PENet: Object Detection using Points Estimation in Aerial Images

Ziyang Tang\(^1\) , Xiang Liu\(^1\), Guangyu Shen\(^1\) and Baijian Yang\(^1\)

Purdue University
\{tang385, xiang35, shen447, byang\}@purdue.edu

Abstract

Aerial imagery has been increasingly adopted in mission-critical tasks, such as traffic surveillance, smart cities, and disaster assistance. However, identifying objects from aerial images faces the following challenges: 1) objects of interests are often too small and too dense relative to the images; 2) objects of interests are often in different relative sizes; and 3) the number of objects in each category is imbalanced. A novel network structure, Points Estimated Network (PENet), is proposed in this work to answer these challenges. PENet uses a Mask Resampling Module (MRM) to augment the imbalanced datasets, a coarse anchor-free detector (CPEN) to effectively predict the center points of the small objects. An adaptive merge algorithm Non-maximum Merge (NMM) is implemented in CPEN to address the issue of detecting dense small objects, and a fine anchor-free detector FPEN to locate the precise positions of the small objects. An adaptive merge algorithm Non-maximum Merge (NMM) is implemented in CPEN to address the issue of detecting dense small objects, and a hierarchical loss is defined in FPEN to further improve the classification accuracy. Our extensive experiments on aerial datasets visDrone (Zhu et al., 2018) and UAVDT (Du et al., 2018) showed that PENet achieved higher precision results than existing state-of-the-art approaches. Our best model achieved 8.7% improvement on visDrone and 20.3% on UAVDT.

1 Introduction

Unmanned Aerial Vehicles (UAV) equipped with a high-resolution camera can be applied to a wide range of applications, including traffic surveillance, smart cities assistance, and disaster respond and recovery. These aerial images are different from the natural imagery datasets, such as COCO (Lin et al., 2014), ImageNet (Deng et al., 2009) and Pascal VOC (Everingham et al., 2010) in the following perspectives: (1) the aerial images are in high-resolution, e.g. the images in visDrone (Zhu et al., 2018) are about 2,000 × 1,500 pixels, while most of the images are less than 500 × 500 pixels in COCO (Lin et al., 2014); (2) the aerial dataset contains both small, dense objects and sparse, large objects. This is because the elevation of the UAVs changes when collecting the images. As shown in Figure 1, the size of cars in Figure 1b and Figure 1a varies from extremely small to relatively large; (3) current public aerial imagery datasets are fairly imbalanced with certain types of objects poorly represented. For instance, the total number of cars in visDrone (Zhu et al., 2018) are 144,866, while the number of awning tricycle in the same datasets is merely 3,246. In addition, different categories are not equally distinctive: some share more similarities than others. e.g. In visDrone (Zhu et al., 2018) dataset, the pedestrian category shares many features with the people category but has little in common with the cars category. In the rest of this paper, we denote this phenomenon as Categorical Similarity Problems (CSP).

These differences between aerial images and natural images makes it challenging to detect objects in aerial images. Recently, CNN-based detectors demonstrated better performance than the handcrafted detectors (Moranduzzo and Melgani, 2014). However, these CNN detectors, including R-CNN families (Girshick, 2015; He et al., 2017), Yolo and SSD families (Redmon et al., 2016; Pedoeem and Huang, 2018; Liu et al., 2016) and CornerNet families (Law and Deng, 2018; Zhou et al., 2019), are all trained from natural images. Due to the aforementioned differences, directly applying the CNN detectors to aerial images will not achieve promising results. In particular, the lack of good processing strategy in high-resolution images and insufficient samples of certain categories are the main reasons to be accounted for.

