Patch-level Augmentation for Object Detection in Aerial Images

Sungeun Hong*
SK T-Brain
csehong@sktbrain.com

Sungil Kang*
SK T-Brain
sung1.kang@sktbrain.com

Donghyeon Cho
SK T-Brain
cdh12242@sktbrain.com

Abstract

Object detection in specific views (e.g., top view, road view, and aerial view) suffers from a lack of dataset, which causes class imbalance and difficulties of covering hard examples. In order to handle these issues, we propose a hard chip mining method that makes the ratio of each class balanced and generates hard examples that are efficient for model training. First, we generate multi-scale chips to train object detector. Next, we extract object patches from the dataset to construct an object pool; then those patches are used to augment the dataset. By this augmentation, we can overcome the class imbalance problem. After that, we perform inference with the trained detector on augmented images, then generate hard chips from misclassified regions. Finally, we train the final detector by both normal and hard chips. The proposed method achieves superior results on VisDrone dataset both qualitatively and quantitatively. Also, our model is ranked 3rd in VisDrone-DET2019 challenge (http://aiskyeye.com/).

1. Introduction

It is one of the challenging problems to cover a wide range of object sizes in object detection. In particular, the problem becomes more severe when the objects to be detected are very small compared to the input image, such as aerial images captured from drones or surface images for defect detection. To alleviate this issue, early works train models over a range of scales or use independent predictions at layers of different resolutions [10, 14]. Moreover, studies about network architecture dealing with multi-scale objects have been proposed. For instance, feature pyramid network (FPN) [15] uses hierarchical features to cover multi-scale objects while deformable convolution [12, 5] adjusts the receptive fields for objects of varying sizes.

Recently, to handle multi-scale objects in the training stage, a new concept called chip was introduced [24, 19, 13]. The main idea of chip-based training is to train models only with sub-images (i.e., positive chips) where objects are likely to present rather than a whole image. For clarity, in the remainder of this paper, we refer to the chip as a sub-image while the image as a whole image. In chip-based training, meaningful regions are extracted from a whole image by using ground-truth bounding boxes, then resized to an appropriate scale. To reduce the false-positive rate and speed up the training process, Singh *et al.* [24] also uses negative chip mining, which can skip easy background regions.

Despite the recent promising results of chip-based training, there are still several issues when the dataset samples are scarce. In this paper, we tackle two issues in object detection.

Class imbalance: In the literature of class imbalance in object detection, most studies have addressed the issue of imbalance between foreground and background. Traditionally, sampling heuristics or online hard example mining were performed to balance between foreground and background [22]. Recently, focal loss was introduced to reduce the contribution of easy background examples [16]. However, class imbalance within foreground classes has rarely been addressed so far. Unfortunately, foreground-class imbalance causes significant performance degradation, especially when the dataset samples are scarce.

Hard chip mining: Conventional negative chips [24, 19] consist of background regions that are likely to include object instances but do not contain ground-truth instances. Using negative chips enables to reduce the false-positive rate. We argue that performance can be further improved by aggressively using hard examples that confuse models. Here, hard examples can be background regions, existing object instances or synthesized object-like instances.

To address these issues, we propose hard chip-based training, taking into account the imbalance between foreground classes. Our core insight is to leverage object patches, which are misclassified by models trained from

*They are equal contributors to this work.

