An effective object detection algorithm for high resolution video by using convolutional neural network

Denis Vorobjov[1], Iryna Zakharava[1], Rykhard Bohush[1], Sergey Ablameyko[2]

[1] Polotsk State University, Blochin st. 29, Novopolotsk, Republic of Belarus
{d.vorobjov,i.zakharova,r.bogush}@psu.by
[2]Belarusian State University, Nezavisimosti avenue 4, Minsk, Republic of Belarus
ablameyko@bsu.by

Abstract. In this paper, an algorithm to detect small objects more accurately in high resolution video is proposed. For this task, an analysis of state-of-the-art algorithms in application to high resolution video processing, which can be implemented into modern surveillance systems is performed. The algorithm is based on CNN in application to high resolution video processing and it consists of the following steps: each video frame is divided into overlapping blocks; object detection in each block with CNN YOLO is performed; post processing for extracted objects in each block is done and merging neighbor regions with the same class probabilities is performed. The proposed algorithm shows better results in application to small objects detection on high resolution video than famous YOLO algorithm.

Keywords: Convolution Neural Networks, video processing, YOLO, high resolution

1 Introduction

Video surveillance systems are used in fields such as pedestrian detection, robots, autopilots and etc. There are many video cameras which can record high resolution video. Such devices allow identification of object details to improve and increase the small objects detection probability.

In recent years, a big number of digital cameras have offered the ability to record 4K high-resolution video. This notion refers to any digital image containing an X resolution of approximately 4000 pixels. The actual dimensions of the X and Y resolution of a 4K image depends on the required aspect ratio of images. Positive side of using 4K resolution systems is possibility to detect objects very precisely and accurately. Higher resolution can detect tiny objects without distortions, and also shows correct shape of large objects. At the same time 4K resolution disadvantage is a big information capacity in every surveillance video frame. That is why real-time 4K video processing system must have fast and effective image processing algorithms.

It is important to take into account that small objects detection percentage ratio can be varied at different resolutions. As an example, 80×80 pixels object takes in 0.69%

adfa, p. 1, 2011.

of whole image in HDTV resolution, 0.3% for 2K and 0.05% for 4K. As results of these relations can be false negative detection results for small-sized objects.

To avoid described disadvantages, we think that the deep machine learning systems provide perfect performance for all challenges. All object detection systems have to dedicate following contributions: accuracy, precise extraction of regions of interest (RoIs) on frames or images and their classification with minimal deviation and speed, processing information in real-time.

A lot of interest in image processing and machine learning fields at this moment focused on convolutional neural networks (CNNs). Unlike traditional networks, CNNs provide reducing a number of extracting parameters and as an alternative of whole image processing we can process only extracted feature map, which take into account image topology and stable to affine transformation. But small objects on large scale images can being missed because scaling image or frame it is first preprocessing step almost at all image and video processing algorithms, and missing small objects on surveillance video tape can being a big problem.

At this moment, almost all systems do not take into account input image resolution. These systems make scaling by input layer size and in case of high resolution image or video frame small objects can be missed. In case of high resolution videos like 4K, these systems are unsuitable for object detection. We propose a novel algorithm based on separating input frame into overlapping blocks. After that, CNN Yolo makes scaling, object detection and classification to each block. As a result we have bounding boxes are built around every detected object and class proposals for them. Finally we merge neighboring RoIs with the same class keeping in mind overlapping.

2 Related Works

At first, we analyze famous neural network architectures which use convolutional and maxpooling layers. CNN model AlexNet presented in [1] including 8 weighted layers. This CNN model has limitations with multiresolution analysis, and liable to overfitting according to side-supervision problem [2]. The main disadvantage that first network layer scale input image or frame and this lead to false negative small objects detection. Also this algorithm has big computational cost.