"Coarse-to-fine” pipeline was proposed to deal with the problem of resolution and size difference. When the object size is large, the detectors can detect it even after the
high-resolution images are down-sampled. When the object sizes are small, the detectors can first find the sub-regions, denoted as clusters, that contains a series of small objects, and detect the small objects based on the clusters. The pipeline is a trade-off of performance and computation efficiency. (Unel et al., 2018) split the images evenly to show the power of tiling in small object detection. (Gao et al., 2018) proposed a dynamic zoom-in strategy to speed up the detection with Reinforcement Learning. (Yang et al., 2019) used K-means to generate high-quality annotations to train a network to find better clusters. To relieve the symptoms of insufficient data samples, (Chen et al., 2019; Zhang et al., 2019) reused the samples from the original images to augment the dataset.

In this paper, we propose the novel PENet structure to detect objects in aerial images. Illustrated in Figure 2, PENet has three components: Mask Re-sampling Module (MRM), Coarse-PENet (CPEN), and Fine-PENet (FPEN). MRM is added between the raw images and CPEN to augment object samples, especially those small confusing objects. CPEN generates the center points of high-quality clusters of small objects using an algorithm Non Maximum Merge (NMM). The cluster chips are then fed into FPEN for further precise detection. Meanwhile, the original images are also forwarded to FPEN to detect large objects from the aerial images. The output from FPEN are combined to provide the final detection results. When training FPEN, a hierarchical loss is applied to address the CSP. In addition, the hierarchical loss makes it possible to jointly train a robust model for different aerial datasets.

Our contributions are as follows:

- We proposed a novel PENet to detect objects of various sizes from high resolution and imbalanced datasets.
- We presented an re-sampling algorithm MRM, a cluster generating algorithm NMM and a hierarchical loss approach. Each of them can be independently applied to CNN-based detectors.
- We achieved state-of-the-art results in visDrone (Zhu et al., 2018) and UAVDT (Du et al., 2018).

2 Related Works

In this section, we review data augmentation methods, benchmarks of object detectors, both anchor-based and anchor-free, followed by recent explorations in aerial images. Due to page limit, we focus on the most relevant work and highlight the differences between the proposed solution and the existing approaches.

2.1 Data Augmentation

To enlarge dataset, researchers have implemented many methods to augment the data, such as random crop, flipping, and inputs with multi-scale. In aerial imagery datasets, an observation is that the instances from different categories are imbalanced. To mitigate the issues caused by the imbalanced data, (Kisantal et al., 2019) pasted small objects to random positions on the training images to improve performance. (Chen et al., 2019) improved this approach by considering perspective and reasonableness. They argued that the pasted car instance should be on the road instead of ”flying” in the sky. Therefore they proposed Adasampling to logistically re-sample the instances. They also discovered that segmentation results are noisy if pre-trained models from natural image datasets are used unmodified. Eroding and median filters were used in their work to partially remove the noises. In contrast, we annotated the segmentation labels and fine-tuned the segmentation models from the aerial images datasets. The high-quality segmentation results act as masks and provide prior information to enable one-step small object pasting. As a result, our MRM can quickly generate augmented data with better quality.

2.2 CNN-based Object Detectors

The CNN-based object detectors can be divided into two major types: anchor-based detectors and anchor-free detectors. The anchor-based detectors use generated anchor candidates to locate the objects and use a fixed IoU to suppress the negative anchors. In recent years, the most representative anchor-based object detectors are the Faster-RCNN families (Girshick, 2015; He et al., 2017), YoLo families (Redmon et al., 2016; Redmon and Farhadi, 2017), and SSD families (Liu et al., 2016). However, the anchor-based detectors rely on good prior anchor sizes, which are often ’inherited’ from different datasets. Problem arises when the ratio of the height and the width is significantly different, making the prior anchor size hard to be determined. In addition, most anchor candidates are negative, leading to an imbalance between negative and positive anchors and resulting poor performance in final detection. To address the imbalance in negative and positive anchor candidates, RetinaNet (Lin et al., 2017b) introduced focal loss by revising the standard cross-entropy loss.

Figure 2: The Overall structure of PENet. MRM augments small objects in the images while CPEN locates the cluster chips from the original high-resolution images. A position refinement is then applied to the candidate cluster chips and the output is fed into FPEN which also takes the input of the original image. The prediction results of FPEN are fused together as the final detection results.

