A novel squeeze YOLO-based real-time people counting approach

Peiming Ren, Lin Wang, Wei Fang* and Shulin Song

Department of Computer Science and Technology,
School of IoT Engineering,
Jiangnan University,
Wuxi, Jiangsu, China
Email: peimingren@163.com
Email: 2546581638@qq.com
Email: fangwei@jiangnan.edu.cn
Email: songshulin@gmail.com
*Corresponding author

Soufiene Djahel

School of Computing, Mathematics and Digital Technologies,
Manchester Metropolitan University,
Manchester, UK
Email: s.djahel@mmu.ac.uk

Abstract: Real-time people counting based on videos is one of the most popular projects in the construction of smart cities. To develop an accurate people counting approach, deep learning can be used as it greatly improves the accuracy of machine learning-based approaches. To this end, we have previously proposed an accurate you only look once (YOLO)-based people counting approach, dubbed YOLO-PC. However, the model of YOLO-PC was very large with an excessive number of parameters, thus it requires large storage space on the device and makes transmission on internet a time consuming task. In this paper, a new real-time people counting method named as squeeze YOLO-based people counting (S-YOLO-PC) is proposed. S-YOLO-PC uses the fire layer of SqueezeNet to optimise the network structure, which reduces the number of parameters used in the model without decreasing its accuracy. Based on the obtained the experimental results, S-YOLO-PC reduces the number of model parameters by 11.5% and 9% compared to YOLO and YOLO-PC, respectively. S-YOLO-PC can also detect and count people with 41 frames per second (FPS) with the average precision (AP) of person of 72%.

Keywords: model compression; people counting; boundary-selection; you only look once; YOLO; SqueezeNet.

Reference to this paper should be made as follows: Ren, P., Wang, L., Fang, W., Song, S. and Djahel, S. (2020) 'A novel squeeze YOLO-based real-time people counting approach', *Int. J. Bio-Inspired Computation*, Vol. 16, No. 2, pp.94–101.

Biographical notes: Peiming Ren is a third year postgraduate student majoring in Computer Science and Technology at Jiangnan University. His research interests include issues related to artificial intelligence and pattern recognition, computer vision and machine learning.

Lin Wang is a second year postgraduate student majoring in Computer Science and Technology at Jiangnan University. Her research interests are related to artificial intelligence and pattern recognition, computer vision and machine learning.

Wei Fang works at Jiangnan University as an Associate Professor and a master. He obtained his Doctorate in Light Industry Information Technology and Engineering of Jiangnan University in March 2008. From April 2013 to April 2014, he went to the University of Birmingham, Professor Xin Yao's Research Group (CERCIA) for a one-year academic study. His research focuses on the particle swarm optimisation (PSO) algorithm and the quantum-behaved PSO (QPSO) algorithm proposed by the research group. He has had 24 papers published in authoritative journals at home and abroad. He is currently a member of IEEE Computer Science Society, IEEE Computational Intelligence Society, China Computer Society and so on.

Shulin Song works at Jiangnan University as a Teacher. His research focuses on pattern recognition and computer vision.

Soufiene Djahel currently works at the School of Computing, Mathematics and Digital Technology, Manchester Metropolitan University. His main research interests include intelligent transportation systems (ITS), wireless networking, networks security and e-health.

This paper is a revised and expanded version of a paper entitled 'A novel YOLO-based real-time people counting approach' presented at 2017 International Smart Cities Conference (ISC2), Wuxi, 14–17 September 2017.

1 Introduction

People detection and counting are key components in intelligent security and intelligent building. One of the main technologies for people detection and counting is based on image classification and object detection. However, the complex background of videos makes it difficult to distinguish people from other objects and the background in the monitoring screen. Recently, deep learning with deep convolutional neural network has been applied in image classification and object detection and achieved high accuracy. Robot control, autonomous driving and other related tasks which highly rely on low-latency systems not only require high accuracy, but also require high speed. In recent years, R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015), and Faster R-CNN (Ren et al., 2015) have become representative approaches in the field of object detection. Redmon et al. (2016) proposed you only look once (YOLO), which is one of the best real-time approaches.

Based on Redmon et al. (2016), which is the neural network framework of YOLO, YOLO-based people counting (YOLO-PC) (Ren et al., 2017) has realised real-time people detection with an accuracy of 72.8%. YOLO-PC has improved the performance of people counting, making people counting more pertinent. YOLO-PC increased the number of cells from 7×7 to 9×9. The number of bounding box of each cell was increased to 3. In addition, the boundary-selection method was integrated in order to make the detection and counting more accurate. However, the number of parameters of YOLO-PC is still large even if it only detected one class. In order to further decrease the number of parameters in YOLO-PC, Squeeze YOLO-based people counting (S-YOLO-PC) is proposed in this paper. Three fire layers are introduced to replace three convolution layers. The number of filters of the compression part in fire layers is further reduced. The experimental results show that the number of model parameters of S-YOLO-PC is lower than that of YOLO and YOLO-PC and the average precision of person of S-YOLO-PC is still high.

