Dense and Small Object Detection in UAV Vision based on Cascade Network

Xindi Zhang, Ebroul Izquierdo
Multimedia and Vision Research Group,
School of Electronic Engineering and Computer Science,
Queen Mary University of London
Mile End Road, London, UK
{xindi.zhang, ebroul.izquierdo}@qmul.ac.uk

Krishna Chandramouli
Venaka Media Limited,
393, Roman Road, E3 5QS,
London, UK
k.chandramouli@venaka.co.uk

Abstract

With the development of Unmanned Aerial Vehicles, drones are being deployed in a number of commercial and civil government applications ranging from remote surveillance and infrastructure maintenance among others. However, processing the videos captured by drones for the extracting meaningful information is hindered by multitude of challenges that include, the appearance of small objects, changes in viewpoint of these objects, illumination changes, large-scale resolution of the captured video, occlusion and truncation. Addressing these challenges, there is a critical need to develop algorithms that is able to efficiently process the videos that can result in robust detection and recognition of small objects. In this paper, we propose a novel processing pipeline, that brings together several key contributions including (i) the introduction of DeForm convolution layers within backbone; (ii) use of the interleaved cascade architecture; (iii) data augmentation process based on crop functionality and (iv) multi-model fusion of sub-category detection networks. The proposed approach has been exhaustively benchmarked against VisDrone-DET object detection dataset, which includes 10,209 images for training, validation and testing. The evaluation of the proposed approach has resulted in 22.61 average precision on the test-challenge set in VisDrone-DET 2019.

1. Introduction

The evolution of aerial technology has seen exponential growth especially for Unmanned Aerial Vehicles (UAV) which have found applications beyond military use and have become powerful business tools according to Goldman Sachs report[1]. However, processing the videos captured by drones for the extracting meaningful information is hindered by multitude of challenges that include, the appearance of small objects, changes in viewpoint of these objects, illumination changes, large-scale resolution of the captured video, occlusion and truncation. An example of these limitations/constriants are presented in Figure 1. While, there already exist several approaches successfully reported in the literatures [21] for addressing the challenge of object detection upon dataset captured from traditional datasets, such as COCO [20], PASCAL [7] and ImageNet [6], these approaches have been to result in lower performance when applied for detecting objects on videos or images from drones [32].

In the context of the research presented in the paper, several articles have been considered from the literature addressing the challenge of small object detection. The reported approches has been broadly categorised into data augmentation techniques and deep-learning network architectures. In [13], the authors present an approach of replicating the appearance of small objects at scale for multiple times. The increased volume of the small object dataset is then subsequently used for training the deep-learning network, which is trained for processing traditional dataset. In contrast, the approach presented in TridentNet [15] and SNIP [28] used dilated convolution network layer and scale normalization respectively. The outcome of these approaches aimed at addressing the uneven distribution of small objects in comparison to the appearance of tradition object sizes. On the other hand, addressing the challenge of identifying dense objects as captured from drones, [32] proposed to add anchor or proposal to contain more objects. In addition, the authors also address the topic of category imbalance through the removal of annotated labels for classes containing large volume of training data.

Despite these techniques, the problem of identifying and categorising the small objects remains an open challenge. Addressing the problem of accurately and robustly categorising the small objects captured from drones, the paper proposes a novel processing pipeline, which integrates

[1] http://www.goldmansachs.com/our-thinking/technology-driving-innovation/drones/

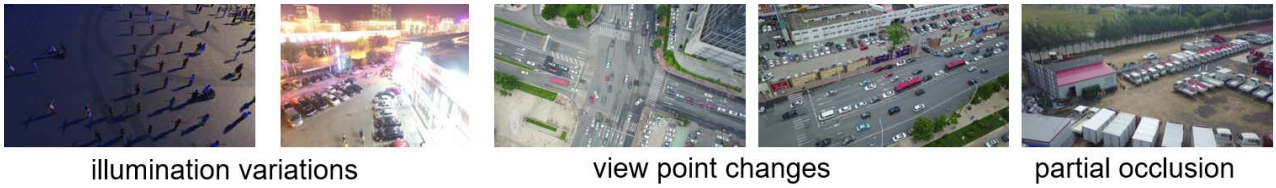

Figure 1. **The challenges in the UAV vision.** Objects are small and densely distributed with partial occlusion and illumination variations. The viewpoint changed due to the different height of drones and camera directions.

