Spatial Resolution-Independent CNN-Based Person Detection in Agricultural Image Data

Alexander Leipnitz(✉), Tilo Strutz, and Oliver Jokisch

Institute of Communications Engineering, Leipzig University of Telecommunications (HfTL), Leipzig, Germany
{leipnitz,strutz,jokisch}@hft-leipzig.de

Abstract. Advanced object detectors based on Convolutional Neural Networks (CNNs) offer high detection rates for many application scenarios but only within their respective training, validation and test data. Recent studies show that such methods provide a limited generalization ability for unknown data, even for small image modifications including a limited scale invariance. Reliable person detection with aerial robots (Unmanned Aerial Vehicles, UAVs) is an essential task to fulfill high security requirements or to support robot control, communication, and human-robot interaction. Particularly in an agricultural context, persons need to be detected from a long distance and a high altitude to allow the UAV an adequate and timely response. While UAVs are able to produce high resolution images that enable the detection of persons from a longer distance, typical CNN input layer sizes are comparably small. The inevitable scaling of images to match the input-layer size can lead to a further reduction in person sizes. We investigate the reliability of different YOLOv3 architectures for person detection in regard to those input-scaling effects. The popular VisDrone data set with its varying image resolutions and relatively small depiction of humans is used as well as high resolution UAV images from an agricultural data set. To overcome the scaling problem, an algorithm is presented for segmenting high resolution images in overlapping tiles that match the input-layer size. The number and overlap of the tiles are dynamically determined based on the image resolution. It is shown that the detection rate of very small persons in high resolution images can be improved using this tiling approach.

Keywords: Convolutional neural network · Person detection · Resolution invariance · Input-layer scaling · Image tiling

1 Introduction

The development of high quality, lightweight camera systems enabled their deployment on drones and therefore many new application areas. The high resolution of the captured images allows the detection of objects from long distances

A. Ronzhin et al. (Eds.): ICR 2020, LNAI 12336, pp. 189–199, 2020.
https://doi.org/10.1007/978-3-030-60337-3_19

and high altitudes. This is an essential safety requirement for an automated operation of flying drones for example in digital farming. In manual operation, the pilot typically analyzes the video-stream of the drone directly on his remote control, however, in an autonomous scenario this task has to be done automatically. The processing of high resolution images with advanced CNN methods requires a lot of computational power and memory and is therefore still limited to high-end hardware. Feasible CNNs, like the popular and state-of-the-art YOLOv3 architecture [1], provide a relativly small input-layer. Hence, an adjustment of the input image is required, which can be realized by (i) cropping, (ii) resizing with aspect-ratio preservation and padding or (iii) resizing by trivial 2D sub-sampling disregarding the aspect ratio. In the original YOLOv3 implementation [2] method (iv) is used, while the framework utilized for our investigations [3] sub-samples the images by method (v). Resizing with aspect-ratio preservation can lead to the smallest object sizes among the three adjustment methods if the input-layer and input-image aspect ratios differ. Using method (vi), the object-information loss is limited by using the appropriate horizontal and vertical scaling-factors. However, different scaling in horizontal and vertical direction leads to aspect-ratio distortions that can decrease the detection performance as CNNs are not robust against such image modifications [4,5]. When addressing high resolution images, scaling can even make the detection of very small objects impossible. Several solutions to this problem have already been proposed in literature.

Single-stage approaches, on the one hand, focus on special network architectures that allow the processing of the whole high resolution image at once and avoid or minimize scaling. In [6], a method is presented that reduces the memory footprint on the GPU by not storing the entire output feature maps after each layer at the same time but only the parts that are needed for the next processing step. However, this approach cannot be used for most network architectures (like YOLOv3) as the whole activation map is not present at any time and certain operations (e.g. batch normalization) are not possible. Other approaches concentrate on special network architectures that have real-time drone application [7,8] or improved scale invariance [9,10] in mind, but still rely on scaling the input image to the input-layer size.