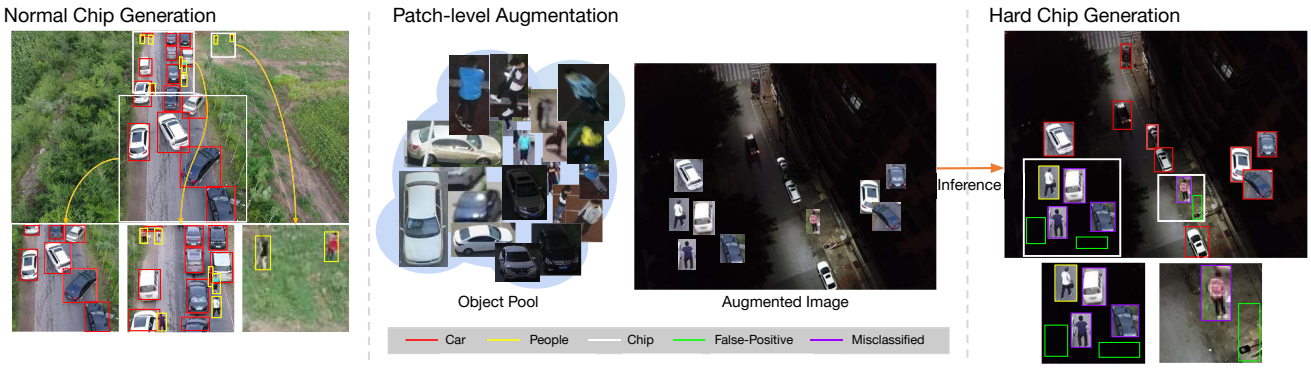

Figure 1. Overview of the proposed method. First, we train our detection model with positive (i.e., normal) chips. We then cut all object instances in the dataset and paste them into the existing randomly selected images considering class imbalance and geometric shape information. We also use an external dataset to reduce the false-positive rate and apply instance masking to resolve heterogeneity between datasets efficiently. Given patch-level augmented images, we make an inference using models trained on normal chips; and then generate hard chips containing misclassified instances. Finally, we retrain our detection model with hard chips as well as normal chips.

normal chips, as hard examples. Model inference is performed on augmented images. For this, we cut object instances from existing images, and paste them to the other images. However, naively cutting and pasting object patches in the existing image set can lead to degrade performance. Therefore, we take into account geometric shapes (e.g., location, scale, orientation) and appearance of transparency when patches are pasted. We further propose a scheme that can efficiently utilize external datasets to compensate for the lack of existing training data. We evaluate our method on VisDrone-DET dataset from VisDrone challenge [27, 28, 25, 29, 6], which enables extensive evaluation and investigation of visual analysis algorithms on the drone platform. With the proposed hard chip mining method, our model is ranked 3rd in VisDrone-DET2019 challenge, which shows promising results of our method quantitatively. Also, we provide inference results of our model on VisDrone-DET2019, which shows the effectiveness of our method qualitatively.

2. Related Work

Although deep learning-based approaches have shown outstanding performance in various research fields [21, 10, 11, 2], there is a constraint that they usually require huge amounts of data to ensure high performance. Data augmentation is one of the effective techniques for increasing both the amount and diversity of data without any supervision. Conventionally, augmentation techniques, such as translation, flipping, and rotation, have been used to improve performance by being applied to the input pipeline of existing models. While various augmentation techniques are widely applied in image classification [8, 1, 3], augmentation techniques have rarely been addressed in object detection. The need for data augmentation in object detection is more crucial because collecting labeled data for object detection is more costly. In considering data augmentation for object detection, patch-level realism in [7] is closely related to our method. While their augmentation method focuses on realistic image synthesis, our patch-level augmentations consider class imbalance and generate hard examples, which lead to improve detection accuracy.

Several attempts have been made to solve class imbalance problem which occurs in object detection. Traditionally, sampling heuristics, such as a fixed foreground-to-background ratio, have been applied to R-CNN-like detectors by a two-stage cascade [21]. Shrivastava et al. [22] propose online hard example mining (OHEM) [22], a simple modification of SGD that focus on hard examples. In their method, training examples are sampled according to a non-uniform, non-stationary distribution that depends on the loss of each example. Recently, focal loss which approaches class imbalance from the perspective of loss [16]. They reshape the standard cross-entropy loss, which enables to down-weights the loss assigned to well-classified examples. Despite numerous efforts for class imbalance between foreground and background regions in object detection, class imbalance within foreground classes has rarely been addressed until now. In this study, we propose a method that can directly handle class imbalance between foreground classes, and demonstrate its effectiveness in our experiments.

Meanwhile, considerable efforts have been devoted to cover a wide range of object sizes in object detection [10, 15, 12, 24]. Among them, SNIPER [24] has shown promising results in recent years, and several subsequent approaches consider chip-based training for efficient object detection [19, 13]. Motivated by these approaches, we train our model with positive chips (i.e., sub-images) where object instances are likely to present. In our approach, instead of using conventional negative chip mining, we propose hard chip mining. While negative chips only focus on

Figure 2. Patch-level augmented images. (top) augmented images from VisDrone-DET dataset (bottom) augmented images from DOTA dataset.

reducing the false-positive rate, our hard chips are designed to not only reduce the false-positive rate but also leverage augmented hard examples. Also, hard examples are augmented considering the number of samples per each class, so foreground-class imbalance can be effectively resolved in our approach.