The structure of this paper is as follows. Related works on object detection, people counting and model compression are introduced in Section 2. In Section 3, S-YOLO-PC is presented including the approach to compress the model by introducing fire layers and the steps of the overall process of people counting. In Section 4, experimental results and discussions are given, where the average precision, the number of model parameters and real time performance are compared. Finally, the conclusions and planned future work are presented in Section 5.

2 Related work

2.1 Object detection and people counting

There are mainly three types of methods to count the number of people in a video. The first type uses a statistical method to estimate the number of people in a region. This method usually associates the pixel or other features of the moving area and the number of people in the area, and then trains a function to estimate the number of people. Lee et al. (2007) and Kim et al. (2008) used the underlying features such as foreground pixels statistics and motion vector for people counting. The number of pixels was associated with the number of people and the motion vector was used to distinguish the directions. In these papers, a simple feature was used to calculate the kernel function and the effect was gratifying. However, the way to calculate the kernel function was radically relevant to the specific scene. The second type of methods consists in combining object detection with object tracking. This type usually has a preprocessing procedure. During this procedure, the moving area is extracted and the people are detected in the moving area. Common object detection methods included the segmentation method based on projection histogram proposed by Zhang and Chen (2007) and Ma et al. (2008), the method based on template matching proposed by Hsieh et al. (2007) and the method based on statistical classification proposed by Li et al. (2008). Bat algorithm is a novel population-based evolutionary (Cai et al., 2018) and bio-inspired stochastic optimisation algorithm (Cai et al., 2016; Cui et al., 2017) and cuckoo search is a recently developed meta-heuristic (Zhang et al., 2018). These algorithms have also been applied in object tracking (Gao et al., 2016; Kang et al., 2018). Due to their high computation overhead and long processing time, these methods are very difficult to be deployed in real-time systems. The third type of methods uses the feature point tracking to obtain the path information of the points, and then performs cluster analysis of the feature point path information to realise people counting. Rabaud and Belongie (2006) proposed a simple means for spatially and temporally conditioning the trajectories. Given this representation, they integrated it with a learned object descriptor to achieve a segmentation of the constituent motions.

In the field of people counting, convolutional neural network (CNN) can automatically learn the high-level semantic features of people through a series of convolution operations and constantly correct feature parameters by back propagation. One of the representative works was proposed by Zhang et al. (2015), which used CNN to automatically extract the population density feature in images to estimate the people number in different scenarios. It performed end-to-end training and detection without the tedious steps such as foreground segmentation, artificial design and feature extraction.

Object detection based on deep learning mainly includes two types of methods. The first type consists of two-stage detection algorithm that divides the problem into two stages:

1 generating region proposals

2 classifying the candidate regions.

The representative of such algorithms was R-CNN (Girshick et al., 2014), Fast R-CNN (Girshick, 2015), and Faster R-CNN (Ren et al., 2015). In contrast to these approaches, the second type of methods is one-stage detection algorithm which uses the CNN network to calculate only once. Redmon et al. (2016) proposed YOLO, which is the pioneering work of this type of methods. Subsequently, SSD (Liu et al., 2016), Mask R-CNN (He et al., 2017) and RetinaNet (Lin et al., 2017) were proposed. YOLO transformed the object detection problem into a regression problem, which directly returned the position of the boundary box and the category probability in the output layer. Since the whole process of YOLO uses the same network, it is easy to implement the end-to-end detection. Yolo has been the subject of several improvements, YOLO9000 (Redmon and Farhadi, 2017) and Yolov3 (Redmon and Farhadi, 2018) were proposed to seek extremely fast and accurate object detection algorithms. With its high accuracy and high speed, YOLO is more suitable for real-time applications than other algorithms.

2.2 Model compression

There is a great deal of redundancy with the parameters trained by neural networks, thus the network model should be simplified. Smaller CNN models have several advantages such as smaller server communication needs, less parameters, smaller amount of data downloaded from the cloud.