Two-stage approaches, on the other hand, search for interesting image areas first and run the object detector only on these regions to minimize scaling of the input image. The region proposals can also be realized with neural networks [11–15]. While these methods limit the computational efforts, the risk of missing very small persons is high. All state-of-the-art methods utilize some scaling or error-prone pre-selection of interesting image areas. Our contribution investigates the capability of different YOLOv3 architectures to find very small persons in high-resolution images and proposes an image scaling-free method to improve the detection rate.

2 Methods

The processing pipeline of the proposed approach is based on the divide-and-conquer principle. At first, the input image is segmented into overlapping regions (tiles) that match the CNN input-layer size to avoid image scaling so that persons keep their size in pixels. One of three investigated CNN architectures (YOLOv3, YOLOv3-tiny, YOLOv3-spp) is then applied to every tile. The last step consists of merging the results for each tile to a global information about people positions.

2.1 Tiling

Any image with a width w and/or height h bigger than the input-layer size of the CNN can be segmented into a minimal number of overlapping tiles with a width w_t and height h_t that match the input-layer size. Figure 1 depicts the segmentation of an image in four tiles. The calculation of the necessary amount of horizontal tiles n_x with an empirically determined maximum overlap x_{max} of 80% to cover an image with arbitrary resolution is shown in Listing 1.1. The same approach can be applied vertically.

Listing 1.1. Calculating the number of horizontal tiles with a width of 416.

```
float x_max = 0.8;
float x = 0.9;
int w_t = 416;
int n_x = ceil(w / w_t);
while (x > x_max) {
  x = (n_x * w_t - w) / (w_t * (n_x - 1));
  n_x -= 1;
}
```

The maximum overlap x_{max} can be adjusted with respect to the person sizes to avoid splitting them between multiple tiles. If no suitable number of tiles ($n_x \geq 1$) can be found based on the minimal/maximal overlap criteria $0 \leq x \leq x_{max}$, the image has to be down-scaled slightly. For images smaller than the input-layer

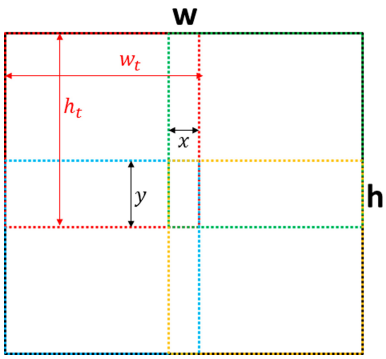


Fig. 1. Image segmentation in four overlapping tiles (red, green, blue, yellow). (Color figure online)

size zero-padding is used. Persons that are completely inside the overlap area are contained in both tiles. If the persons are larger than the overlap area or on a border between two tiles, only a part of them is represented in one tile. The smaller such a part is, the more difficult it is to derive relevant features of the person.

2.2 YOLOv3

The YOLOv3 architecture [1] is a convolutional neural network for object detection that processes the whole input image in a single pass. It predicts bounding boxes around objects in the original image on three different scales (three output paths) to allow multi-scale detection. The highest-resolution output path is able to find the smaller objects in the input image, while the smallest-resolution output path contains information about the bigger objects. The whole architecture consists of 107 layers making it rather complex. In the default configuration, the input layer has a size of 416×416 pixel. The horizontal and vertical dimensions of the input layer can be adjusted in steps of 32 pixels, and all the feature maps will scale accordingly. In our tests, we used a standard input-layer size of 416×416 pixels and also a slightly increased size of 608×608 pixels to address the differences for high-resolution images. Larger input-layer sizes are not feasible as the memory requirements scale linearly with the increased size. To avoid hardware limitations, other compromises would have to be made in the training process, which outweigh the advantage of the bigger input-layer size with less input-image scaling.

2.3 YOLOv3-tiny

YOLOv3-tiny is a slimmer and less complex variant of the YOLOv3 architecture due to a reduced number of layers (21 instead of 107). It lacks the branch that can detect the smallest objects but allows faster processing. The lower complexity also reduces the tendency to overfit on small data sets.