3. Method

Our objective is to solve the foreground-class imbalance problem in object detection and to exploit hard examples aggressively. To this end, we train our model using both normal chips (i.e., positive chips) and our proposed hard chips at multiple scales. Fig. 1 illustrates the overview of the proposed method.

3.1. Network Architecture

Architecture choice is a very important issue in deep learning-based approaches. Conventional object detection techniques using deep neural networks can be divided into two categories: one-stage detector and two-stage detector. One-stage approaches jointly localize object instances and classify their labels without the proposal extraction stage commonly used in two-stage detectors. Compared to two-stage detectors, the one-stage detectors are faster and simpler but usually show a lower accuracy. As indicated by [28], YOLO [20], SSD [18], and RetinaNet [16] are representative one-stage models. On the other hand, two-stage detectors first generate a pool of object proposals using a separated region proposal generator and then predict accurate object regions and their class labels. Representative two-stage detectors include Faster R-CNN [21], R-FCN [4], Mask R-CNN [9], and FPN [15]; and most studies use a Faster R-CNN model as a baseline in recent years.

According to the report of VisDrone-DET2018 challenge [28], a total of 34 different object detection methods were submitted and they can be categorized into four baseline models: SSD [18], Faster R-CNN [21], R-FCN [4], and FPN [15]. Consistent with the comparison results in the MS COCO dataset [17], FPN achieves the best performance, SSD performs the worst, and R-FCN outperforms Faster R-CNN. Taking into account promising results of FPN in previous studies, our model architecture is based on FPN-based Faster R-CNN in which the backbone is ResNet-101.

3.2. Normal Chip Mining

Since most of the background regions are easy to classify, object detection can be efficiently processed on the multiple regions-of-interest (ROIs) rather than a whole image. Here, we consider the ROIs containing ground truth instances in an image as normal chips. It is known that it is better to ignore gradients of extremely large or small objects for each scale during multi-scale training [23]. To take advantage of this scheme, we construct normal chips on a multi-scale in which there is a desired area range for each scale.

Given an image, $N_i \times N_i (i = 1, 2, \ldots, K)$ local window traverses the whole image with a stride, and counts the number of valid ground-truth instances on each scale. We then select the local window with the largest number of valid ground-truth instances over all scales as a normal chip. We additionally select the window that most encloses the remaining instances and repeat this process until all ground-truth instances in the image are covered by normal chips. Once normal chips are selected considering all scales, they are resized to the same size, with the meaning of size normalization. Training with normal chips uses only a subset of the entire image, and thus, it allows the detection model to be learned effectively with fewer operations. In other words, since training with normal chips uses only a small part of the whole image, it allows considerable savings in computation.

3.3. Hard Chip Mining

If we only use normal chips that involve ground-truth object instances, our model rarely observes background-regions, which lead to increase false-positive rate. One of the solutions is to use background regions for training, in which object instances are likely to be present. For this, previous approaches [24, 13] first train region proposal network (RPN) for a couple of epochs. Then, object proposals from this light RPN are used to find false-positive regions. Instead of using light RPN, we use models trained from normal chips, which enables us to leverage hard examples.

In addition to false-positive cases, if the imbalance between foreground classes is severe, the overall detection accuracy could be lowered. To resolve foreground-class imbalance while efficiently reducing misclassification between foreground classes, we perform patch-level instance augmentation as shown in the middle of Fig. 1. As a first step, we generate object pool by using ground truth bounding boxes from all instances in the dataset. We then sample patches from object pool with different probabilities for each class and paste them into the existing randomly selected images.

Note that we paste patches in the object pool to the entire image (referred to canvas image) not the chip. We keep the existing annotations in each canvas image and continue to add annotations for added instances. During patch-level augmentation, we obtain both the object pool and the canvas images from the existing dataset. In addition to the existing dataset, we use external datasets for obtaining canvas images. One of the challenges when using external datasets is that classes can be different between an existing dataset and external datasets. When we use canvas images from external datasets, existing object instances with ambiguous or conflicting classes in the canvas images are masked so that they could not affect the training process. Fig. 2 shows samples of patch-level augmented images. In the figure, the red bounding boxes refer to the masked regions due to class ambiguity when using external datasets. Fig. 3 shows the ratio of object instances in each category before and after applying chip mining and patch-level augmentation. From the figure, we can clearly see that patch-level augmentation can efficiently handle the foreground-class imbalance that exists in the dataset.

