IterDet: Iterative Scheme for Object Detection in Crowded Environments

Danila Rukhovich
d.rukhovich@samsung.com
Konstantin Sofiiuk
k.sofiiuk@samsung.com
Danil Galeev
d.galeev@samsung.com
Olga Barinova
o.barinova@samsung.com
Anton Konushin
a.konushin@samsung.com

Samsung AI Center

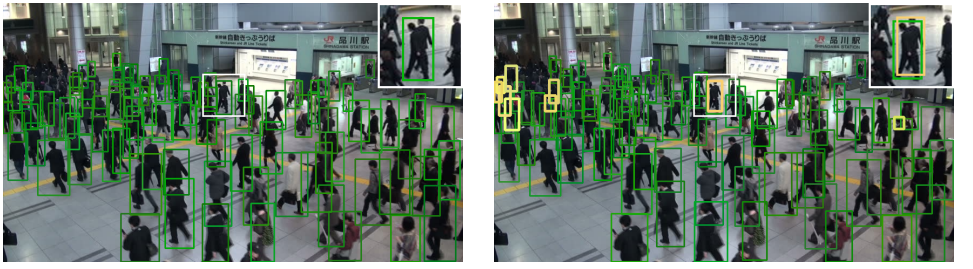
Abstract

Deep learning-based detectors usually produce a redundant set of object bounding boxes including many duplicate detections of the same object. These boxes are then filtered using non-maximum suppression (NMS) in order to select exactly one bounding box per object of interest. This greedy scheme is simple and provides sufficient accuracy for isolated objects but often fails in crowded environments, since one needs to both preserve boxes for different objects and suppress duplicate detections. In this work we develop an alternative *iterative scheme*, where a new subset of objects is detected at each iteration. Detected boxes from the previous iterations are passed to the network at the following iterations to ensure that the same object would not be detected twice. This iterative scheme can be applied to both one-stage and two-stage object detectors with just minor modifications of the training and inference procedures. We perform extensive experiments with two different baseline detectors on four datasets and show significant improvement over the baseline, leading to state-of-the-art performance on CrowdHuman and WiderPerson datasets. The source code and the trained models are available at https://github.com/saic-vul/iterdet.

1 Introduction

The general task of object detection is to map an image to a set of boxes with one box per object of interest and each box tightly enclosing corresponding object. In recent years, deep learning-based methods for object detection have evolved and showed significant improvements in terms of speed and accuracy [8, 10, 12, 14, 20].

All deep learning-based detectors densely sample and independently evaluate possible object locations, resulting in numerous boxes containing almost identical image content. Thus, instead of one detection per object, each object triggers several bounding boxes of



baseline recall: 78.8, AP: 76.81

IterDet 1 iter. recall: 75.9, AP: 74.28
IterDet 2 iter. recall: **82.5**, AP: **79.59**

Figure 1: Comparison of original Faster RCNN detections (left image) and the proposed IterDet based on Faster RCNN (right image) on the same image from CrowdHuman *test* set with *visible* annotations. The boxes found on the first and second iterations are marked in green and yellow respectively. The values of recall and AP for baseline and IterDet after the first and the second iterations are shown below the images. See text for more details.

varying confidence. This redundant set of detected bounding boxes is then filtered by non-maximum suppression (NMS) or similar techniques in order to produce exactly one bounding box per object. This *greedy scheme* is designed mainly for the cases when isolated instances of the same object class are present in the image.

One of the known problems of all modern object detectors is the difficulty to handle crowded environments that contain multiple overlapping objects of the same class (*e.g.* people in the street or bacteria in microscopy images). Main reasons for this effect are as follows. First, in presence of multiple objects of the same class it becomes extremely difficult to distinguish whether two boxes belong to the same object or correspond to different overlapping objects. Second, weak visual cues of heavily occluded instances can hardly provide sufficient information for accurate object detection. A few works try to improve the NMS step of the standard greedy scheme [2, 7, 9, 11, 15, 21]. Despite improving accuracy, these approaches do not fully solve the problem. In all variants of NMS there is always a trade-off between precision and recall, as one needs to both remove redundant detections of the same object and preserve the hard-to-detect occluded objects.

