, X.; Li, J.; Tang, J.; and Yang, J. 2021. Generalized Focal Loss V2: Learning Reliable Localization Quality Estimation for Dense Object Detection. In *CVPR*.

Li, X.; Wang, W.; Wu, L.; Chen, S.; Hu, X.; Li, J.; Tang, J.; and Yang, J. 2020b. Generalized Focal Loss: Learning Qualified and Distributed Bounding Boxes for Dense Object Detection. In *NeurIPS*.

Lin, T.; Dollár, P.; Girshick, R. B.; He, K.; Hariharan, B.; and Belongie, S. J. 2017. Feature Pyramid Networks for Object Detection. In *CVPR*.

Lin, T.; Goyal, P.; Girshick, R. B.; He, K.; and Dollár, P. 2020. Focal Loss for Dense Object Detection. *TPAMI*, 42(2): 318–327.

Lin, T.; Maire, M.; Belongie, S. J.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft COCO: Common Objects in Context. In *ECCV*.

Liu, B.; Rao, Y.; Lu, J.; Zhou, J.; and Hsieh, C. 2021a. Multi-Proxy Wasserstein Classifier for Image Classification. In *AAAI*.

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. HRD-Net: high-resolution detection network for small objects. In *ICME*.

Najibi, M.; Singh, B.; and Davis, L. 2019. AutoFocus: Efficient Multi-Scale Inference. In *ICCV*.

Noh, J.; Bae, W.; Lee, W.; Seo, J.; and Kim, G. 2019. Better to Follow, Follow to Be Better: Towards Precise Supervision of Feature Super-Resolution for Small Object Detection. In *ICCV*.

Qian, Q.; Shang, L.; Sun, B.; Hu, J.; Tacoma, T.; Li, H.; and Jin, R. 2019. SoftTriple Loss: Deep Metric Learning Without Triplet Sampling. In *ICCV*.

Redmon, J.; and Farhadi, A. 2018. YOLOv3: An Incremental Improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S.; He, K.; Girshick, R. B.; and Sun, J. 2017. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *TPAMI*, 39(6): 1137–1149.

Singh, B.; Najibi, M.; and Davis, L. S. 2018. SNIPER: Efficient Multi-Scale Training. In *NeurIPS*.

Tian, Z.; Shen, C.; Chen, H.; and He, T. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *ICCV*.

Unel, F. O.; Özkalayci, B. O.; and Çigla, C. 2019. The Power of Tiling for Small Object Detection. In *CVPRW*.

Wei, Z.; Duan, C.; Song, X.; Tian, Y.; and Wang, H. 2020. AMRNet: Chips Augmentation in Aerial Images Object Detection. *arXiv preprint arXiv:2009.07168*.

Yang, F.; Fan, H.; Chu, P.; Blasch, E.; and Ling, H. 2019. Clustered Object Detection in Aerial Images. In *ICCV*.

Yu, X.; Gong, Y.; Jiang, N.; Ye, Q.; and Han, Z. 2020. Scale Match for Tiny Person Detection. In *WACV*.

Zhang, D.; Shi, L.; Leung, S. C.; and Wu, T. 2016. A priority heuristic for the guillotine rectangular packing problem. *IPL*, 116(1): 15–21.

Zhang, J.; Huang, J.; Chen, X.; and Zhang, D. 2019. How to Fully Exploit The Abilities of Aerial Image Detectors. In *ICCVW*.

Zhou, J.; Vong, C.; Liu, Q.; and Wang, Z. 2019. Scale adaptive image cropping for UAV object detection. *NEUCOM*, 366: 305–313.

Zhu, C.; He, Y.; and Savvides, M. 2019. Feature Selective Anchor-Free Module for Single-Shot Object Detection. In *CVPR*.

Zhu, P.; Wen, L.; Bian, X.; Ling, H.; and Hu, Q. 2018. Vision Meets Drones: A Challenge. *arXiv preprint arXiv:1804.07437*.