In this section, we review data augmentation methods, benchmarks of object detectors, both anchor-based and anchor-free, followed by recent explorations in aerial images. Due to page limit, we focus on the most relevant work and highlight the differences between the proposed solution and the existing approaches.

2.1 Data Augmentation

To enlarge dataset, researchers have implemented many methods to augment the data, such as random crop, flipping, and inputs with multi-scale. In aerial imagery datasets, an observation is that the instances from different categories are imbalanced. To mitigate the issues caused by the imbalanced data, (Kisantal et al., 2019) pasted small objects to random positions on the training images to improve performance. (Chen et al., 2019) improved this approach by considering perspective and reasonableness. They argued that the pasted car instance should be on the road instead of ”flying” in the sky. Therefore they proposed Adasampling to logistically re-sample the instances. They also discovered that segmentation results are noisy if pre-trained models from natural image datasets are used unmodified. Eroding and median filters were used in their work to partially remove the noises. In contrast, we annotated the segmentation labels and fine-tuned the segmentation models from the aerial images datasets. The high-quality segmentation results act as masks and provide prior information to enable one-step small object pasting. As a result, our MRM can quickly generate augmented data with better quality.

2.2 CNN-based Object Detectors

The CNN-based object detectors can be divided into two major types: anchor-based detectors and anchor-free detectors. The anchor-based detectors use generated anchor candidates to locate the objects and use a fixed IoU to suppress the negative anchors. In recent years, the most representative anchor-based object detectors are the Faster-RCNN families (Girshick, 2015; He et al., 2017), YoLo families (Redmon et al., 2016; Redmon and Farhadi, 2017), and SSD families (Liu et al., 2016). However, the anchor-based detectors rely on good prior anchor sizes, which are often ’inherited’ from different datasets. Problem arises when the ratio of the height and the width is significantly different, making the prior anchor size hard to be determined. In addition, most anchor candidates are negative, leading to an imbalance between negative and positive anchors and resulting poor performance in final detection. To address the imbalance in negative and positive anchor candidates, RetinaNet (Lin et al., 2017b) introduced focal loss by revising the standard cross-entropy loss.
The anchor-free detectors detect the objects based on point estimation. DeNet (Tychsen-Smith and Petersson, 2017) estimates the object locations by the top-left, bottom-left or bottom-right corner of a bounding box. It then generates the corner combinations and classifies the anchors using two-stage methods. Point Linking Network (PIL) (Wang et al., 2017) predicts the four locations as well as the center of the objects. CornerNet (Law and Deng, 2018) predicts the top-left and bottom-right corners of the bounding boxes and uses corner pooling to combine the corners to get the results. Keypoint estimation (Zhou et al., 2019) predicts the corners or the center points and achieved better performance than previous methods.

Our work adopted anchor-free detectors as the backbone. This is because it is difficult to find good prior anchor sizes that work for all the objects when they have different relative sizes. (LaLonde et al., 2018) also showed that it is easier to locate extremely small objects with points instead of anchors. Our further experiments also show that CNN detectors based on point estimation achieve better performance in aerial datasets.

2.3 Object Detection in Aerial Image

The challenge of detecting objects in aerial images is to locate the small objects in high-resolution images. Ultimately, the goal is to make it fast and accurate. However, to the best of our knowledge, none of the existing detectors can achieve this goal. A more realistic approach is to find a better trade-off between accuracy and computational efficiency. The workflow is to find low-resolution image chips from the original images; apply aforementioned detectors; and combine the results. Therefore, the task is to minimize the number of clusters while maintaining good accuracy. (Gao et al., 2018) used a reinforcement learning strategy to adaptively zoom in the sub-regions for fine detection. (Yang et al., 2019) designed a clusDet network and partitioned a high-resolution image into multiple small image chips for detection. (Unel et al.,) implemented the power of tiling for small object detection. While the existing methods are predominantly anchor-based, our proposed anchor-free PENet is different from all of them. Unlike (Gao et al., 2018), our CPEN can adaptively generate clusters for different images in the dataset. Compared to (Yang et al., 2019), our design does not need ROI pooling and scalenet. Instead, our CPEN can adaptively generate the clusters and remove some overlapped clusters with position refinement. (Unel et al.,) splits an image evenly with fixed sizes, while our approach offers a few options for the users to choose as a trade-off between performance and computation efficiency: the fastest version generates fewer image chips, while the best model achieves better accuracy.