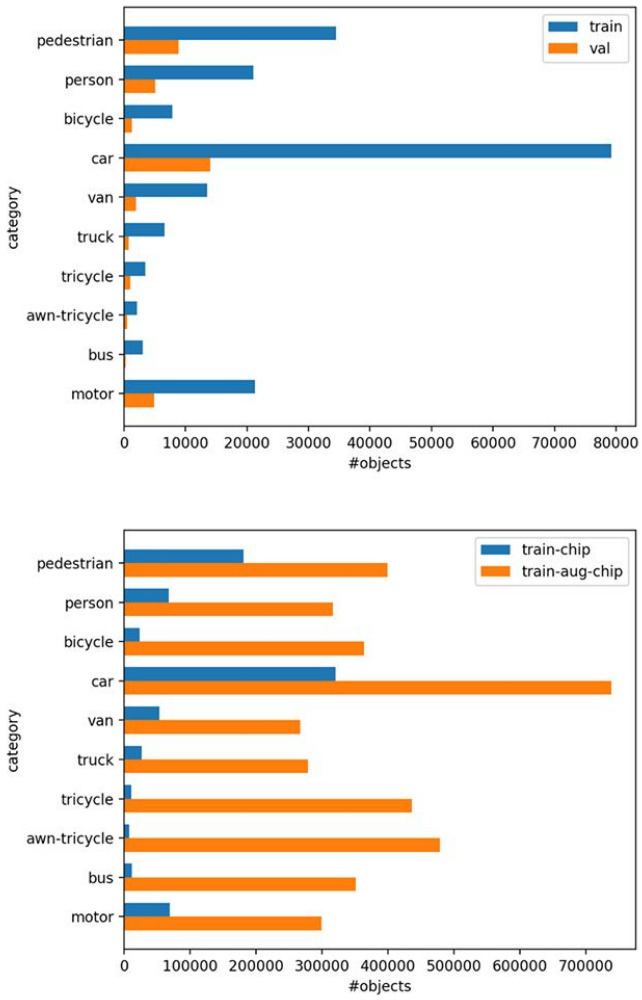

Figure 3. The number of object instances per category in VisDrone-DET. (top) number of object instances in original images (bottom) number of object instances on chips. 'train-chip' refers to a set consisting of chips from training images, while 'train-aug-chip' refers to a set consisting of chips to which patch-level-augmentation is applied.

Once the patch-level augmented images are obtained, we can generate hard examples by performing an inference to them with the model, which was already trained from normal chips. Note that inference proceeds at image-level, not at chip-level. We then generate hard chips for each scale, as in normal chip mining, from the misclassified foreground or background regions. Finally, we retrain the model using both normal chips and the proposed hard chips. As a result,