Another approach CNN model is R-CNN and it proposed that the previous RoIs extraction can make CNN work faster and more accurately. The main steps of R-CNN algorithm are preliminary extraction of RoIs using Selective Search algorithm, classification of extracted regions with AlexNet from [1], also, at the last stage, binary linear SVM classification model with Non-Maximum suppression to refine bounding boxes around RoIs, which belong to the one class. Comparison with initial AlexNet model shows that the R-CNN algorithm accomplishes better results in detection [3]. But R-CNN takes a long processing time per image and has almost all AlexNet disadvantages too. Trying to make R-CNN faster, in paper [4] it was proposed Fast R-CNN which calculated convolutional features together with Selective Search and after that, each RoI is presented like convolutional feature map. New version is proposed in paper [5] and it is called Faster R-CNN. This network involves Region Proposal Net-

work, which make proposals about RoIs location. This CNN works fast and accurate but low-resolution frames cannot be processed. Faster R-CNN is unsuitable for pedestrian detection on low resolution videos [6]. Also, according to proposed algorithm input image scaled to input layer size and small objects could be missed.

CNN GoogleNet [7] was contained about 100 layers, but fully connected layers were not used at all. Proposed architecture extracted 12 times less parameters than AlexNet and all architecture is used convolutional layers with different sizes. This approach allowed extracting various sized features. Previous RoIs extraction was improved by multibox method from [8]. The main disadvantage is the system did not take into account context image information and has big computational cost. Modified GoogLeNet with Inception v2 was proposed in [9]. With factorization convolutional layer into smaller convolutional kernels, computational speed became higher at 33%. Also, to avoid representational bottlenecks, half of features was gone to maxpooling layer and the next half was gone to the following convolutional layer simultaneously. For GoogLeNet Inception v3 modification was added Batch Normalization layer [10] instead of dropout technology. Inception v2 and Inception v3 took 2.5 computational time more than Inception. Also, all modifications were not taking into account input image resolution.

Proposed CNN topology ResNet in [11] was provided shortcut connection between repeated blocks and it helps to avoid detector degradation problem of deep neural networks. Top-5 error was 3.57%. But image scaling was preprocessing step yet.

Novel block Inception v4 proposed in [12], all CNN GoogLeNet trained on one PC. Also in [12], it is proposed GoogLeNet with Inception-ResNet v1 and Inception-ResNet v2. Proposed modifications do not take into account input image resolution.

Next approach presented in paper [13] and it is called CNN YOLO. YOLO showed the best result by computational cost. As disadvantage, we have to admit many localization errors especially for small objects, because this algorithm scale input image too. At paper [14], it is proposed YOLO v2, YOLO9000, modifications of YOLO. Better segmentation an classification was achieved by: batch normalization from [10]; high resolution classifier; convolutional with anchor boxes; using k-means; direct location prediction; RPN usage; fine-grained features; multi-scale training; novel classification model Darknet-19. Top-5 accuracy was 91.2% but proposed algorithm took a long of computational cost. Also multi-scale training gave better opportunity to detect small objects, but it is not enough for 4K resolution systems.

YOLO favorably differentiates from other systems because it processes entire image without previous RoIs extraction. Also, as said at [13], YOLO is the fast object detection system which works better than Faster R-CNN. This is very important because we have to process a high-resolution video frames, and computational costs is very necessary parameter.

3 The Proposed Algorithm

Applying considered algorithms and methods for high resolution video frames demanding that input image scaling which lead to small-size objects are missing. For this problem evaluation we propose an algorithm that is based on CNN YOLO and

consist of following steps: each video frame is divided into intersecting blocks; object detection in each block is performed with CNN YOLO; postprocessing is performed for extracted objects in each block and merging is done for neighbor regions with the same class probabilities.

Step 1. Frame separating. Input frame I with sizes $H \times W$ is divided into overlapping blocks $C_{i,j}$ with sizes $ch \times cw$, $i = \overline{0, H/ch - 1}$, $j = \overline{0, W/cw - 1}$. Step size is calculated like block size plus 10% overlapping. Overlap size can be vary by input frame resolution or object size.