In this work we develop a novel *iterative scheme* (IterDet) for object detection. Rather than detecting all objects in the image simultaneously, our scheme provides detection results in iterations. At each iteration, a new subset of objects is detected. Detected boxes from the previous iterations are passed to the network at the next iterations to avoid repetitive detections. Proposed iterative scheme can be applied to any of the existing object detection methods and requires only minor modifications to the training and inference procedures.

Figure 1 shows the results of IterDet for Faster R-CNN on a test image from CrowdHuman dataset. True positive boxes with scores above 0.1 are shown, and false positives are omitted for clarity. At the second iteration, 9 additional objects (shown in yellow) from 137 are added, overtaking the baseline Faster RCNN by 5 true positives and 2.7% of average precision (AP). In the top-right corner of the images we show an example of two strongly overlapping objects that the baseline detector is unable to find, while IterDet detects both objects after just two iterations.

A few works have introduced alternative network architectures that can handle image context and are more suitable for crowded environments [4, 6, 8]. For instance, [19] proposed a convolutional-recurrent model for sequence generation that is trained with a special

Hungarian loss function. In contrast, our approach is not restricted to detecting one object per iteration and is more computationally efficient. Instead of using LSTM memory for storing the information about previously detected objects, we explicitly provide it to the network in a form of object masks. On the one hand, such approach guarantees that no previously detected bounding boxes are accidentally forgotten. On the other hand, it allows to use the history of detections in a much deeper network compared to [19]. Another advantage of our approach is the ease of integration into the state-of-the-art object detection methods. For instance, one of the recent methods PS-RCNN [4] is based on a similar idea of first detecting simple objects and then detecting more difficult ones. However, it requires deep integration into an object detection network, therefore such modification is presented only for RCNN-based detectors.

We perform extensive experiments with both one-stage (RetinaNet) and two-stage (Faster RCNN) object detectors on four challenging datasets (AdaptIS ToyV1 and ToyV2 [18], CrowdHuman [17], and WiderPerson [23]). We compare results of IterDEt with baseline and the results from the literature. Experiments show significant improvement of accuracy over the baselines on all datasets, setting new state-of-the-art on both CrowdHuman and WiderPerson datasets.

2 Related work

Standard methods for object detection. Deep learning-based object detectors can be roughly divided into two groups: two-stage detectors and one-stage detectors. Two-stage detectors are based on proposal-driven mechanism [5, 14]. They contain two subnetworks: the first one produces a sparse set of candidate object locations, while the second one classifies object locations as one of the foreground classes or as a background. Despite the advances in one-stage methods, two-stage methods still demonstrate state-of-the-art accuracy on some challenging datasets. One-stage methods are applied over a regular, dense sampling of object locations, scales, and aspect ratios [10, 12]. Latest works on one-stage detectors demonstrate higher speed with the same or even better accuracy as two-stage methods. Most recently, anchor-free one-stage methods [20] have emerged. Our iterative scheme is applicable to both one-stage and two-stage object detectors.

In all of aforementioned deep learning-based methods object detection is interpreted as a classification problem that estimates probabilities of object classes being present for multiple locations in an image. Class probabilities are estimated independently for each location. On the contrary, in our iterative scheme the history of detections from the previous iterations is passed to the detector at the following iterations, providing the context for resolving ambiguities.