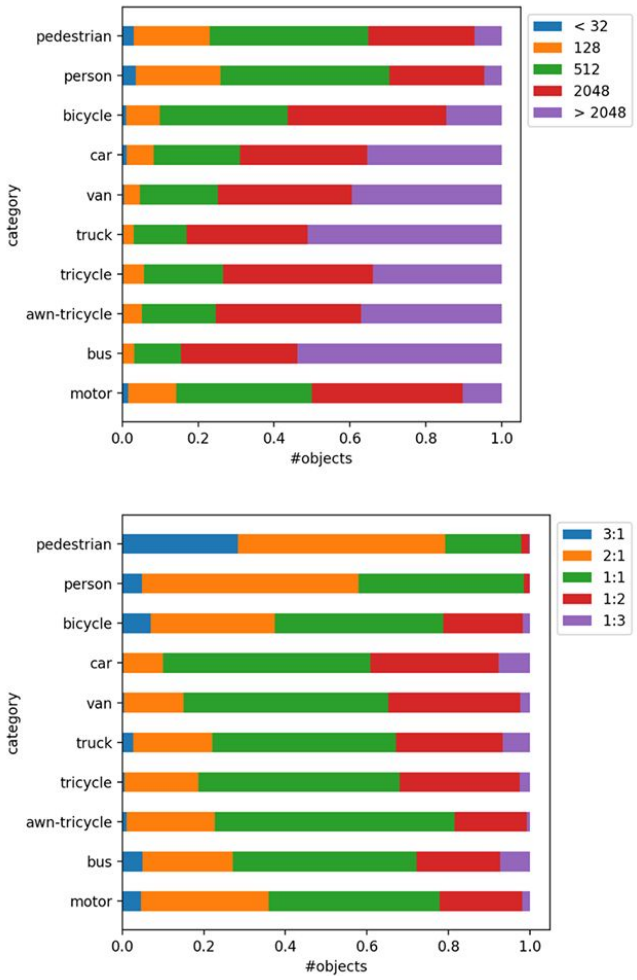

Figure 4. Rate information for object instances per category in VisDrone-DET. (top) sizes of object instances (bottom) aspect ratios of object instances

our model can efficiently solve foreground-class imbalances while actively using hard examples for model training.

4. Experiments

We demonstrate the effectiveness of our hard chip mining method on VisDrone-DET dataset [27]. In Section 4.1, we briefly describe the datasets and experimental settings used in our experiments. We then present comparative results in Section 4.2. Finally, we show the qualitative results of the proposed method in Section 4.3.

4.1. Experimental Setup

We evaluate our method on VisDrone-DET dataset [6], a large-scale benchmark of 10,209 images taken from drones. This dataset focuses on detecting predefined categories of objects, e.g., ignored regions (0), pedestrian (1), people (2), bicycle (3), car (4), van (5), truck (6), tricycle (7), awning-tricycle (8), bus (9), motor (10), others (11). Fig. 4 shows

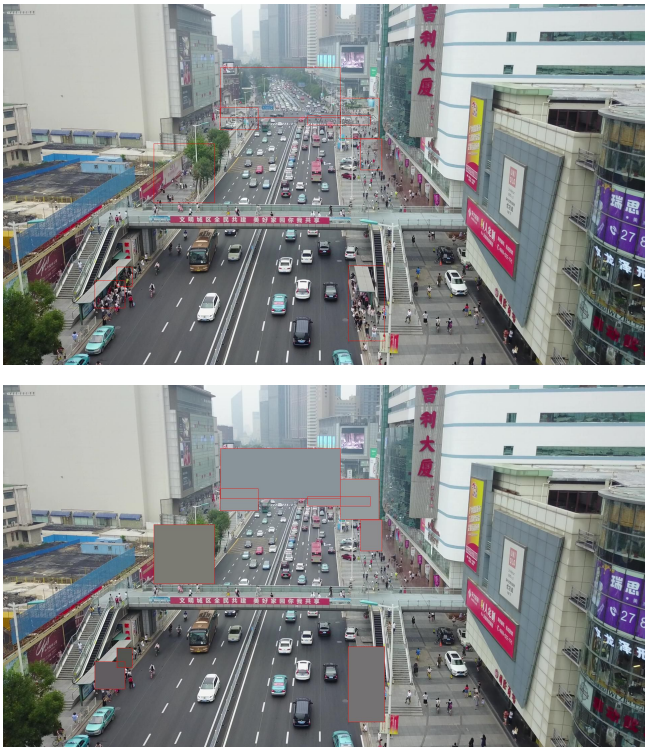

Figure 5. Images before and after applying masking to 'ignored regions' highlighted in a red box (top) original image (bottom) masked image

detailed information of object instances for each category. As can be seen in the figure, VisDrone dataset includes a large scale difference between categories, and the aspect ratios of object instances are different compared to those in MS COCO dataset. Based on the analysis of information about object instances, we adjust the aspect ratio and scale of anchor. By only adjusting the hyperparameters based on the sample distribution analysis without changing the model architecture, we confirmed a high-performance improvement.

Following the previous approaches [27], we use the $AP^{IoU=0.50:0.05:0.95}$, $AP^{IoU=0.50}$, $AP^{IoU=0.75}$, $AR^{max=1}$, $AR^{max=10}$, $AR^{max=100}$, and $AR^{max=500}$ metrics to evaluate the results of detection algorithms. These criteria penalize missing detection of objects as well as duplicate detection results. Since 'ignored regions' and 'others' categories are not used in the evaluation, we do not use both categories for training. Specifically, we masked the regions to explicitly exclude the 'ignored regions', which leads to improve performance. Fig. 5 shows the input images before and after applying to mask to the 'ignored regions.' Also, in order to compensate for the lack of training data from VisDrone-DET, we additionally use DOTA [26], another large-scale dataset for object detection in aerial images. As in 'ignored regions', we apply instance