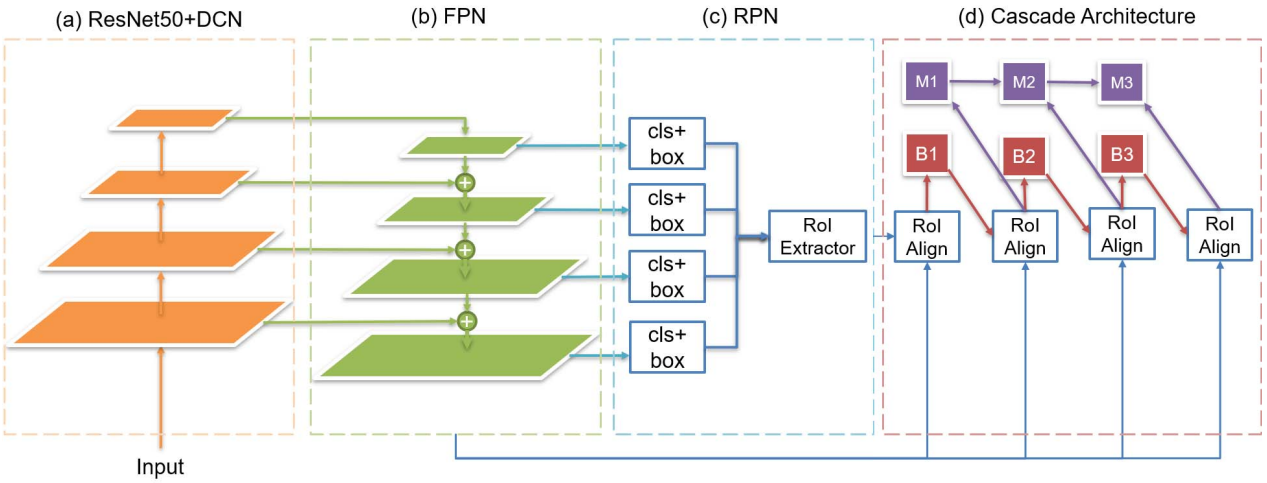

Figure 2. **The whole network structure.** An input image will be input to the backbone (a) ResNet50, which is implemented with deformable convolution. The feature maps further refine with (b) Feature Pyramid Network. Then (c) Region Proposal Network extract some Region of Interest(RoI). The RoI and feature maps input to (d) Cascade Architecture to refine the bounding box prediction with three stages interleaved box head Bi and mask head Mi.

four processing components and the overall architecture of the proposed solution is presented in Figure 2. The framework integrates the ResNet network as proposed in [10] for feature extraction, which is complemented with deformable convolution layer (DCN) as reported in [5]. In addition, the architecture also interfaces with Feature Pyramid Network (FPN) [18] for effectively combining the features at different scales. Subsequently, the architecture integrates the Region Proposal Network (RPN)[26] for the extraction of Region of Interest (RoI). Finally, the interleaved cascade architecture is used to predict box and mask for the candidate region. The box branch and mask branch are interleaved and reciprocal to each other as presented in [3]. Mask branch is for instance segmentation, which generates a pixel-wise mask of the object. However, the training of segmentation networks requires more precise labelled data sets, and not easy to transfer to the problem with low-cost labelling. We also use some learning strategies, like OHEM [27], soft-NMS [1] and warmup learning rate. A detailed outline of the various processing steps is presented in Section 3. The main contributions of this paper are as follows:

- To validate the addition of a deformable convolution layer in the last three stages of ResNet to learn more distinguishable feature representations.
- To process the modified features through the box and mask interleaved cascade architecture for predicting and refining the position and size of the detected object.
- To propose a data augmentation process prior to training and testing for improved performance of the network.
- To implement a parallel process of architecture that is able to fuse the outputs of two network models, each trained on a sub categories of classes.

2. Related Work

In this section, a summary of the related work is presented within the scope of the research work presented broadly categorised into general object detection strategies and small object detection approaches.

2.1. General Object Detection

Object detection algorithms reported in the literature can be divided into two categories: single-stage and two-stage. The single-stage detector directly predict the location

of objects without extracting proposal, such as SSD [23], YOLO [25], RetinaNet [19]. The two-stage detector generates a set of region proposal and then predict the object class inside the region of interest (RoI) as well as refine the proposals to according define the position and size of the object. This approach has been adopted, such as Fast R-CNN [8], Faster R-CNN [26], RFCN [4]. Normally, the single-stage approach is faster than two-stage, while the two-stage method has higher accuracy than single-stage. With the improvement over the past few years, the single-stage approach as presented RefineDet [30] also make progress and outperform the performance of the two-stage algorithm. However, those algorithms are designed for general object detection, which is not good at detecting on the small and densely distributed object.

2.2. Small Object Detection

However, addressing the challenge of detecting small objects, which are inherently present in the well-known COCO dataset, several approaches have been reported. To address the challenge of class imbalance due to the sparse appearance of the small objects in the dataset, the use of data augmentation techniques has reported in [13] by copying and pasting to increase the number of small objects. For algorithm design, a lot of methods are based on multi-scale image pyramid [11, 22, 5] to improve the performance of small and large object scales. SNIP [28] used scale normalization for image pyramids with mutli-scale training. TridentNet [15] applied dilated convolutional layers [29] with different dilation rate to solve the scale variation. But these kinds of methods aim to address the problem of scale variations. In general dataset, small objects are few in number. These methods can improve the performance of small object detection as compared to that of the medium and large object.