Variations of non-maximum suppression. The standard NMS algorithm greedily selects detections with higher score and removes the less scored neighbours. A wide suppression parameters thus improves the precision and the narrow suppression improves the recall. Consequently, crowded environments are the most challenging case for NMS, since both wide and narrow suppressions lead to errors. Several works try to modify the NMS algorithm. Rothe *et al*. [15] explore a formulation of NMS as a clustering problem. Hosang *et al*. [7] reformulate NMS as a rescoring task that seeks to decrease the score of detections that cover the already detected objects. Soft-NMS [2] is a simple algorithm that decays the detection scores of all other objects as a continuous function of their overlap with a target object. Fitness NMS [21] can be used in conjunction with Soft NMS for additional improvements.

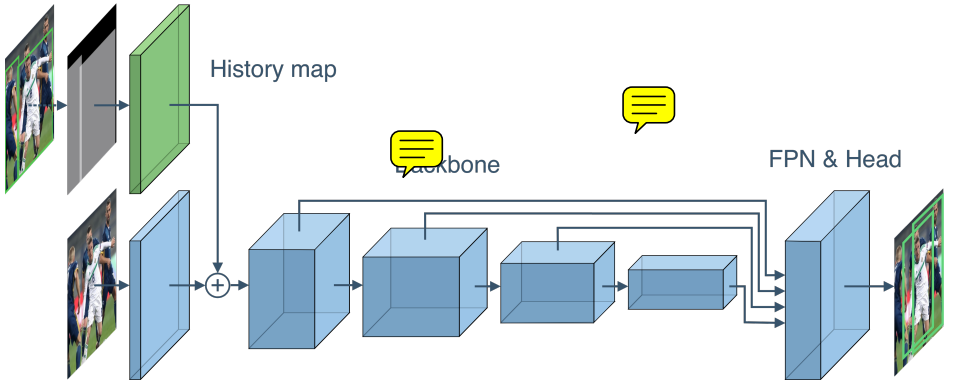


Figure 2: Proposed IterDet scheme. The unchanged meta-architecture of an arbitrary detector is marked blue. The added fusing block for the history map is marked green. Out of the 4 overlapping objects in the image, 2 are in the history, where they were either randomly sampled at the training step, or detected during previous iterations of the inference. The remaining 2 are predicted by the detector.

Adaptive NMS [11] adds an extra branch to the network that estimates density of the objects, which is later used for choosing parameters of NMS. R2NMS [9] simultaneously predicts the full and visible boxes of an object, requiring an additional annotation for training.

In contrast to these works, our proposed scheme is iterative, and at each iteration we need to detect only a subset of objects. Therefore, we are free to miss the more difficult objects at the first iteration, since missed objects can be detected later on. We do not need to assure high recall at each iteration, being able to set wider suppression parameters to favor precision.

Alternative network architectures for crowded environments. A few works propose alternative architectures for object detection beyond independent evaluation of class probabilities at each location. Stewart *et al*. [19] propose a recurrent LSTM layer for sequence generation trained with a Hungarian loss function that operates on sets of detections. Hu *et al*. [8] propose an object relation module that processes a set of objects simultaneously through interaction between their appearance feature and geometry, allowing modeling of their relations. Goldman *et al*. [6] propose a layer for estimating the Jaccard index as a detection quality score and a novel EM merging unit, which uses these quality scores to resolve detection overlap ambiguities. Ge *et al*. [4] introduce a variant of two-stage detectors called PS-RCNN that first detects non-occluded objects with RCNN module and then suppresses the detected instances by object-shaped masks so that the features of heavily occluded instances could stand out. At the second step, another RCNN module specialized in heavily occluded objects detects the rest of the objects.

Compared to these works, our iterative scheme is much easier to integrate into the standard deep learning architectures for object detection.

3 Proposed method

The proposed iterative scheme is shown in Figure 2. First, we introduce notation and describe the inference process. Then, we explain the modifications to the training procedure.