2.4 YOLOv3-spp

The spatial pyramid pooling module (spp) introduced in [10] can be integrated into the YOLOv3 architecture to obtain a more scale-invariant solution. It utilizes three different parallel max-pooling layers, which pool the input-feature map with three sliding window sizes. The resulting feature maps are then concatenated with the input-feature map before the YOLOv3 architecture continues. This allows the pooling and concatenation of multi-scale local region features.

2.5 Merging

YOLOv3 architectures output a list of bounding boxes accompanied by confidence scores per tile. This local information has to be merged into a global one by

applying offsets to bounding boxes according to the tile location in the original image. In image areas with overlapping tiles, objects are often detected twice. In order to decide which bounding box to keep, the Intersection over Union (IoU) between two overlapping boxes is calculated. If $IoU > 0.7$, the two boxes get merged by using the most-outer corners, and the higher of the two confidence scores of the two boxes is kept. If $0.5 < IoU \leq 0.7$, the box with the higher confidence score is kept and the other one is deleted. For boxes with very small overlap ($IoU \leq 0.5$) the boxes are considered to contain different persons.

3 Investigations

We investigated three different variants of the YOLOv3 network architecture to identify their abilities for a detection of small persons in high-resolution images and to compare the default image-scaling with our segmentation-and-merging approach.

All models have been pre-trained on the MSCOCO trainval dataset [16], while a mini-batch size of 64 has been used for all trainings. Each of the three YOLOv3 architectures is fed with either resized (method (iii)) or segmented input images that fit the input-layer size.

3.1 VisDrone Data Set

The VisDrone object detection data set [17] is a large-scale benchmark for object detection tasks with drones. It contains various scenarios in urban and country environment. The corresponding annotation comprises ten classes (*pedestrian*, *person*, *car*, *van*, *bus*, *truck*, *motor*, *bicycle*, *awning-tricycle*, and *tricycle*). As we focus on person detection only, the classes *pedestrian* and *person* are utilized and merged into a single *person* class. The VisDrone data set consists of 8 629 images with public available annotation. This set is divided into 75.0% training, 6.4% validation and 18.6% additional data for debugging and further validation (test-dev data). The resolution of the images ranges from 480×360 up to $2\,000 \times 1\,500$ ($w \times h$) pixels. The data set contains images with many very small persons. The smallest annotated bounding box is only one pixel tall (h_b) and three pixels wide (w_b) in an image with a resolution of $1\,916 \times 1\,078$ pixels. This corresponds to a person size which is even smaller than one pixel after the image is down-scaled to $w_t \times h_t = 416 \times 416$ or 608×608 pixels depending on the CNN input-layer. The scaling factor f corresponds to

$$f = \frac{w_t \cdot h_t}{w \cdot h} \tag{1}$$

and the resulting person size s is

$$s = w_b \cdot h_b \cdot f \ . \tag{2}$$

When the images are segmented into tiles, the small persons are represented by an unchanged amount of pixels in the input-layer, and not one persons vanishes (s large enough) due to $f = 1$.

The characteristics of this data set when either using image-scaling or the proposed segmentation approach is summarized in Table 1. The number of boxes represents the number of annotated persons in the data set (147, 747). When scaling to the input-layer size of 416×416 or 608×608 pixels some of these person will vanish as they will be smaller than one pixel ($\min(s) < 1$).

Segmenting the image in overlapping tiles increases the total number of bounding boxes due to their multiple representation in the overlapping areas (cf. Sect. 2.1). If a bounding boxes is only partially inside a tile, it has to be cropped to the tiles dimensions. This leads to the varying person size s distribution between the two different tile sizes. However, if the new bounding box size is smaller than 20% of the original bounding box size in one of those tiles, we discard the corresponding box as it is not very likely to contain enough features for a person.

Table 1. Analysis of VisDrone data set in terms of person sizes depending on input-layer size.