In contrast to the approaches reported in the literature, the research presented in the paper implements an object detection framework for object idenfication on the UAV captured images that integrates the interleaved cascade architecture with deformable convolution layer. In addition, the proposed approach also includes data augmentation process for improving the robustness of the trained network.

3. Method

The proposed network architecture is presented in Figure 2, which is divided into four parts. The first part serves as a backbone, used to extract feature maps from the input image. Subsequently the framework integrates, the ResNet50 network with deformable convolutional layers, which is described in Section 3.1. The second processing component aims to exploit and refine the feature maps obtained from ResNet50 through the use of FPN. The third component includes the region proposal network (RPN) to extract potential proposals of objects contained in the image. The final component is a task head for specific targets. The component uses an interleaved cascade architecture to assign bounding box and mask prediction. This part can be found in Section 3.2.

In order to address the challenge of detection small and dense object, image cropping function is used as a data augmentation process during the training stage (in Section 3.3). Then all the cropped images and the original image resized to the input specification of the network. Also, multi-model fusion is used to solve the imbalance categories distribution (in Section 3.4). The test time augmentation helps the accuracy improve further (in Section 3.5).

3.1. Deformable Convolution

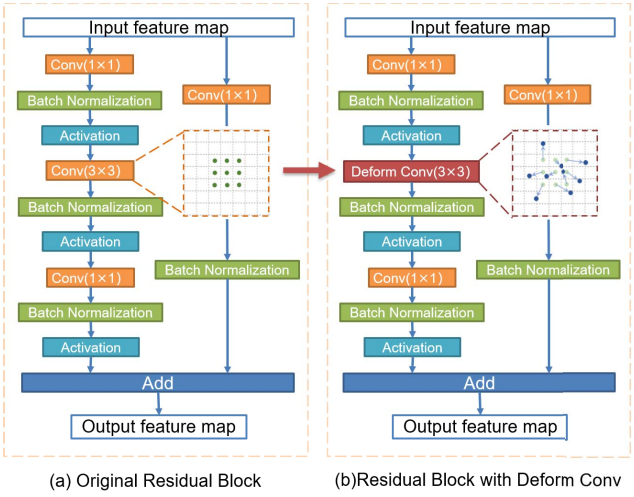

Figure 3. **Residual Block with Deformation Convolution.** The (a) Original Residual Block uses regular convolutional layers. In (b) the second convolution layer on the left branch changes to Deform Conv. The Deform Conv block contains an additional convolutional layer to learn offset and combined with the input feature map to a deformable convolutional layer.

The deformable convolution [5] layer is used in the last three-stage (res3, res4, res5) of the backbone. The traditional convolution network has limited performance on geometric transformation due to the restricted form of convolutional layers and pooling layers. The traditional network architecture is not able to transfer well on the drone based object detection tasks. The image used for training and for testing in real applications cannot be perfectly consistent in distribution and scale viewpoint. Viewpoint variation is one of the biggest challenges in images captured from drones, since the dataset distribution contains images captured in top view angle, while other images might be captured from a lower view angle. The features learned from the object in different angle is not transferable. In order to improve the

transferability of the learned features, the deformable convolutional layer has been adopted within ResNet50 for feature extraction, since deformable convolution can change the reception field dynamically. The proposed change to the ResNet50 leads to semantic representation of object features and localisation at higher layers.

The traditional convolution progress can be represented as below:

$$R = \{(-1,-1),(-1,0),...,(0,1),(1,0)\} \tag{1}$$

$$y(P_0) = \sum_{P_n \in R} (w(P_n) \cdot x(P_0 + P_n)) \tag{2}$$

R is the grid of a 3×3 convolution layer. P_0 is the coordinate of feature map y. P_n is the coordinate in R. So $P_0 + P_n$ is the sample point coordinate, $x(P_0 + P_n)$ is the corresponding pixel value in the input feature map x. $w(P_n)$ is the weights in convolution kernel.

The output of deformable convolutiona layer is:

$$y(P_0) = \sum_{P_n \in R} (w(P_n) \cdot x(P_0 + P_n + \Delta P_n)) \tag{3}$$

$$\{\Delta p_n | n = 1, ..., N\}, N = |R|. \tag{4}$$

The offset is added to R, so the sample point become $P_0 + P_n + \Delta P_n$. The ΔP_n is fractional. So the sample point in the input feature map will locate in a fractional position without pixel value. To get the pixel value of $x(P_0 + P_n + \Delta P_n)$, it needs to operate bilinear interpolation. The offset is learned by additional convolution.

In each of the residual block in stage Res3, Res4 and Res5, the second simple convolution are changed to deformable convolution. As shown in Figure 3, original convolution kernel is a regular rectangle. In the deformable convolution kernel, offset are added to each sample point and the arrangement becomes irregular. There is an additional convolution layer to learn offset. Then the input feature map is combined with offset together input to the deformable convolution to do offset and convolution.