Inference process. A typical object detector D is an algorithm that maps image $I \in \mathbb{R}^{w \times h \times 3}$ to a set of bounding boxes $B = \{(x_k, y_k, w_k, h_k)\}_{k=1}^{n}$. Each box is represented by the coordinates of its top left corner (x, y), width w and height h. For a given set of boxes B we define a history image H of the same size as an input image, where each pixel records the number of already detected boxes that cover that pixel:

$$H_{xy} = \sum_{k=1}^{|B|} \mathbb{1}_{\,x_k \le x \le x_k + w_k,\; y_k \le y \le y_k + h_k} \tag{1}$$

Figure 2 shows an example of the history, where its values are color-coded. We can make a detector D' history-sensitive if we pass the history H along with the image I as its inputs.

Let us now introduce the iterative scheme $IterDet(D')$, that given an image I produces a set of bounding boxes B in an iterative manner. At the first iteration $t = 1$ history H_1 is empty and D' maps an image I and H_0 to a set of bounding boxes B_1. Second, B_1 is mapped to history H_2 which, in turn, at iteration $t = 2$ is mapped to B_2 by D'. This process stops when the limit of iterations is reached or when $|B_m| = 0$. The final prediction of $IterDet(D')$ is $B = \bigcup_{i=1}^{m} B_i$.

The described scheme requires two design choices: 1) how to modify an arbitrary detector D to a history-sensitive D' and 2) how to force D' to predict different sets of objects B_t on each iteration t. The detailed explanations are provided below.

Architecture of a history-aware detector. State-of-the-art deep object detection pipelines start with passing an image to an already pretrained backbone, *e.g.* ResNet, HRNet, VGG, etc. Then multilevel features are fed into additional feature extractors, *e.g.* Region Proposal Network, Feature Pyramid Network, etc. Finally, these features are transformed into predicted bounding boxes by a head module followed by non-maximum suppression. We try to introduce minimal changes into these original network architectures and fuse an image with the history in the earliest layers of the network.

Proposed architecture of the history-aware detector is simple yet efficient. The history passed through one convolution layer is then added together with the output of the first convolution layer of the backbone. This scheme can be applied to any backbone. In case of ResNet-like backbone, before adding image is passed through a convolution layer with 64 filters of size 7 and stride 2, Batch Normalization layer and ReLU activation layer. The convolution layer accepting history has 64 filters of size 3 and stride 2.

Training procedure. During training we randomly split the set of ground truth bounding boxes $\hat{B}$ into two subsets B_{old} and B_{new}, such that $B_{old} \cup B_{new} = B$ and $B_{old} \cap B_{new} = \emptyset$. We map B_{old} to a history H and force D' to predict the bounding boxes B_{new} that are missing in history. Thus, we optimize the losses of D' by back propagation of the error between the predicted boxes B and target boxes B_{new}. On the one hand, this method of training forces the model to exploit the history and predict only new objects at each iteration of inference. On the other hand, it provides additional source of augmentations by sampling different combinations of B_{old} and B_{new}.

A few previous works proposed the methods that predicted one object per iteration [1, 19]. Our iterative scheme is also able to predict one object per iteration *e.g.* by selecting the most confident detection. However, in practice such an approach would be inefficient, as the time for processing an image would be proportional to the number of objects in that image. Our experiments in section 4 demonstrate that two iterations is enough to achieve the best accuracy. Increasing the number of iterations improves the recall but tends to lower the precision, resulting in worse mMR and AP metrics.

4 Experiments

4.1 Datasets and implementation details

To validate the effectiveness of our proposed iterative scheme, we conduct experiments on three crowded datasets: AdaptIS ToyV1 and ToyV2 [18], CrowdHuman [17] and WiderPerson [23].

	Toy V1	Toy V2	CrowdHuman	WiderPerson
object/image	14.88	31.25	22.64	29.51
pair/image				
IoU > 0.3	3.67	7.12	9.02	9.21
IoU > 0.4	1.95	3.22	4.89	4.78
IoU > 0.5	0.95	1.25	2.40	2.15
IoU > 0.6	0.38	0.45	1.01	0.81

Table 1: Comparison in terms of the average number of objects and pair-wise overlap between two instances on the four datasets used in our experiments.