3 Methodology

3.1 Overview

The structure of PENet is shown in Figure 2. We first use a Mask Resampling Module (MRM) for data augmentation. We then apply Non-Maximum Merge (NMM) algorithm to merge the small objects into several clusters of similar sizes. Through Coarse Points Estimation Net (CPEN), we get the candidate center points and the size of the clusters. The cluster chips are then processed by position refinement (PR) and results are fed into Fine Points Estimation Net (FPEN) to detect the center points and the sizes of small objects. The FPEN also employs hierarchical loss for better classification results. Large objects can be detected by using standard down-sampling approach by feeding the image direct into FPEN. The final results are obtained by fusing the local results from all the cluster chips and the global results from the original image.

3.2 MRM

One of the problems in current aerial images datasets is the imbalanced categories. For example, VisDrone (Zhu et al., 2018) has 144,866 cars instances in total, while the number of awning tricycle is merely 3,246. Without appropriate pre-processing, our experiments demonstrated that the accuracy on cars are far better than the accuracy on awning tricycles. Previous work in RRnet (Chen et al., 2019) showed that randomly re-sample the images may lead to background mismatch and scale mismatch. They proposed to use a pre-trained segmentation network to generate proper positions for data augmentation. However, the network is trained on natural datasets and the discrepancy between aerial images and natural images making the augmented data noisy and unde-
sirable. In our work, we annotated the segmentation labels for the aerial images, and fine-tuned the segmentation networks for aerial images, resulting in far better road maps. In the experiment, we annotated about 500 images from the aerial datasets, and use the fine-tuned model to predict the segmentation results. In addition, we also provided add-on masks of the ground truth to determine proper positions for the augmented objects. MRM generates a pool of confusing object images from the whole dataset. Objects are randomly selected from this image pool and pasted to each input image. Since the results of this segmentation network serve as a mask, we named this augmentation component Mask Re-sampling Module (MRM). As Figure 3 shows, the MRM pipeline uses the segmentation network as a mask. It then re-samples the confusing images from the image pool to synthesize the augmented data. In the experiments, we collected the confusing images from the training data and paste five objects onto each image.

3.3 CPEN

The goal of CPEN is to train a model that can estimate high-quality clusters of small objects. In previous work, (Yang et al., 2019) proposed a k-means algorithm to generate the ground truth of the clusters with each image having the same number of the clusters. However, in aerial images, the number of the clusters can be non-identical because of the difference in camera perspective, shooting time and other situations. Therefore a fixed number of clusters is not a good representative of all the situations. In this paper, we proposed Non-maximum Merge (NMM) algorithm to adaptively generate different number of clusters for different images. Shown in Algorithm 1, starting from the top-left boxes, we apply a cluster $C_i$ with size of $(w, h)$ and merge all the small objects $B$ in $C_i$. This process is repeated until all the objects are merged. $C$ is the set with all the clusters $C_i$. Note that some objects may be cropped into two parts during the merge process. Therefore, we set up a threshold $\tau$ to allow a slight overlap among the clusters. It could happen that the clusters may go beyond the boundary of the original images. Under this circumstance, the RECENT function relocates the center of $C_i$ to ensure the whole cluster is still in the image. During the inference stage, we predict the $\text{topk}$ center points of the clusters. Some clusters may share a large overlap with others. These candidate clusters will be removed by the position refinement (PR) operation. Figure 4 shows that the number of the clusters is reduced from 10 to 5 after PR. It is evident that PR helps to speed up the inference time while maintaining good performance. The trade-off of performance and computation time can be made by carefully choosing the value of $\text{topk}$ and the cluster overlap threshold $\tau$.