Step 2. Blocks classification. Each block goes to YOLO. In this network, for block $C_{i,j}$ conditional class probabilities $Pr(Class_l|C_{i,j})$ is calculated for every class $Class_l$, where $l = \overline{0, classNumber - 1}$, *classNumber* - the number of all classification classes. For each block $C_{i,j}$, the RoI $B_{i,j}^k$, $k = \overline{0, bbNumber - 1}$ is declared where *bbNumber* - the number of RoIs for each block. In YOLO RoIs is frame fragment which is equal for every block and have center in the middle of them. Every RoI $B_{i,j}^k$ determines the followings values: $B_{i,j}^k(x, y)$ - the top left corner coordinates relative to the whole frame I; $B_{i,j}^k(w, h)$ - the width and height are predicted relative to the whole frame I; confidence prediction $\Pr\left(B_{i,j}^k\right)$ - detection of object probability. If no object exists in that block, the confidence scores should be zero, $\Pr\left(B_{i,j}^k\right) = 0$.

Step 3. Blocks postprocessing. The neighbor RoIs, which have combined overlapped region located closer than 10% from blocks edge, are searched. If these blocks are found, we calculate *IoU* (Intersection over Union) which describe two regions overlapping:

$$Iou = \frac{In}{B_{i0,\,j0}(w) \cdot B_{i0,\,j0}(h) + B_{i1,\,j1}(w) \cdot B_{i1,\,j1}(h) - In}, \tag{1}$$

where:

$$\begin{aligned} In = &\left(\min\left(B_{i0,\,j0}(x) + B_{i0,\,j0}(w),\, B_{i1,\,j1}(x) + B_{i1,\,j1}(w)\right) - \max\left(B_{i0,\,j0}(x),\, B_{i1,\,j1}(x)\right)\right) \times \\ &\left(\min\left(B_{i0,\,j0}(y) + B_{i0,\,j0}(h),\, B_{i1,\,j1}(y) + B_{i1,\,j1}(h)\right) - \max\left(B_{i0,\,j0}(y),\, B_{i1,\,j1}(y)\right)\right) \end{aligned} \tag{2}$$

where w and h - RoI width and height relative to the whole frame, x,y - represent the top left corner coordinates of the RoI.

If $Iou > T$, then RoIs $B_{i0,j0}$ and $B_{i1,j1}$ are combined, where T - coefficient which set identity degree for merged locations.

Fig. 1 shows generalized algorithm that contains all steps described previously.

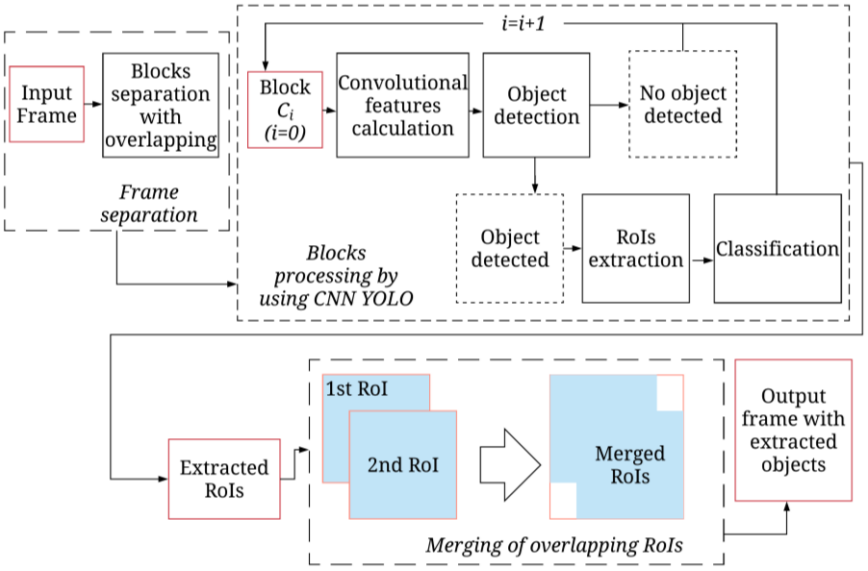

Fig. 1. The proposed algorithm

4 Experimental Results

For the proposed algorithm, training dataset PASCAL VOC [15], grid size *S=7*, numbers of classes for each block *bbNumber=2* and all classification classes number *classNumber=20* were used. The classes names are: «bicycle», «bird», «boat», «bottle», «bus», «car», «cat», «chair», «cow», «dining table», «dog», «horse», «motorbike», «person», «potted plant», «sheep», «sofa», «train», «TV monitor» are used too. For experiments, the proposed algorithm was realized on C++ with GCC compiler and computer vision library OpenCV 3.1.