3.2. Interleaved Cascade

The task head for box and mask prediction at the end of the whole network is an interleaved cascade architecture, which is derived from the intermediate form of Hybrid Task Cascade (HTC) [3]. Cascade is an architecture, which can improve the performance of multiple tasks through multi-stage refinement. Rather than executing object detection and object segmentation in parallel, the two tasks are interleaved and reciprocal with each other in this framework.

The mask head is designed for instance segmentation. Instance segmentation is more complex than object detection. It needs to classify every pixel in the picture. The mask and the box prediction are influenced by each other in this architecture. So using this pixel-wise algorithm is more precise on dense and small objects than traditional object detection algorithms. However, segmentation requires more accurate labelling, which is time-consuming and laborious. Therefore, it is difficult to transfer for other low-labelling problems. The object labelling considers all the pixels in the ground truth bounding box are labelled as the mask of the object. This saves the cost of labelling, and can still help improve the object detection performance.

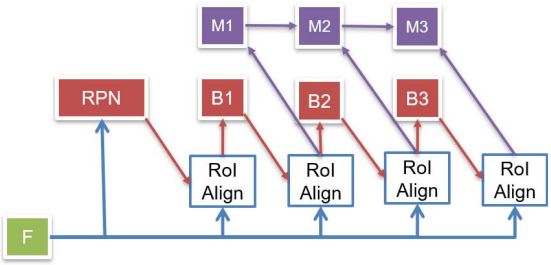

Figure 4. **The Interleaved Cascade architecture.** The feature map is input into each stage. And use RoI Align as bounding box RoI extractor. B_t is the box head to predict the box location and the corresponding class. M_t is the mask head to predict the mask at t-th stage.

The architecture used in this paper is shown as Figure 4. In each stage of the cascade, the bounding box prediction comes from:

$$x_t^{box} = P(x, r_{t-1}), r_t = B_t(x_t^{box}) \tag{5}$$

In this formula, x represents the features extracted by the backbone network. After pooling operated by $P(\cdot)$, the box features x_t^{box} are derived. B_t is the box head at t-th stage. The box prediction r_t is generated from box features by box head. In every t-th stage, The box features are considered both features from backbone and box features from the previous stage. Therefore, the bounding box prediction and be refined and improved in every stage.

$$x_t^{mask} = P(x, r_t), m_t = M_t(F(x_t^{mask}, m_{t-1}^-)) \tag{6}$$

Mask branch interleaves with box branches so that it can be benefited by the updated box predictions. x_t^{mask} is mask features, which is determined by backbone features and box features of the current stage. M_t is mask head at t-th stage. This head not only considers mask features of the current stage but also take previous stage intermediate mask features m_{t-1}^- into account. This connection between different stages improves the mask prediction further, instead of relying on box refinement only. F is the function to combine two features, which can be shown as below:

$$F(x_t^{mask}, m_{t-1}) = x_t^{mask} + g_t(m_{t-1}^-) \tag{7}$$

m_{t-1}^{-} represents intermediate mask feature, which is the RoI feature before the deconvolutional layer. g_t is a $1 * 1$ convolutional layer for embedding the feature to be aligned with x_t^{mask}.

The loss function considers multi-task together.

$$L = \sum_{t-1}^{T} (L_{bbox}^t + L_{mask}^t), \tag{8}$$

$$L_{bbox}^t(c_i, r_t, \hat{c}_t, \hat{r}_t) = L_{cls}(c_t, \hat{c}_t) + L_{reg}(r_t, \hat{r}_t), \tag{9}$$

$$L_{mask}^t(m_t, \hat{m}_t) = BCE(m_t, \hat{m}_t). \tag{10}$$

L_{bbox}^t is for bounding box prediction. It combines classification loss L_{cls} and bounding box regression loss L_{reg} together with the same definition of Cascade R-CNN [2]. L_{mask}^t regards mask prediction. BCE stands for binary cross entropy used in Mask R-CNN [9]. The overall loss L is the combination of L_{bbox}^t and L_{mask}^t at each stage t.

3.3. Data Augmentation

For data augmentation, image cropping function is used to enhance the quality of the training dataset. The image is segmented into 4x4 blocks on average and merged into the training set with the original images. The training set is increased 5x as much as the original one.

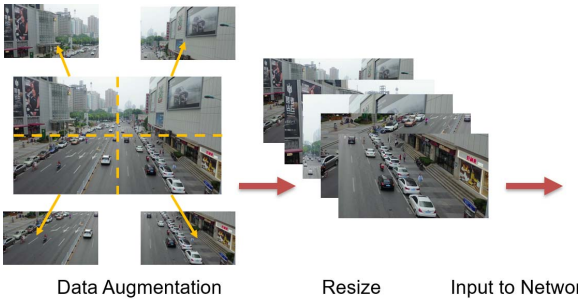

Figure 5. As the data augmentation method, we crop the image into four pieces averagely, and incorporate in the training set. All the images in training set are resized to the network input size.