3.4 FPEN

FPEN is inspired by the hierarchical classification from YOLO9000 (Redmon and Farhadi, 2017), where a word tree strategy is used to handle classes with CSP issues. In the aerial dataset like visDrone (Zhu et al., 2018), the classes have similar CSP problems. For example, people class and pedestrian class share more similarities than people class and cars class. Therefore, we propose a hierarchical loss in our anchor-free detectors. The backbone network can be any detectors in the CornerNet series (Law and Deng, 2018; Zhou et al., 2019; Maninis et al., 2018). In our experiments, we implemented our hierarchical loss based on CenterNet (Zhou et al., 2019):

$$L_{shm} = \frac{-1}{N} \sum_{Y} \begin{cases} (1 - \hat{Y})^\alpha \log(\hat{Y}) & \text{if } Y = 1 \\ (1 - Y)^\beta \log(1 - \hat{Y}) & \text{otherwise} \end{cases}$$

(1)

Where $Y \in [0, 1]^{W \times H \times (C+C_s)}$, $W$, $H$, and $C$ represents width, height and number of classes, respectively. $R$ is the ratio for down-sampling. $C_s$ is the stacked classes. Followed the setting in (Zhou et al., 2019), we set $\alpha = 2$, $\beta = 4$ and $R = 4$. The settings of $C$ and $C_s$ are different for each dataset. For example, in visDrone (Zhu et al., 2018), $C = 11$ and $C_s = 3$ with additional classes of human, vehicles and non-motor-vehicles. To generate the bounding box from the predicted center points, we also implement a regression model with the $L_{uh}$ and $L_{off}$:

$$L_{uh} = \frac{1}{N} \sum_{k=1}^{N} |x_k - s_k|$$

$$L_{off} = \frac{1}{N} \sum_{p} |\hat{O}_p - (\frac{p}{R})|$$

(2)\( (3)\)

, where $N$ is the total number of the centerpoints, and $s$ represents the size of the bounding boxes. Therefore, the object function can be summarized as follows:

$$L = \lambda_{shm} L_{shm} + \lambda_{uh} L_{uh} + \lambda_{off} L_{off}$$

(4)

In our experiments, we set $\lambda_{shm} = 1, \lambda_{uh} = 0.1, \lambda_{off} = 1$. 

---

**Algorithm 1 Non Max Merge**

**Input:** Sorted bounding boxes $B = \{B_i\}_{i=1}^{N_B}$, classes of the bounding boxes, desired bounding box height $h_b$ and width $w_b$, non max merge threshold $\tau$  

**Output:** Merged clusters bounding boxes $C$

1: $\mathcal{C} = \{\}$  
2: for $i \leftarrow 1$ to $N_B$ do  
3: if $B_i$ is visited then  
4: continue  
5: Flag $B_i$ as visited  
6: $C_i \leftarrow \text{RECENTER}(B_i, h_b, w_b)$  
7: for $j = i + 1$ to $N_B$ do  
8: if IoU($C_i$, $B_j$) $> \tau$ then  
9: $\mathcal{C} \leftarrow \mathcal{C} \cup \{B_j\}$  
10: Flag $B_j$ as visited  
return $\mathcal{C}$

---

Figure 4: An example of the position refinement to get rid of some candidate clusters. In this example, we can get rid of 5 candidate clusters from the top 10 prediction results.
Table 1: The comparison of the performance in visDrone