Table 1 shows that with increasing input frame resolution the proposed algorithm works more accurately than YOLO.

Table 1. Proposed algorithm and YOLO experimental results

Algorithm	Accuracy, δ_E, %			Time per frame, s		
	1024×1024	2048×2048	4096×3072	1024×1024	2048×2048	4096×3072
YOLO	57	48	31	0.11		
Proposed algorithm	49	47	42	0.46	1.72	2.64

Fig. 2 shows RoIs localization for *3872×2592* resolution frame. If object parts are shift we can see that one object is separated into two parts, but accuracy that can be escalated by *bbNumber* value increasing.

Fig. 3 shows results after merging overlapped regions from Fig. 2.

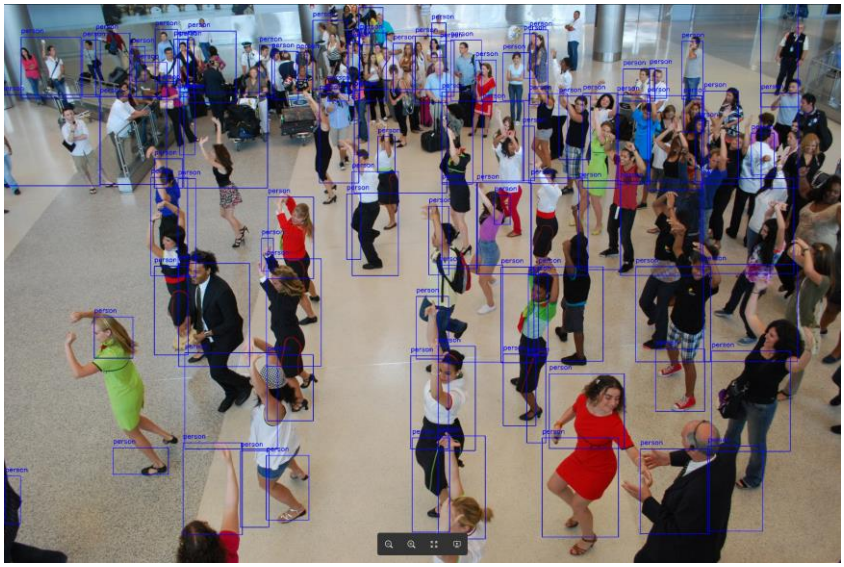

Fig. 2. Extracted RoIs before merging

Fig. 3. An example of merged RoIs

For localization error calculation is used:

$$\delta_E = \sqrt{\frac{\sum_{i=1}^{N}\left(X_{itr} - X_i\right)}{N}} \cdot 100\% \tag{3}$$

where i - frame index, N - objects on image number, X_{itr} - ground-truth coordinates, X_i - after processing coordinates.

Examples of object detection based on our algorithm and YOLO shown at Fig.4.

Experiments were conducted on PC with main characteristics: CPU i7 4.3GHz, RAM 32 Gb. and GPU Nvidia GeForce GTX 1070. For experimental results showed at Fig. 4, video sequences with complex background and different range foreground objects numbers were used. In our approach, blocks were processed serially and it

leads to bigger computational cost than YOLO but, in case of parallel processing, our algorithm can show results comparable to CNN YOLO.

Fig. 4. Examples of object detection: a,c) for YOLO; b,d) for the proposed algorithm

5 Conclusion

We proposed objects detection algorithm for which main application is high-resolution surveillance video processing. The proposed algorithm more accurately detects small objects in high-resolution video. It is achieved by a fact that input frames are not scaled before frame division on blocks and also blocks overlapping gives higher precision. Block by block processing allows avoiding big computational cost. With our algorithm we achieve 35% higher results for 4K video processing than YOLO algorithm. For PC with main parameters CPUi7 4.3 GHz, RAM 32 Gb, GPU Nvidia GeForce GTX 1070 one frame with 4K resolution was processed at 2.64 s.