AdaptIS. AdaptIS Toy V1 and Toy V2 are two synthetic datasets originally used for instance segmentation task [18]. Available annotation allows using them for object detection. Each image from aforementioned datasets contains about 30 objects on average, with many of those strongly overlapping. The statistics of the datasets are shown in Table 1. Training and validation splits of Toy V1 dataset contain 2000 and 10000 images of size 96×96 pixels, respectively. Toy V2 is split into 3 parts: training, validation, and test with 25000, 1000, and 1000 images of size 128×128 pixels respectively. We have chosen AP as the main metric for Toy datasets. For consistency, we provide the values of recall. We do not report mMR metric, since it has proven unrepresentative at a small number of errors, turning zero in case an average number of false positives per image is less than 1.

CrowdHuman. The recently introduced CrowdHuman dataset is the most complex compared to other human image datasets in terms of both number of persons per image and number of pairs of intersecting bounding boxes with IoU>0.5, according to [17]. It contains 15000, 4370, and 5000 images for training, validation, and testing, respectively. Each image has an average of about 23 people and 3 boxes for each of them: *full body*, *visible body* and *head*. The most challenging and most frequently used in other works is full body annotation, where the boxes not only overlap more strongly, but also go beyond the edges of the image. We also conduct experiments on visible annotation, training models on the training part of the data, and benchmarking metrics on validation.

[17] also provide the metrics of detection quality for two standard detectors. These are the single stage RetinaNet detector and the two stage Faster RCNN detector, both using ResNet-50 as a backbone. In addition to standard metrics - recall and AP (average precision), mMR is proposed as the main metric. mMR denotes the log average missing rate over 9 points ranging from 10^2 to 10^0 FPPI (false positives per image).

WiderPerson. WiderPerson [23] is another dense human detection dataset collected from various sources. It contains five types of annotations âĂŞ pedestrians, riders, partially visible persons, crowd and ignored regions. Following [9], in our experiments we merge the last four types into one category for both training and testing. WiderPerson contains 8000, 1000, and 4382 images in train, validation, and test sets. The annotations for the test part are not publicly available.

Implementation details. Our implementation of the proposed IterDet and all baseline models is based on the MMDetection framework [3]. This framework is implemented on top of the PyTorch deep learning library [13]. It contains implementations of more than a dozen state-of-the-art one- and two-stage detectors and has a modular design that allows easy incorporation of our iterative scheme. For our experiments we use RetinaNet and Faster RCNN implementations based on ResNet-50 with default parameters, including the number of GPUs equal to 8 with 2 images per each. The minor modifications are described below. First, we add a Batch Normalization layer after each convolution layer to the FPN of both detectors, which slightly improves performance. Secondly, we do not freeze the first block of ResNet as we add history together with the trainable convolution layer before this block. To simplify the hyperparameter tuning, Adam optimizer with initial learning rate 0.0001 is used for IterDet experiments. For the baseline experiments, we use SGD optimizer with momentum 0.9, weight decay parameter 0.0001, and initial learning rate 0.02. The training process finishes at the end of the 24th epoch, and the learning rate is decreased by 0.1 after 16th and 22th epochs.

Dataset-specific hyperparameters. To be consistent with CrowdHuman benchmark, during inference the input image is re-scaled such that its shortest edge is 800 pixels, and the longest side is not beyond 1400 pixels. We do not use test-time augmentations. Two augmentations are applied to the images during the training procedure: horizontal flips and varying a size within 25%. We also find that using information about ignored regions when sampling negative examples during training slightly increases accuracy in all CrowdHuman experiments. For experiments with *full* body annotations on CrowdHuman, we also use the same settings of design parameters [9, 11, 17], such as $[1.0,\ 1.5,\ 2.0,\ 2.5,\ 3.0]$ anchor ratios and no clipping proposals. Images from AdaptIS Toy V1 and Toy V2 datasets are upscaled to 384×384 pixels, following the original work [18], since the original sizes of 96×96 and 128×128 pixels are too small for ResNet architecture. In the experiments on WiderPerson dataset we use the same hyperparameters as for CrowdHuman.