| Methods                  | AP[%] | AP50[%] | AP75[%] |
|--------------------------|-------|---------|---------|
| RetinaNet (Lin et al., 2017b) | 11.81 | 21.37   | 11.62   |
| RefineDet (Zhang et al., 2018) | 14.90 | 28.76   | 14.08   |
| DetNet59 (Li et al., 2018)    | 15.26 | 29.23   | 14.34   |
| FPN (Lin et al., 2017a)      | 16.51 | 32.20   | 14.91   |
| Light-RCNN (Li et al., 2017) | 16.53 | 32.78   | 15.13   |
| CornetNet (Law and Deng, 2018) | 17.41 | 34.12   | 15.78   |
| Cascade-RCNN (Cai and Vasconcelos, 2018) | 16.09 | 31.91   | 15.01   |
| RPN (Chen et al., 2019)      | 29.62 | 34.00   | 28.70   |
| ClusDet (Yang et al., 2019)  | 32.4  | 56.2    | 31.6    |
| CenterNet (Zhou et al., 2019) | 14.2  | 19.3    | 15.5    |
| CPEN+CenterNet (Zhou et al., 2019) | 29.3  | 38.7    | 32.4    |
| CPEN+FPEN                  | 36.7  | 53.2    | 39.5    |
| CPEN+FPEN+MRM              | 41.1  | 58.0    | 44.3    |

3.5 Results Fusion

Final outputs are generated by merging the detection results from each cluster chip. To predict large objects, we feed-forward original images using standard down-sampling operations. The merge process is a standard Non Maximum Suppression. In some cases, an object may be cropped into two parts in different clusters and therefore have multiple bounding boxes. This can be fixed by merging the two bounding boxes into one larger box.

4 Experiments

We implemented the proposed PEnet on Pytorch 1.13. The models are trained and tested with two RTX 2080Ti GPUs. In the training phase, we trained our models for 200 epochs, with the learning rate starting from 1.25e^-4, decaying by 10 separately at epoch 90 and 120. Two public aerial image datasets: visDrone (Zhu et al., 2018) and UA VDT (Du et al., 2018) are used in our study. In visDrone, we chose Top10 cluster centers and chop the image chips into FPN. We then chose Top100 object center points for each image chip as the detection results. Whereas in UA VDT, we chose top5 in CPEN and top100 in FPN. The maximum detected objects is set to 500. The $\tau$ is set to 0.8 for all our experiments. We configured the same settings from (Zhou et al., 2019) and used maxpooling instead of standard NMS for evaluation. Figure 5 shows a few examples of the outputs of PEnet.

4.1 Evaluate Metrics

We adopted standard AP criterion from (Lin et al., 2014) to evaluate the performance of our methods against other benchmarks. AP is the average of all 10 IOU thresholds range from [0.5 : 0.95] with step size of 0.05 of all classes. AP50 and AP75 are computed with the IOU threshold set to 0.5, and 0.75 of all categories, respectively.

4.2 VisDrone

The datasets consists 6,471 training images and 548 validation images. Unfortunately, the server is shut down and we were not able to evaluate our results on the test set. Instead, validation set are used to evaluate different models. The whole dataset contains 12 classes: ignored regions, pedestrian, person, bicycle, car, van, truck, tricycle, awning tricycle, bus, motor, others. To implement hierarchical loss, we added three additional classes: Human, non-vehicles, and vehicles. The average resolution of the images is about 2,000 \times 1,500. Table 1 shows CPEN improves 15% than the baseline; FPN has a 7% improvement; and MRM is seen 5% increasement. Our best model increased 8.7% in mAP than the state-of-the-art detectors. It is worth noting that our proposed PEnet has better performance in AP75, indicating that our predictions are closer to the ground truth. A possible explanation for better results achieved by our model is that our prediction is based on the center points of the object and thus the predicted bounding boxes have bigger overlap with the ground truth.

4.3 UAVDT

UAVDT is a video aerial dataset that contains three categories of objects: car, bus, and truck. It has 80,000 annotated frames from 100 videos. The video sequences cover different weather conditions, including daylights, night and fog. The dataset also records from different perspectives, such as the perspective of front, side or bird view. The resolution of the frame is 1,024 \times 540 on average. The comparison of the performance is shown in table 2. Comparing against existing work, our best proposed PEnet improves the performance by 20.6% than the state-of-the-art. One explanation is that the previous work are anchor-based detectors. In such a challenging dataset with different weather conditions and distinct shooting views, it is really difficult to find a limited number of prior anchors to represent all the object with different sizes. The glamouring results proved that our point estimation approach is a good fit for detecting small objects in high resolution images.