	Scaling to		Tile size	
	416×416	608×608	416×416	608×608
No. of boxes	147,747	147,747	657,111	990,809
max (s)	9,669.9	20,655.9	60,629.0	72,669.9
mean (s)	75.8	161.9	488.4	550.0
min (s)	0.25	0.54	2.0	3.0

3.2 AgriDrone Data Set

AgriDrone is a self-captured data set with focus on person detection in agricultural applications. All 4 586 images have been captured by two different drones: DJI Mavic2 Enterprise and DJI Mavic Pro between Spring and Winter. They share the same resolution of $3\,840 \times 2\,160$ pixels. The data set is split into 70% training 10% validation and 20% test data. The AgriDrone data set is much smaller than the VisDrone set, and the relative sizes of persons are larger. Table 2 summarizes the AgriDrone data and show the same basic findings as Table 1. In this data set, the persons are large enough to be preserved despite the scaling of the images ($\min(s) > 1$).

4 Results and Discussion

In total, 24 CNN models have been trained, validated and tested. Three architectures have been investigated in combination with two different input-layer sizes (416×416 and 608×608), with or without tiling, and both different data sets. Following measures in the style of [16] are used for the evaluation of the detection performance: the mean Average Precision (mAP) metric at a IoU threshold

Table 2. Analysis of AgriDrone data set in terms of person sizes depending on input-layer size.

	Scaling to		Tile size	
	416 × 416	608 × 608	416 × 416	608 × 608
No. of boxes	8,746	8,746	14,155	13,296
max (s)	6,890.2	14,718.1	127,360.0	195,048.0
mean (s)	235.5	503.0	8,460.6	9,415.7
min (s)	6.68	14.26	196.0	247.0

Table 3. Object detection results with an input-layer size of 416 × 416 pixels.

Data set	Seg. & Mer.	YOLOv3			YOLOv3-tiny			YOLOv3-spp		
		mAP	mAP_{50}	mAP_{75}	mAP	mAP_{50}	mAP_{75}	mAP	mAP_{50}	mAP_{75}
VisDrone training										
Validat	Off	9.56	33.04	2.19	2.53	10.84	3.21	9.57	32.98	2.12
Validat	**On**	**19.31**	**55.02**	**8.33**	**8.92**	**31.10**	**2.13**	**18.70**	**55.75**	**6.89**
Test-dev	Off	4.19	15.16	0.89	0.96	4.29	0.09	4.35	15.79	0.93
Test-dev	**On**	**10.54**	**32.73**	**4.01**	**4.52**	**16.54**	**1.02**	**10.12**	**31.69**	**3.82**
AgriDrone training										
Validat	Off	**35.55**	**89.83**	**18.32**	21.23	64.96	7.96	**38.58**	**91.61**	**23.39**
Validat	**On**	31.08	78.50	17.64	**31.32**	**77.86**	**17.78**	32.65	79.90	18.88
Test	Off	**34.01**	**85.66**	**17.04**	21.87	65.05	8.31	**36.92**	**88.32**	**21.55**
Test	**On**	29.25	73.74	15.38	**31.11**	**76.53**	**16.07**	32.36	78.13	18.18

of 0.5 (mAP_{50}) and at a IoU threshold of 0.75 (mAP_{75}) as well as the averaged precision over IoU thresholds 0.5 . . . 0.95 in steps of 0.05 (mAP).

Table 3 (input-layer size 416 × 416) and Table 4 (input-layer size 608 × 608) list the overall results of our investigations. When comparing the entries related to the original approach using scaling (Segmentation & Merging, Seg. & Mer.: Off), it can be seen that an increased input-layer size helps to detect more persons. The improvements range up to 12.92% in the mAP_{50} metric on the VisDrone validation set using the YOLOv3-spp architecture. The segmentation-and-merging method always leads to an improvement in detection performance (with one exception), regardless of the architecture and input-layer size applied to the VisDrone data set, and the best results could be achieved, when the corresponding architecture was trained on the segmented images with a tile size of 416 × 416 pixels. These results prove that an enlarged input layer of size 608 × 608 pixels is still not sufficient for a reliable person detection. Instead, the person size should be kept using the proposed segmentation-and-merging approach. For the VisDrone data set, the YOLOv3-tiny architecture generally performs the worst due to its low complexity, while the two other ones are on par.