The proposed data augmentation technique has several advantages, including (i) deep learning algorithm often requires large-volume of training data for improved representation of trained featuers; (ii) objects are often occluded and truncate thus the network needs to learn the features of a partial object and predict the whole through the part. Cropping increases the proportion of truncate objects and enables the network to learn partial features better; and (iii) training the small cropped image as a whole does increase the scale variation of the object, and learn more details and characteristics of the small object.

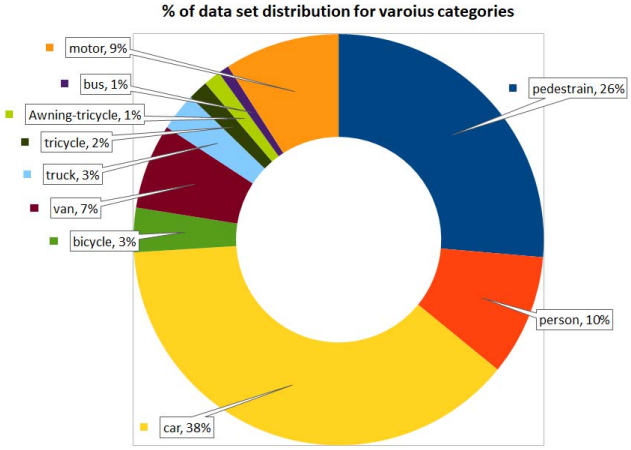

Figure 6. The proportion of objects of different categories.

3.4. Multi-Model Fusion

One of the key research problems to be addressed in training the network relates to the establishing a balance between the difference datasets. As the deep-learning networks require large-quantities of both positive and negative samples, the unbalanced distribution of annotated classes might lead to under performance of trained network. In the context of the research presented in the paper, Figure 6, presents an overview of various classes upon which the proposed network has been trained (additional details on the dataset is presented in Section 4.1).

In order to resolve the unbalanced distribution of object categories, two sub-category models have been proposed which includes a sub-set of data objects for training and testing purposes. The information extracted from the both sub-category network models are subsequently used to fuse the outcomes and validate the training. The overview of the proposed approach for multi-model solution is presented in Figure 7. The top-3 classes in first network relates to pedestrian, person and car. The rest of the classes namely bicycle,van,truck, tricycle, awning-tricycle, bus and motor, are used to train a second model. The separation of the sub-categories of image dataset has been identified empirically. The advantage of training two sub-category of networks is that, each network is trained on a balanced distribution of dataset resulting in well modelled features for each of the category. In addition, the training input to each network is also extended by complementary datasets considered as negative samples, resulting in two network models that are able to effectively and efficiently learn the underlying complex feature patterns. The output of both sub-category network are cumulatively added towards the final outcome assigned for each object identified and localised in the input image.

Figure 7. **Multi-Model fusion.** We use one network to train pedestrian, person and car, which have a big amount in the dataset. Use another network to train bicycle, van, truck, tricycle, awning-tricycle, bus and motor, which numbers are sparse. Fuse two models at the testing time can address the problem of unbalanced category.

Method	AP[%]	AP50[%]	AP75[%]	AR1[%]	AR10[%]	AR100[%]	AR500[%]
CornerNet [14]	17.41	34.12	15.78	0.39	3.32	24.37	26.11
Light-RCNN ResNet-101 [16]	16.53	32.78	15.13	0.35	3.16	23.09	25.07
DetNet 59 [17]	15.26	29.23	14.34	0.26	2.57	20.87	22.28
RefineDet512-VGG16 [30]	14.9	28.76	14.08	0.24	2.41	18.13	25.69
retinanet [19]	11.81	21.37	11.62	0.21	1.21	5.31	19.29
fpn [18]	16.51	32.2	14.91	0.33	3.03	20.72	24.93
cascadercnn [2]	16.09	16.09	15.01	0.28	2.79	21.37	28.43
Ours	**22.61**	**45.16**	**19.94**	**0.42**	**2.84**	**17.1**	**35.27**

Table 1. Results on test-challenge set. Compare with the official baselines, ours is better.

3.5. Test Time Augmentation

Data augmentation is to expand the training set to help the network adapt with more situation, while test time augmentation (TTA) is to do data augmentation at the testing stage. TTA usually make random modifications like the method in data augmentation, such as rotation, shift, flip and translation. Then test the trained model on all the modified images and average the prediction to gain the final result. The benefit of doing TTA can be shown as follow. First, test on only one image may occur error which will reduce the accuracy. TTA on several images, modified from the original one, can help mitigate the error. Second, using the same augmentation within the training phrase can help the network adapt to change. It is better when the testing set has same characteristic with training set, such as same lighting conditions, similar scale variations and class distribution.