Method	Detector	Toy V1		Toy V2	
		Recall	AP	Recall	AP
Baseline	RetinaNet	95.46	94.46	96.27	95.62
IterDet, 1 iter.		95.21	95.31	96.27	94.17
IterDet, 2 iter.		**99.56**	**97.71**	**99.35**	**97.27**
Baseline	Faster RCNN	94.05	93.96	94.88	94.81
IterDet, 1 iter.		94.34	94.27	94.97	94.89
IterDet, 2 iter.		**99.60**	**99.25**	**99.29**	**99.00**

Table 2: Experimental results on AdaptIS Toy V1 and Toy V2 dataset.

4.2 Results and discussion

Results on AdaptIS datasets. Table 2 shows a comparison of IterDet and baseline metrics on AdaptIS Toy V1 and Toy V2 datasets. For both datasets and detectors IterDet substantially increases AP. This increase expands 4% for Faster RCNN bringing the final AP to 99%.

Results on CrowdHuman. Results on full body and visible body annotations of CrowdHuman dataset are presented in Tables 3 and 4 respectively. We compare the proposed IterDet scheme to previously published results that do not use additional data or annotations

during training, on two detectors: RetinaNet and Faster RCNN. Our main result is a significant improvement of all three metrics on the most challenging *full* body annotation, as shown in the last two rows of Table 3. Thus, IterDet improves recall by more than 5.5%, AP - by 3.1% and mMR - by 1.0% compared to baseline. These improvements remain significant even when compared to previous state-of-the-art approaches such as Adaptive NMS and PS-RCNN. By the basic metrics on this benchmark (mMR), IterDet outperforms all existing methods in all four scenarios: single- and two-stage detectors, visible and full body annotations. This gap exceeds 6% for the RetinaNet detector on both types of annotations. It is also worth noting that such an improvement in mMR is achieved even at 1 iteration, indicating the effectiveness of history-sensitive training to regularize an arbitrary detector. Despite a slight degradation of mMR with increasing number of iterations, the growth of AP always remains significant. Thus, we improve over previous best results for RetinaNet by 3.9% AP on both types of annotations.

Method	Detector	Recall	AP	mMR
Baseline [17]	RetinaNet	**93.80**	80.83	63.33
IterDet, 1 iter.		79.68	76.78	**53.03**
IterDet, 2 iter.		91.49	**84.77**	56.21
Baseline [17]	Faster RCNN	90.24	84.95	50.49
Soft NMS [2, 10]		91.73	83.92	51.97
Adaptive NMS [10]		91.27	84.71	49.73
Repulsion Loss [4, 23]		90.74	85.71	-
PS-RCNN [4]		93.77	86.05	-
IterDet, 1 iter.		88.94	84.43	**49.12**
IterDet, 2 iter.		**95.80**	**88.08**	49.44

Table 3: Experimental results on CrowdHuman dataset with *full* body annotations.

Method	Detector	Recall	AP	mMR
Baseline [17]	RetinaNet	**90.96**	77.19	65.47
Feature NMS [16]		-	68.65	75.35
IterDet, 1 iter.		86.91	81.24	**58.78**
IterDet, 2 iter.		89.63	**82.32**	59.19
Baseline [17]	Faster RCNN	91.51	**85.60**	55.94
IterDet, 1 iter.		87.59	83.28	**55.54**
IterDet, 2 iter.		**91.63**	85.33	55.61

Table 4: Experimental results on CrowdHuman dataset with *visible* body annotations.

Results on WiderPerson. Results on WiderPerson dataset are presented in Table 5. We use the results from the baseline work [23] for *hard* subset of annotations, which implies all the boxes larger than 20 pixels in height. Following the protocol from [4], we assign to even more challenging task with using all bounding boxes with no height limits during testing. For both detectors, the proposed iterative scheme significantly outperforms all previous results in terms of recall, AP and mMR.