The mentioned effects can only partially be reproduced on the AgriDrone data. The YOLOv3 and YOLOv3-spp models are probably overfitting on this data, as it is smaller and has a lower variability. Only with the YOLOv3-tiny

architecture, the segmentation-and-merging approach leads to an improvement in the detection performance. As the persons are larger in this data set, the best results with the YOLOv3-tiny architecture are archived with a tile size of 608 × 608 pixels (mAP_{50} of 85.63% and 86.25%, respectively). The improvements of the proposed segmentation-and-merging approach are not that noticeable with this data set as the input-image scaling does not lead to vanished persons (see Table 2). It can also be speculated that the very large persons that can fill up to 73.6% of a 416 × 416 pixel-sized tile (Table 2) are too large to be detected or spread over multiple tiles, so that the bounding boxes cannot be properly merged. Figure 2 visualizes the improvement of the segmentation-and-merging approach for the (a) VisDrone and (b) AgriDrone data sets. The magenta bounding boxes represent the ground-truth, the yellow boxes the true-positives with image-scaling and the cyan ones the true-positives with our approach (segmentation plus merging). The proposed method detects a lot more of the small humans in Fig. 2(a) while also the relatively big human on the left in Fig. 2(b) is now localized correctly.

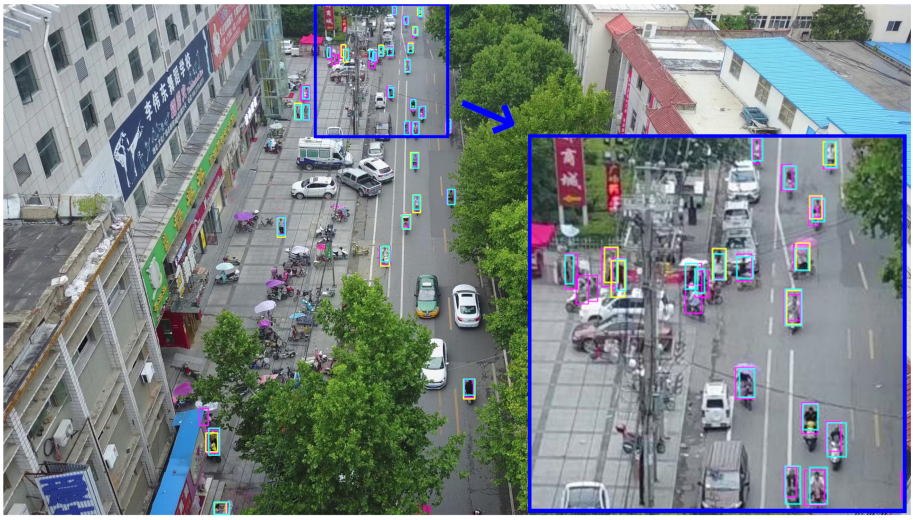
Table 4. Object detection results with an input-layer size of 608 × 608 pixels.