4.4 Ablation Study

In this section, we perform an ablation study on the performance of each component of PEnet.

CPEN

CPEN uses the proposed NMM algorithm to adaptively generate the center points of the clusters. By providing more precise cluster chips, CPEN enjoys a significant improvement over the baseline of CenterNet (Zhou et al., 2019). In visDrone, CPEN improved by 15.1% compared to the baseline. In UAVDT, CPEN improved 38.1%. From Table 3, the mAP of class bicycle increased significantly from 1.94 to 18.96. The great improvement indicates the importance of extracting the clusters chips when detecting small object in aerial images. Note that the improvement comes from the high quality cluster chips generated from our CPEN, and our CPEN can be extended as an additional component for all other types of
CNN-based object detectors, including anchor-free detectors, e.g. CenterNet (Zhou et al., 2019), as well as anchor-based detectors, e.g. Faster-RCNN (Girshick, 2015), YOLO (Redmon et al., 2016) or SSD (Liu et al., 2016).

**FPEN**

FPEN employs hierarchical loss to address the CSP. Shown in Table 1, it is seen a 7.4% improvement in AP. The improvements of mAP in each category are also evaluated. Since UAVDT data only has three categories, it is not very beneficial to apply hierarchical loss. Results of visDrone dataset, shown in Table 3, reveals that adding hierarchical loss can improve the mAP for all the categories, especially those small objects, e.g. bicycle and tricycle. The promising results demonstrated the importance of addressing the CSP in CNN-based detectors. Another noteworthy advantage is that hierarchical loss makes it possible to train a model with multiple different datasets. As a data-driven technique, more data from different distribution will empower CNN-based detectors to be more robust and to have better generalization ability.

**MRM**

As aforementioned, we extract the road maps and logistically add instance patches. In the experiments, we added five instances in each input image. Table 3 shows MRM improved the mAP by up to 4.4% in the final performance of visDrone data and 4.7% in UAVDT data. The results show the value of augmenting data in improving the final performance. Note that MRM is a generalized data augmentation approach and can be implemented on other CNN-based object detectors as an alternative of re-sampling modules.

## 5 Conclusion

In this paper, we presented a novel object detector PEnet for object detections in aerial image datasets. CPEN can estimate the most relevant cluster chips. FPEN can predict the precise location of small objects and address the importance of CSP in detection tasks. MRM provides an alternative re-sampling method as a data augmentation approach. Our experiments showed that PEnet achieved state-of-the-art performance in two aerial imagery datasets, visDrone (Zhu et al., 2018) and UAVDT (Du et al., 2018).

### References

(Cai and Vasconcelos, 2018) Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 6154–6162, 2018.
(Chen et al., 2019) Changrui Chen, Yu Zhang, Qingxuan Lv, Shuo Wei, Xiaorui Wang, Xin Sun, and Junyu Dong. Rnet: A hybrid detector for object detection in drone-captured images. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 0–0, 2019.

(Deng et al., 2009) Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition, pages 248–255, 2009.

(Du et al., 2018) Dawei Du, Yuankai Qi, Hongyang Yu, Yifan Yang, Kaiwen Duan, Guorong Li, Weigang Zhang, Qingming Huang, and Qi Tian. The unmanned aerial vehicle benchmark: Object detection and tracking. In Proceedings of the European Conference on Computer Vision (ECCV), pages 370–386, 2018.

(Everingham et al., 2010) Mark Everingham, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes (voc) challenge. International journal of computer vision, 88(2):303–338, 2010.

(Gao et al., 2018) Mingfei Gao, Ruichi Yu, Ang Li, Vlad I Morariu, and Larry S Davis. Dynamic zoom-in network for fast object detection in large images. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 6926–6935, 2018.

(Girshick, 2015) Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

(Hel et al., 2017) 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.

(Kisantal et al., 2019) Mate Kisantal, Zbigniew Wojna, Jakub Murovski, Jacek Naruniec, and KyungHyun Cho. Augmentation for small object detection. arXiv preprint arXiv:1902.07296, 2019.