In this paper, we use image cropping and Non-Maximum Suppression (NMS) as test time augmentation. During the training stage, the images are cropped averagely into four pieces. But cropping the same during the test time will cause problems. The object on the cropping edge only gets half bounding box. To avoid this problem, the cropping weight and height are still half of the original images, while the strides become one-quarter of corresponding side length. In that case, the image is cropped into nine pieces overlap with each other. Then mapping the bounding box result back to the original image. After that, using NMS to remove the overlapping boxes. This method helps to improve performance, especially for small object detection. Because crop and resize help the algorithm zoom in and gain more features on small objects.

4. Experiments

The section presents an outline of the performance evaluation carried out along with a detailed outline of the dataset against which the proposed architecture has been benchmarked. In addition, various evaluation metrics has also been included for completeness.

4.1. Datasets and evaluation metrics

Datasets. VisDrone-Det [31] is an object detection dataset with drone perspective. Most of the objects in this dataset are small, densely distributed and partially occluded. The viewpoint changes with the different flying height of drone and camera direction. There are also illumination and perspective changes in different scenarios. The object categories in this dataset can be regarded as two major categories: human beings and means of transportation. In this dataset, human beings are divided into person and pedestrian. Pedestrian are those human with standing or walking pose and person with other poses. Means of transportation contain car, van, bus, truck, motor, bicycle, awning-tricycle, and tricycle. There are 6471 images in the training set, 548 in the validation set and 1580 in the test-challenge set.

Evaluation Metrics. Similar to the evaluation protocol in MS COCO [20], we use AP, APIOU=0.50, APIOU=0.75, ARmax=1, ARmax=10, ARmax=100, and ARmax=500 metrics to evaluate the results of detection algorithms. Unless otherwise specified, the AP and AR

Method	pedestrain	person	bicycle	car	van	truck	tricycle	awning-tricycle	bus	motor
baseline	21.91	12.74	10.19	62.89	32.1	24.28	17.58	5.46	42.19	23.94
baseline + TTA	33.7	17.61	6.72	65.61	21.41	13.22	11.7	2.71	15.82	22.97
DA + fusion	28.1	20.83	17.98	67.63	40.01	34.57	25.48	11.33	49.69	32.68
DA + fusion + TTA	**49**	**40.33**	**31.28**	**75.3**	**44.65**	**36.23**	**32.11**	**12.63**	**53.41**	**51.78**

Table 2. The result of each class with different implementation in the validation set. The baseline is a pure network. Test Time Augmentation (TTA) crop the image at the testing time. DA stands for data augmentation, which operates image cropping at the training phrase. Multi-Model fusion combines two networks trained with a different group of classes. Each module helps to improve performance.

metrics are averaged over multiple intersection over union (IoU) values. Specifically, we use ten IoU thresholds of [0.50:0.05:0.95]. All metrics are computed allowing for at most 500 top-scoring detections per image (across all categories). These criteria penalize missing detection of objects as well as duplicate detections (two detection results for the same object instance). The AP metric is used as the primary metric for ranking the algorithms.

4.2. Implementation details

The algorithm is implemented with PyTorch [24] and mmdetection [12]. The GPU used for training is Nvidia GeForce 1060 6GB.

The backbone is used to extract feature maps. We use ResNet50 as the backbone in this experiment. The residual block in the last 3 stages of ResNet (res3, res4, res5) was changed. Some regular convolutional layers were changed to deformable convolutional layers. Head is for specific tasks, e.g. bounding box prediction and mask prediction. We use the 3-stage interleaved cascade architecture to refine the box and mask prediction. The specific structure is described in the previous section.

The images cropped into four pieces as data augmentation. As for test time augmentation, each image is cropped into nine pieces. The final results were fused by NMS. For model fusion, we trained two different models. One is training for pedestrian, person and car. Another one is training for the rest of the class. Then combine the result at testing time.

At the training stage, the mask information used is all the pixels in the ground truth box, which is a rectangle mask.

The experimental output shows that the network does not automatically learn the foreground and background information of the target in the bounding box. In the inference phase, the output mask is also a rectangle. But compared with other object detection algorithms, this method can improve the effect, which is better than all official baseline as shown in Table 1.

4.3. Ablation Experiments

The ablation experiments result in the validation set is shown in Table 3. The first line is the baseline performance, which represents interleaved cascade with resnet50 FPN trained in 20 epoch. There is no pre-processing and post-processing. In the second line, we add deformable convolution layers and the performance improved. As described in method, we use image cropping as data augmentation. In the third line, every image in the training set is cropped into four pieces and combine with the original image to expand five times than before. After training with the augmented dataset, the performance increased. For model fusion, the 10 categories in Visdrone are divided into two groups. One group contain person, pedestrian and car, the other group containing the rest of the categories. We use two different networks to train those two group and combine the result. The result is shown in the fourth line. The fifth line represents the output of adding test time augmentation. The test image is cropped into nine pieces. The width and height of the cropped image are half of the width and height of the original image. The stride is one-quarter of the corresponding edge. Then test on the nine cropped images and the original one. Use NMS to remove duplicates. Each module helps to improve the accuracy and performance of small object detection. We use all the module together on test-challenge set and the result exceeds all the baseline algorithm from the challenge officials.