Choice of the number of iterations. Table 6 shows a comparison of AP for different detectors and datasets with different number of iterations of the proposed iterative scheme. One can observe that after the second iteration there is no increase in AP.

Method	Detector	Recall	AP	mMR
Baseline [23]	RetinaNet	-	-	48.32
IterDet, 1 iter.		90.38	87.17	**43.23**
IterDet, 2 iter.		**95.35**	**90.23**	43.88
Baseline [23]	Faster RCNN	-	-	46.06
Baseline [4]		93.60	88.89	-
PS-RCNN [4]		94.71	89.96	-
IterDet, 1 iter.		92.67	89.49	**40.35**
IterDet, 2 iter.		**97.15**	**91.95**	40.78

Table 5: Experimental results on WiderPerson dataset.

	CrowdHuman		Toy V2	
# iter.	Faster RCNN	RetinaNet	Faster RCNN	RetinaNet
1	84.43	76.78	94.89	95.62
2	**88.08**	**84.47**	**99.00**	**97.27**
3	87.71	84.65	98.96	97.23
4	87.16	83.10	98.96	97.22

Table 6: Comparison of AP for different number of iterations for IterDet based on Faster RCNN or RetinaNet. Full body annotation is used for CrowdHuman.

Figure 3 shows an example of results of IterDet with Faster R-CNN on the four datasets used in our experiments. One can see that in cases when objects significantly overlap, the second iteration indeed helps to recover many occluded objects.

5 Conclusion

We present an iterative scheme (IterDet) for object detection designed for crowded environments. It can be applied to both two-stage and one-stage object detectors. Experiments on challenging AdaptIS ToyV1 and ToyV2 datasets with multiple overlapping objects demonstrate that IterDet is able to achieve almost perfect detection accuracy. Extensive comparison on CrowdHuman and WiderPerson benchmarks shows that proposed scheme achieves higher accuracy compared to existing works when applied to both two-stage Faster RCNN and one-stage RetinaNet detectors.

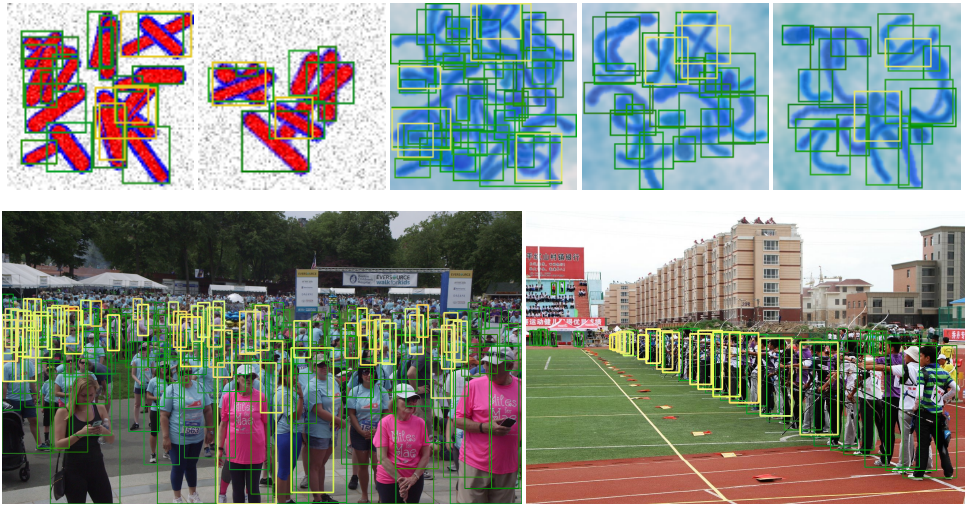


Figure 3: Examples of IterDet results on ToyV1, ToyV2, CrowdHuman (with *full* body annotataions), and WiderPerson. The boxes found on the first and second iterations are marked in green and yellow respectively. The scores thresholded for visualization are above 0.1.