Method	AP[%]	AP_{50}[%]	AP_{75}[%]	AR_1[%]	AR_{10}[%]	AR_{100}[%]	AR_{500}[%]
Ours (multi-scale)	**37.15**	**65.54**	**36.56**	0.32	1.47	7.28	**53.78**
Ours (single-scale)	35.64	63.96	34.27	0.4	2.74	17.52	50.19
FPN	32.88	60.66	30.86	**0.43**	**2.77**	14.38	47.72
FPN (Default setting on MS COCO)	29.1	52.84	28.12	0.42	**2.77**	**26.41**	41.34

Table 1. Ablation studies on validation set of VisDrone2019-DET dataset.

Method	AP[%]	AP_{50}[%]	AP_{75}[%]	AR_1[%]	AR_{10}[%]	AR_{100}[%]	AR_{500}[%]
DPNet-ensemble	**29.62**	54.00	**28.70**	0.58	3.69	17.10	42.37
RRNet	29.13	**55.82**	27.23	**1.02**	**8.50**	**35.19**	46.05
Ours	29.13	54.07	27.38	0.32	1.48	9.46	44.53
S+D	28.59	50.97	28.29	0.50	3.38	15.95	42.72
BetterFPN	28.55	53.63	26.68	0.86	7.56	33.81	44.02
HRDet	28.39	54.53	26.06	0.11	0.94	12.95	43.34
CN-DhVaSa	27.83	50.73	26.77	0.00	0.18	7.78	**46.81**
SGE-cascade R-CNN	27.33	49.56	26.55	0.48	3.19	11.01	45.23
EHR-RetinaNet	26.46	48.34	25.38	0.87	7.87	32.06	38.42
CNAnet	26.35	47.98	25.45	0.94	7.69	32.98	42.28
CornerNet*	17.41	34.12	15.78	0.39	3.32	24.37	26.11
Light-RCNN*	16.53	32.78	15.13	0.35	3.16	23.09	25.07
FPN*	16.51	32.20	14.91	0.33	3.03	20.72	24.93
Cascade R-CNN*	16.09	31.91	15.01	0.28	2.79	21.37	28.43
DetNet59*	15.26	29.23	14.34	0.26	2.57	20.87	22.28
RefineDet*	14.90	28.76	14.08	0.24	2.41	18.13	25.69
RetinaNet*	11.81	21.37	11.62	0.21	1.21	5.31	19.29

Table 2. Top 10 comparisons results in the VisDrone-DET2019 challenge. * indicates that the baseline algorithm submitted by committee. More details can be found on the VisDrone homepage (http://aiskyeye.com/)

masking to resolve heterogeneity of categories between VisDrone-DET and DOTA.

4.2. Comparison Results

Since the proposed method is based on FPN, we first show our ablation studies with FPN trained on VisDrone-DET. First of all, we use the default hyperparameters set to MS COCO dataset. Then, we adjust those hyperparameters to be suitable to VisDrone dataset. From the third and last rows of Table. 1, we can see that simple hyperparameter adjustment can improve detection performance significantly. With our hard chip mining method and multi-scale inference, the performance is much more enhanced, as shown in the first and second rows of the Table. 1.

We also evaluate our method by participating in the Visdrone-DET2019 challenge [6]. Table. 2 shows the comparison results on the leaderboard. A total of 46 teams participated in the VisDrone-DET2019 challenge and our method was ranked 3rd, including the baseline methods provided by the challenge organizers. As can be seen in Table. 2, the proposed method outperforms the baselines significantly. Our method shows 11.72 AP higher than CornerNet, which is the best baseline method, and also shows the promising result comparable to the state-of-the-art methods.

4.3. Qualitative Results

The qualitative results of object detection are shown in Fig. 6. From the figure, we can see that our hard chip mining enables to detect small objects or dense objects that have been difficult to detect by previous methods. Furthermore, we can see that the proposed method shows a high detection accuracy regardless of the scale (large or small), illumination of the image (day or night), and the shooting angle (top-view or front-view).