Method	AP[%]	AP50[%]	AP75[%]
Interleaved Cascade	25.8	40.8	27.9
+ dconv c3-c5	26.8	42.1	28.8
+ data augmentation	28.8	47.1	29.3
+ model fusion	29.93	50.37	**30.61**
+ test time augmentation	**30.12**	**58.02**	27.53

Table 3. **The ablation experiments on the validation set.** The first line is the result of cascade architecture with ResNet50. Then we add deformable convolution layers in the last three stages of ResNet and the performance improved, shown in the second line. Then we add data augmentation, multi-model fusion and test time augmentation, the accuracy grows higher respectively.

4.4. The result in each class

Table 2 shows the influence on each class by using prepossessing and post-processing methods. The number in the table is the mAP scores of the corresponding class. The

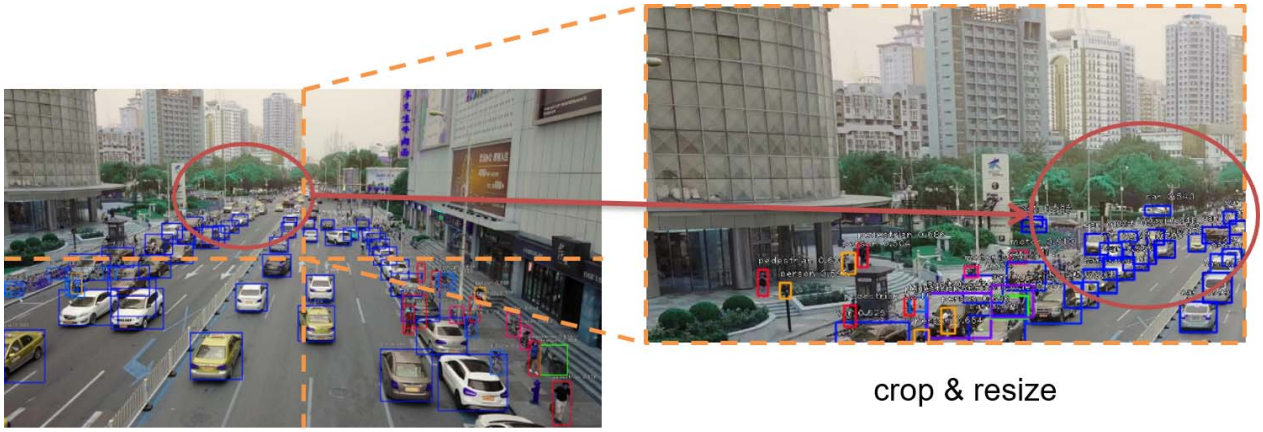

Figure 8. The result on cropped image. The left-hand side is the result of the original image. Some objects are too small to be detected, as the red circle region. On the right is the testing result on the cropped image, the object in the red circle becomes detectable.

first line is the result of a single network without bells and whistles.

Test Time Augmentation. Then we use image cropping as post-processing at test time. The mAP scores of pedestrian, person and car are increased, while others are decreased. The performance increased because some objects, that are too small to be detected, become detectable after cropping and resizing. It decreases because of the scale of objects changed. After crop and resize, object become larger than before, which is easy to detect. But the new scale of objects is not the same as the training set. So the trained network is not adapt to them. Pedestrian, person and car have a big amount of samples in the dataset with various size, so the network learns enough feature to represent them. However, other classes don't have enough samples to train, so the performance of them decreased.

Multi-Model Fusion. The unbalanced distribution influenced a lot. Train all the classes together cannot learn enough feature for every class. So we decide to train pedestrian, person and car with one network, the rest of the classes with another network. Then the results of two models at the test time are combined.

Data Augmentation. To increase the scale variations, the image cropping was used to do data augmentation as prepossessing. The result can be seen in the third line of Table 2. The performance of every class improved to a large extent.

At last, we use data augmentation, test time augmentation and model fusion together to get the best result. Data augmentation can help the network learn more features with different scales. Model fusion can handle the class imbalance. Test time augmentation can zoom in to small part of the image and detect the tiny object.

5. Conclusion

Object detection in UAV vision is extremely difficult due to small objects, densely distributed, viewpoint changes, illumination variations and partial occlusion. The dataset VisDrone-DET also have problem with category imbalance. The deformable convolution improves the adaptability of viewpoint variations due to geometric transformation learning. The interleaved cascade architecture resolve the detection of dense and occlusion objects through refining the bounding box prediction in three stages. The data augmentation and test time augmentation greatly improve performance, especially for small objects. Multi-model fusion address the problem of unbalanced categories. The overall network exceeds all the official baseline in VisDrone-DET 2019 challenge.

In future work, we will focus on network efficiency and operation speed. The network pruning can be implemented to achieve a light-weight algorithm, which can be operated in the real UAV platform.