References

1. Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton: ImageNet classification with deep convolutional neural networks. In: K. Pereira, C. J. C. Burgos, L. Bottou, and K. Q. Wein-

berger (eds.), Advances in Neural Information Processing Systems 25, pp. 1097-1105. Curran Associates, Inc., New York (2012). doi: 10.1007/978-3-319-46654-5_20.
2. Han, X., Zhong, Y., Cao, L., and Zhang, L.: Pre-Trained AlexNet Architecture with Pyramid Pooling and Supervision. In: Remote Sensing, 9, 848. MDPI AG, Basel (2017). doi: 10.3390/rs9080848.
3. R-CNN for Object Detection: https://courses.cs.washington.edu/courses/cse590v/14au/cse590v_wk1_rcnn.pdf
4. Ross Girshick: Fast R-CNN. In: Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), pp. 1440-1448. IEEE Press, Washington DC (2015). doi: 10.1109/ICCV.2015.169
5. Shaoqing Ren, Kaiming He, Ross B. Girshick, Jian Sun: Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. In: ArXiv e-prints (2015). arXiv: 1506.01497
6. Zhang L., Lin L., Liang X., He K: Is Faster R-CNN Doing Well for Pedestrian Detection? In: Leibe B., Matas J., Sebe N., Welling M. (eds) Computer Vision – ECCV 2016. ECCV (2016). Lecture Notes in Computer Science, vol 9906. Springer, Cham (2016). doi: 10.1007/978-3-319-46475-6
7. Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E. Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich: Going deeper with convolutions. In: IEEE Conference on Computer Vision and Pattern Recognition pp. 1–9. IEEE Press, Washington DC (2015). doi: 10.1109/CVPR. 2015.7298594.
8. Dumitru Erhan, Christian Szegedy, Alexander Toshev, Dragomir Anguelov: Scalable object Detection Using Deep Neural Networks. In: CVPR '14 Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition pp. 2155-2162. IEEE Press, Washington DC (2014). doi: 10.1109/CVPR.2014.276
9. Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna: Rethinking the inception architecture for computer vision. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, pp. 2818–2826. IEEE Press, Washington DC (2016). doi: 10.1109/CVPR.2016.308.
10. Sergey Ioffe, Christian Szegedy: Batch normalization: accelerating deep network training by reducing internal covariate shift. In: Proceedings of the 32nd International Conference on Machine Learning pp. 448–456. Microtome Publishing, Brookline (2015).
11. Kaiming He, Xiangyu Zhang, Shaoqing Ren, Jian Sun: Deep Residual Learning for Image Recognition. In: Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference. IEEE Press, Washington DC (2016). doi: 10.1109/CVPR.2016.90
12. Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Alex Alemi: Inception v4, Inception-ResNet and the impact of residual connections on learning. In: The Association for the Advancement of Artificial Intelligence Conference on Artificial Intelligence (AAAI), pp. 4278–4284. AAAI Press, Washington DC (2017)
13. Joseph Redmon, Santosh Kumar Divvala, Ross B. Girshick, Ali Farhadi: You Only Look Once: Unified, Real-Time Object Detection. In: IEEE conference Computer Vision and Pattern Recognition (CVPR). IEEE Press Washington DC (2017). doi: 10.1109/CVPR.2016.91
14. Joseph Redmon and Ali Farhadi: YOLO9000: Better, Faster, Stronger. In: 2017 IEEE Conference on Computer Vision and Pattern Recognition pp. 6517–6525. IEEE Press, Washington DC (2017). doi:10.1109/CVPR.2017.690
15. Mark Everingham, Luc J. Van Gool, Christopher K. I. Williams, John M. Winn, Andrew Zisserman: The Pascal Visual Object Classes (VOC) Challenge. In: International Journal of Computer Vision 88(2) pp. 303-338. Springer, New York (2010)

An Effective Object Detection Algorithm for High Resolution Video by Using Convolutional Neural Network

Chapter · May 2018

DOI: 10.1007/978-3-319-92537-0_58

CITATIONS
9

READS
774

4 authors, including:

Iryna Zakharava
Polotsk State University
6 PUBLICATIONS **37** CITATIONS

SEE PROFILE

Rykhard Bohush
Polotsk State University
61 PUBLICATIONS **160** CITATIONS

SEE PROFILE

Sergey Ablameyko
Belarusian State University
161 PUBLICATIONS **585** CITATIONS

SEE PROFILE