Data set	Seg. & Mer.	YOLOv3			YOLOv3-tiny			YOLOv3-spp		
		mAP	mAP_{50}	mAP_{75}	mAP	mAP_{50}	mAP_{75}	mAP	mAP_{50}	mAP_{75}
VisDrone training										
Validat	Off	14.91	45.78	5.15	5.07	19.46	0.88	15.01	45.90	4.85
Validat	**On**	**19.61**	**53.93**	**8.76**	**8.77**	**29.99**	**2.26**	**17.42**	**50.16**	**7.40**
Test-dev	Off	7.04	23.40	2.12	2.14	8.83	0.27	7.64	25.44	2.19
Test-dev	**On**	**10.79**	**32.65**	**4.11**	**4.63**	**16.81**	**1.08**	**9.70**	**30.51**	**3.51**
AgriDrone training										
Validat	Off	**43.04**	**95.35**	**31.11**	31.61	84.15	14.93	**39.83**	**94.47**	**23.41**
Validat	**On**	38.36	85.56	24.31	**37.44**	**85.73**	**23.36**	38.10	85.95	24.23
Test	Off	**41.72**	**91.99**	**29.35**	31.45	81.08	15.39	**39.31**	**91.39**	**23.60**
Test	**On**	38.42	87.32	24.32	**38.23**	**86.28**	**25.08**	37.71	85.83	23.14

The improvement in object detection rates especially for small persons can also be seen in Fig. 3, in particular for the VisDrone data in Fig. 3(a). The relative person size ρ represents the person size in relation to the original image resolution with

$$\rho = \frac{w_b \cdot h_b}{w \cdot h} \,. \tag{3}$$

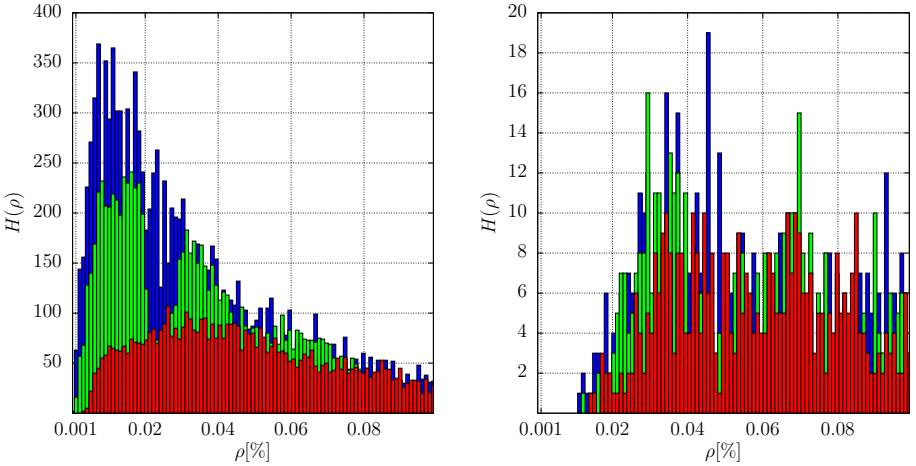
The blue bars show the number of ground-truth bounding boxes of the according relative bounding box size. Using the YOLOv3 416 × 416 architecture and input-image scaling shows a drop-off in detections (true-positives at $IoU = 0.5$) towards smaller person sizes (red bars). When the proposed segmentation-and-merging algorithm is applied, the detection of small persons is improved (green bars). In Figure 3(b) the improvements are not as visible due to the lack of very small persons and overfitting of the YOLOv3 architecture on this data set.



(a) VisDrone data

(b) AgriDrone data

Fig. 2. Example detections with YOLOv3 416 × 416. Magenta: ground-truth, yellow: true-positives at $IoU = 0.5$ (image-scaling), cyan: true-positives at $IoU = 0.5$ (image segmentation and merging). (Color figure online)



(a) VisDrone validation data **(b)** AgriDrone validation data

Fig. 3. Histogram of small relative person sizes ρ: annotated (blue), correctly detected with default image scaling (red) and correctly detected with segmentation-and-merging approach (green). (Color figure online)

5 Summary

The study has shown that the detection rate of YOLOv3 architectures can be improved with the proposed method. Instead of using image-modifying preprocessing (scaling or region selection), a segmentation-and-merging approach has been investigated. All three investigated YOLOv3 architectures are able to run in real-time (more than 30 frames per second) on a NVIDIA RTX 2080TI GPU in the original version. However, the processing time scales linearly with the number of tiles. The time saved by omitting the slow scaling functions is consumed by the additional merging algorithm. Especially with very high resolution drone images with a lot of tiles (e.g. AgriDrone images) this can lead to non-real-time processing, even with optimizations like parallel processing of tiles. In long-distance image capturing scenarios with a drone, the safety requirements outweigh the real-time requirements, since the detection of humans from a greater distance gives the drone more time to react.