5. Conclusion

Motivated by the observations about foreground-class imbalance and lack of training data (especially hard examples), in this paper, we have presented a novel method called a hard chip mining. We first train our model with normal chips, which consist of ground-truth instances, then use this model to extract hard examples. Finally, our detection model is trained by using a combination of normal and hard chips. In the experiments, we have verified the effectiveness of the proposed method by achieving competitive results on VisDrone-DET2019 challenge. Also, we have conducted ablation studies and shows qualitative results to demonstrate the extensibility of the proposed method.

Figure 6. Qualitative result on VisDrone2019-DET validation set. Different colored bounding boxes mean different kinds of categories.

References

[1] Antreas Antoniou, Amos Storkey, and Harrison Edwards. Data augmentation generative adversarial networks. *arXiv preprint arXiv:1711.04340*, 2017.

[2] Donghyeon Cho, Sungeun Hong, Sungil Kang, and Jiwon Kim. Key instance selection for unsupervised video object segmentation. *arXiv preprint arXiv:1906.07851*, 2019.

[3] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

[4] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In *Proc. of Neural Information Processing Systems (NIPS)*, pages 379–387, 2016.

[5] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 764–773, 2017.

[6] Wen Bian Ling Hu Peng Du, Zhu et al. Visdrone-det2019: The vision meets drone object detection in image challenge results. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 0–0, 2019.

[7] Debidatta Dwibedi, Ishan Misra, and Martial Hebert. Cut, paste and learn: Surprisingly easy synthesis for instance detection. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 1301–1310, 2017.

[8] Alhussein Fawzi, Horst Samulowitz, Deepak Turaga, and Pascal Frossard. Adaptive data augmentation for image classification. In *IEEE Int'l Conf. on Image Processing (ICIP)*, pages 3688–3692. Ieee, 2016.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 2961–2969, 2017.

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[11] Sungeun Hong, Woobin Im, and Hyun S Yang. Cbvmr: content-based video-music retrieval using soft intra-modal structure constraint. In *ACM Int'l Conf. on Multimedia Retrieval (ICMR)*, pages 353–361. ACM, 2018.

[12] Yunho Jeon and Junmo Kim. Active convolution: Learning the shape of convolution for image classification. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 4201–4209, 2017.

[13] Hengduo Li, Bharat Singh, Mahyar Najibi, Zuxuan Wu, and Larry S Davis. An analysis of pre-training on object detection. *arXiv preprint arXiv:1904.05871*, 2019.

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

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[16] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proc. of Int'l Conf. on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[17] 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 *Proc. of European Conf. on Computer Vision (ECCV)*, pages 740–755. Springer, 2014.

[18] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 21–37. Springer, 2016.

[19] Mahyar Najibi, Bharat Singh, and Larry S Davis. Autofocus: Efficient multi-scale inference. *arXiv preprint arXiv:1812.01600*, 2018.

[20] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 779–788, 2016.

[21] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proc. of Neural Information Processing Systems (NIPS)*, pages 91–99, 2015.

[22] Abhinav Shrivastava, Abhinav Gupta, and Ross Girshick. Training region-based object detectors with online hard example mining. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 761–769, 2016.

[23] Bharat Singh and Larry S Davis. An analysis of scale invariance in object detection snip. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 3578–3587, 2018.

[24] Bharat Singh, Mahyar Najibi, and Larry S Davis. Sniper: Efficient multi-scale training. In *Proc. of Neural Information Processing Systems (NIPS)*, pages 9310–9320, 2018.

[25] Zhu Pengfei Du Dawei Bian Wen, Longyin et al. Visdrone-sot2018: The vision meets drone single-object tracking challenge results. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 0–0, 2018.

[26] Gui-Song Xia, Xiang Bai, Jian Ding, Zhen Zhu, Serge Belongie, Jiebo Luo, Mihai Datcu, Marcello Pelillo, and Liangpei Zhang. Dota: A large-scale dataset for object detection in aerial images. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[27] Wen Longyin Bian Xiao Haibin Ling Hu Qinghua Zhu, Pengfei. Vision meets drones: A challenge. *arXiv preprint arXiv:1804.07437*, 2018.

[28] Wen Longyin Du Dawei Bian Zhu, Pengfei et al. Visdrone-det2018: The vision meets drone object detection in image challenge results. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 0–0, 2018.

[29] Wen Longyin Du Dawei Bian others Zhu, Pengfei. Visdrone-vdt2018: The vision meets drone video detection and tracking challenge results. In *Proc. of European Conf. on Computer Vision (ECCV)*, pages 0–0, 2018.