References

[1] Olga Barinova, Victor Lempitsky, and Pushmeet Kholi. On detection of multiple object instances using hough transforms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1773–1784, 2012.

[2] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017.

[3] Kai Chen, Jiaqi Wang, Jiangmiao Pang, Yuhang Cao, Yu Xiong, Xiaoxiao Li, Shuyang Sun, Wansen Feng, Ziwei Liu, Jiarui Xu, Zheng Zhang, Dazhi Cheng, Chenchen Zhu, Tianheng Cheng, Qijie Zhao, Buyu Li, Xin Lu, Rui Zhu, Yue Wu, Jifeng Dai, Jingdong Wang, Jianping Shi, Wanli Ouyang, Chen Change Loy, and Dahua Lin. MMDetection: Open mmlab detection toolbox and benchmark. *arXiv preprint arXiv:1906.07155*, 2019.

[4] Zheng Ge, Zequn Jie, Xin Huang, Rong Xu, and Osamu Yoshie. Ps-rcnn: Detecting secondary human instances in a crowd via primary object suppression. *arXiv preprint arXiv:2003.07080*, 2020.

[5] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[6] Eran Goldman, Roei Herzig, Aviv Eisenschtat, Jacob Goldberger, and Tal Hassner. Precise detection in densely packed scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5227–5236, 2019.

[7] Jan Hosang, Rodrigo Benenson, and Bernt Schiele. Learning non-maximum suppression. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4507–4515, 2017.

[8] Han Hu, Jiayuan Gu, Zheng Zhang, Jifeng Dai, and Yichen Wei. Relation networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3588–3597, 2018.

[9] Xin Huang, Zheng Ge, Zequn Jie, and Osamu Yoshie. Nms by representative region: Towards crowded pedestrian detection by proposal pairing. *arXiv preprint arXiv:2003.12729*, 2020.

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

[11] Songtao Liu, Di Huang, and Yunhong Wang. Adaptive nms: Refining pedestrian detection in a crowd. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6459–6468, 2019.

[12] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *European conference on computer vision*, pages 21–37. Springer, 2016.

[13] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas KÃűpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library, 2019.

[14] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015.

[15] Rasmus Rothe, Matthieu Guillaumin, and Luc Van Gool. Non-maximum suppression for object detection by passing messages between windows. In *Asian conference on computer vision*, pages 290–306. Springer, 2014.

[16] Niels Ole Salscheider. Featurenms: Non-maximum suppression by learning feature embeddings, 2020.

[17] Shuai Shao, Zijian Zhao, Boxun Li, Tete Xiao, Gang Yu, Xiangyu Zhang, and Jian Sun. Crowdhuman: A benchmark for detecting human in a crowd. *arXiv preprint arXiv:1805.00123*, 2018.

[18] Konstantin Sofiiuk, Olga Barinova, and Anton Konushin. Adaptis: Adaptive instance selection network. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7355–7363, 2019.

[19] Russell Stewart, Mykhaylo Andriluka, and Andrew Y Ng. End-to-end people detection in crowded scenes. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2325–2333, 2016.

[20] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9627–9636, 2019.

[21] Lachlan Tychsen-Smith and Lars Petersson. Improving object localization with fitness nms and bounded iou loss. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6877–6885, 2018.

[22] Wang Xinlong, Xiao Tete, Jiang Yuning, Shao Shuai, Sun Jian, and Shen Chunhua. Repulsion loss: Detecting pedestrians in a crowd. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7774–7783, 2018.

[23] Shifeng Zhang, Yiliang Xie, Jun Wan, Hansheng Xia, Stan Z Li, and Guodong Guo. Widerperson: A diverse dataset for dense pedestrian detection in the wild. *IEEE Transactions on Multimedia*, 2019.