The proposed approach is suitable for all image resolutions bigger than the input-layer size. We proved that the detection rates of small persons in high-resolution images can be improved, which makes CNNs usable for person detection with an UAV. Future work should also include the scaled image version to avoid not detecting persons larger than the tile size. To support reproducible research, we made the software and data of our study publicly available at [18].

Acknowledgements. The authors acknowledge the financial support by the Federal Ministry of Education and Research of Germany (BMBF) within the framework of the EU Era.Net RUS+ project HARMONIC (national project number 01DJ18011).

References

1. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
2. Redmon, J.: Darknet framework for object detection. https://github.com/pjreddie/darknet. Accessed 23 Jul 2020
3. Bochkovskiy, A.: Improved darknet framework for object detection. https://github.com/AlexeyAB/darknet. Accessed 23 Jul 2020
4. Rosenfeld, A., Zemel, R.S., Tsotsos, J.K.: The elephant in the room. arXiv preprint arXiv:1808.03305 (2018)
5. Azulay, A., Weiss, Y.: Why do deep convolutional networks generalize so poorly to small image transformations? J. Mach. Learn. Res. **20**(184), 1–25 (2019)
6. Pinckaers, H., Litjens, G.J.S.: Training convolutional neural networks with megapixel images. arXiv preprint arXiv:1804.05712 (2018)
7. Zhang, P., Zhong, Y., Li, X.: Slimyolov3: Narrower, faster and better for real-time UAV applications. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (2019)
8. Steinmann, L., Sommer, L., Schumann, A., Beyerer, J.: Fast and lightweight person detector for unmanned aerial vehicles. In: EUSIPCO (2019)
9. Tayara, H., Chong, K.: Object detection in very high-resolution aerial images using one-stage densely connected feature pyramid network. Sensors **18**(10), 3341 (2018)
10. Huang, Z., Wang, J.: DC-SPP-YOLO: dense connection and spatial pyramid pooling based YOLO for object detection. arXiv preprint arXiv:1903.08589 (2019)
11. Yang, F., Fan, H., Chu, P., Blasch, E., Ling, H.: Clustered object detection in aerial images. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 8311–8320 (2019)
12. Lu, Y., Javidi, T.: Efficient object detection for high resolution images. In: 2015 53rd Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 1091–1098 (2015). https://doi.org/10.1109/ALLERTON.2015.7447130
13. Zhang, J., Huang, J., Chen, X., Zhang, D.: How to fully exploit the abilities of aerial image detectors. In: The IEEE International Conference on Computer Vision (ICCV) Workshops (2019)
14. Pang, J., Li, C., Shi, J., Xu, Z., Feng, H.: $\mathcal{R}^2$ -CNN: Fast tiny object detection in large-scale remote sensing images. IEEE Trans. Geosci. Remote Sens. **57**(8), 5512–5524 (2019). https://doi.org/10.1109/TGRS.2019.2899955
15. Růžička, V., Franchetti, F.: Fast and accurate object detection in high resolution 4k and 8k video using gpus. In: 2018 IEEE High Performance extreme Computing Conference (HPEC), pp. 1–7 (2018). https://doi.org/10.1109/HPEC.2018.8547574
16. Lin, T.-Y., et al.: Microsoft COCO: common objects in context. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 740–755. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_48
17. Zhu, P., Wen, L., Bian, X., Ling, H., Hu, Q.: Vision meets drones: A challenge. arXiv preprint arXiv:1804.07437 (2018)
18. Leipnitz, A.: Tile based object detection. http://www1.hft-leipzig.de/leipnitz/papers/TiledDetection-resources. Accessed 23 Jul 2020