Acknowledgements. The research activity leading to the publication has been partially funded by the European Union Horizon 2020 research and innovation program under grant agreement No. 787123 (PERSONA RIA project).

References

[1] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis. Improving object detection with one line of code. *CoRR*, abs/1704.04503, 2017.

[2] Z. Cai and N. Vasconcelos. Cascade R-CNN: delving into high quality object detection. *CoRR*, abs/1712.00726, 2017.

[3] K. Chen, J. Pang, J. Wang, Y. Xiong, X. Li, S. Sun, W. Feng, Z. Liu, J. Shi, W. Ouyang, C. C. Loy, and D. Lin. Hybrid task cascade for instance segmentation. *CoRR*, abs/1901.07518, 2019.

[4] J. Dai, Y. Li, K. He, and J. Sun. R-FCN: object detection via region-based fully convolutional networks. *CoRR*, abs/1605.06409, 2016.

[5] J. Dai, H. Qi, Y. Xiong, Y. Li, G. Zhang, H. Hu, and Y. Wei. Deformable convolutional networks. *CoRR*, abs/1703.06211, 2017.

[6] J. Deng, W. Dong, R. Socher, L. jia Li, K. Li, and L. Fei-fei. Imagenet: A large-scale hierarchical image database. In *In CVPR*, 2009.

[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *International Journal of Computer Vision*, 88(2):303–338, June 2010.

[8] R. B. Girshick. Fast R-CNN. *CoRR*, abs/1504.08083, 2015.

[9] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick. Mask R-CNN. *CoRR*, abs/1703.06870, 2017.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[11] J. Huang, V. Rathod, C. Sun, M. Zhu, A. Korattikara, A. Fathi, I. Fischer, Z. Wojna, Y. Song, S. Guadarrama, and K. Murphy. Speed/accuracy trade-offs for modern convolutional object detectors. *CoRR*, abs/1611.10012, 2016.

[12] J. P. Y. C. Y. X. X. L. S. S. W. F. Z. L. J. X. Z. Z. D. C. C. Z. T. C. Q. Z. B. L. X. L. R. Z. Y. W. J. D. J. W. J. S. W. O. C. C. L. D. L. Kai Chen, Jiaqi Wang. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[13] M. Kisantal, Z. Wojna, J. Murawski, J. Naruniec, and K. Cho. Augmentation for small object detection. *CoRR*, abs/1902.07296, 2019.

[14] H. Law and J. Deng. Cornernet: Detecting objects as paired keypoints. *CoRR*, abs/1808.01244, 2018.

[15] Y. Li, Y. Chen, N. Wang, and Z. Zhang. Scale-aware trident networks for object detection. *CoRR*, abs/1901.01892, 2019.

[16] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Lighthead R-CNN: in defense of two-stage object detector. *CoRR*, abs/1711.07264, 2017.

[17] Z. Li, C. Peng, G. Yu, X. Zhang, Y. Deng, and J. Sun. Detnet: A backbone network for object detection. *CoRR*, abs/1804.06215, 2018.

[18] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie. Feature pyramid networks for object detection. *CoRR*, abs/1612.03144, 2016.

[19] T. Lin, P. Goyal, R. B. Girshick, K. He, and P. Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017.

[20] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[21] L. Liu, W. Ouyang, X. Wang, P. W. Fieguth, J. Chen, X. Liu, and M. Pietikäinen. Deep learning for generic object detection: A survey. *CoRR*, abs/1809.02165, 2018.

[22] S. Liu, L. Qi, H. Qin, J. Shi, and J. Jia. Path aggregation network for instance segmentation. *CoRR*, abs/1803.01534, 2018.

[23] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. E. Reed, C. Fu, and A. C. Berg. SSD: single shot multibox detector. *CoRR*, abs/1512.02325, 2015.

[24] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer. Automatic differentiation in PyTorch. In *NIPS Autodiff Workshop*, 2017.

[25] J. Redmon, S. K. Divvala, R. B. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. *CoRR*, abs/1506.02640, 2015.

[26] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. *CoRR*, abs/1506.01497, 2015.

[27] A. Shrivastava, A. Gupta, and R. B. Girshick. Training region-based object detectors with online hard example mining. *CoRR*, abs/1604.03540, 2016.

[28] B. Singh and L. S. Davis. An analysis of scale invariance in object detection - SNIP. *CoRR*, abs/1711.08189, 2017.

[29] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. 11 2015.

[30] S. Zhang, L. Wen, X. Bian, Z. Lei, and S. Z. Li. Single-shot refinement neural network for object detection. *CoRR*, abs/1711.06897, 2017.

[31] P. Zhu, L. Wen, X. Bian, L. Haibin, and Q. Hu. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018.

[32] P. Zhu, L. Wen, D. Du, X. Bian, H. Ling, Q. Hu, Q. Nie, H. Cheng, C. Liu, X. Liu, et al. Visdrone-det2018: The vision meets drone object detection in image challenge results. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018.