(LaLonde et al., 2018) Rodney LaLonde, Dong Zhang, and Mubarak Shah. Clusternet: Detecting small objects in large scenes by exploiting spatio-temporal information. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4003–4012, 2018.

(Law and Deng, 2018) Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In Proceedings of the European Conference on Computer Vision (ECCV), pages 734–750, 2018.

(Li et al., 2017) Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Light-head r-cnn: In defense of two-stage object detector. arXiv preprint arXiv:1711.07264, 2017.

(Li et al., 2018) Zeming Li, Chao Peng, Gang Yu, Xiangyu Zhang, Yangdong Deng, and Jian Sun. Detnet: A backbone network for object detection. arXiv preprint arXiv:1804.06215, 2018.

(Lin et al., 2014) 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.

(Lin et al., 2017a) Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.

(Lin et al., 2017b) Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017.

(Liu et al., 2016) 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.

(Maninis et al., 2018) K.K. Maninis, S. Caelles, J. Pont-Tuset, and L. Van Gool. Deep extreme cut: From extreme points to object segmentation. In Computer Vision and Pattern Recognition (CVPR), 2018.

(Moranduzzo and Melgani, 2014) Thomas Moranduzzo and Farid Melgani. Detecting cars in uav images with a catalog-based approach. IEEE Transactions on Geoscience and Remote Sensing, 52(10):6356–6367, 2014.

(Pedoeem and Huang, 2018) Jonathan Pedoeem and Rachel Huang. Yolo-lite: a real-time object detection algorithm optimized for non-gpu computers. arXiv preprint arXiv:1811.05588, 2018.

(Redmon and Farhadi, 2017) Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7263–7271, 2017.

(Redmon et al., 2016) Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

(Tychsen-Smith and Petersson, 2017) Lachlan Tychsen-Smith and Lars Petersson. Denet: Scalable real-time object detection with directed sparse sampling. In Proceedings of the IEEE International Conference on Computer Vision, pages 428–436, 2017.

(Unel et al., 2018) F Ozge Unel, Burak O Ozkalayci, and Cevahir Cigla. The power of tiling for small object detection.

(Wang et al., 2017) Xinggang Wang, Kaibing Chen, Zilong Huang, Cong Yao, and Wenyu Liu. Point linking network for object detection. arXiv preprint arXiv:1706.03646, 2017.

(Wang et al., 2019) Tiancai Wang, Rao Muhammad Anwer, Himash Cholakkal, Fahad Shabbaz Khan, Yanwei Pang, and Ling Shao. Learning rich features at high-speed for single-shot object detection. In Proceedings of the IEEE International Conference on Computer Vision, pages 1971–1980, 2019.

(Wu et al., 2019) Zhenyu Wu, Karthik Suresh, Priya Narayanan, Hongyu Xu, Heesung Kwon, and Zhangyang Wang. Delving into robust object detection from unmanned aerial vehicles: A deep finescale disentanglement approach. In The IEEE International Conference on Computer Vision (ICCV), October 2019.

(Yang et al., 2019) Fan Yang, Heng Fan, Peng Chu, Erik Blasch, and Haibin Ling. Clustered object detection in aerial images. arXiv preprint arXiv:1904.08008, 2019.

(Zhang et al., 2018) Shifeng Zhang, Longyin Wen, Xiaobin, Zhen Lei, and Stan Z Li. Single-shot refinement neural network for object detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4203–4212, 2018.

(Zhang et al., 2019) Junyi Zhang, Junying Huang, Xuankun Chen, and Dongyu Zhang. How to fully exploit the abilities of aerial image detectors. In Proceedings of the IEEE International Conference on Computer Vision Workshops, pages 0–0, 2019.

(Zhou et al., 2019) Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. arXiv preprint arXiv:1904.07850, 2019.

(Zhu et al., 2018) Pengfei Zhu, Longyin Wen, Xiao Bian, Ling Hainin, and Qinghua Hu. Vision meets drones: A challenge. arXiv preprint arXiv:1904.07437, 2018.