There are two types of methods for model compression. The first type mainly included pruning, sharing the weights, quantisation, neural network binarisation, etc. These methods are used to make modification on the trained model, and then the model is fine-tuned to the original accuracy. It is not easy to restore the accuracy at the beginning in these methods. The other type of methods consists in designing a network based on the new way of convolution computation. MobileNet (Howard et al., 2017) and SqueezeNet (Iandola et al., 2016) are representative works. The SqueezeNet proposed by UC Berkeley and Stanford researchers is similar to MoibleNet. The classification accuracy of SqueezeNet on ImageNet is close to that of AlexNet (Krizhevsky et al., 2012), and the model size of SqueezeNet has been reduced by nearly 500 times compared to that of AlexNet. Apte et al. (2017) investigated the effect of a fire layer on the classification performance. While this layer did increase the speed of classification, the decrease in accuracy was suitable for use on mobile devices. S-YOLO-PC uses fire layer more reasonably to maintain high accuracy for people detection and counting.

3 The proposed S-YOLO-PC approach

3.1 Overview of S-YOLO-PC

YOLO-PC divided an image into 9×9 cells, each of which was tested separately. More cells lead to more detection boxes and higher confidence. YOLO-PC uses the cell to define the boundary, chooses different boundary according to the actual scene for people counting and ignores irrelevant and uninterested areas, making the people detection more pertinent and improving the counting accuracy.

The model of YOLO is very complicated, and the structure of YOLO-PC is more segmented, resulting in the increase of the number of parameters and model storage space. Thus, the cost of model transmission and storage on Internet is tremendous. S-YOLO-PC is an improved version of YOLO-PC. S-YOLO-PC uses the fire layer to replace the traditional convolution layer to improve the structure. In this paper, the fire layer uses the 1×1 convolution kernel to replace the usual 3×3 convolution kernel. The fire layer decreases the number of parameters with fewer filters, so the model size is reduced without decreasing the accuracy. The new model decreases the number of parameters by 11.5% compared to YOLO, and 9% compared to YOLO-PC. With the support of NVIDIA Titan X, S-YOLO-PC can detect and count people in a continuous high definition video with 41 fps with AP of person of 72%. This is a good real-time performance compared to the 30 fps played by the video player.

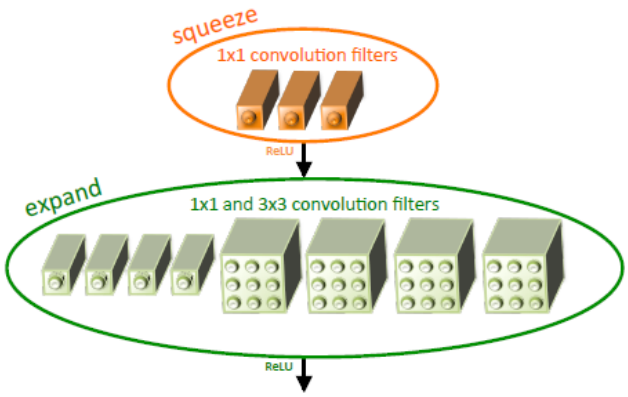
3.2 Introducing the fire layer to compress model

Weights trained by deep neural network tend to occupy large storage space due to the large number of parameters. It has become necessary to reduce the number of parameters of the model. This subsection presents the potential methods to reduce the number of parameters of YOLO-PC by using the fire module of SqueezeNet. It is necessary to optimise the network structure reasonably by reducing the number of parameters instead of just deleting the convolution layers blindly.

The concept of fire module is proposed in SqueezeNet. It significantly reduces the number of parameters and realises the purpose of model compression. As shown in

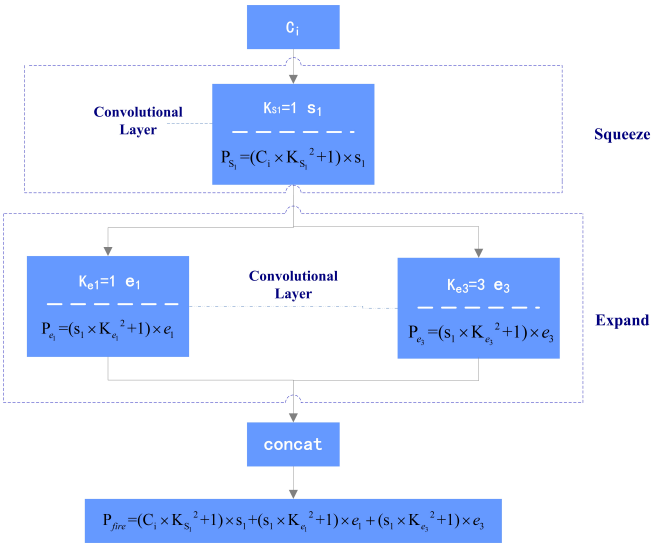
Figure 1, the fire module has three convolution layers. The fire module is divided into the squeeze part and the expand part to compress and expand the data respectively. The squeeze part only uses the 1×1 convolution kernels to replace the usual 3×3 convolution kernels. In the expand section, both a convolution layer with 1×1 filters and a layer with 3×3 filters are used. Then, the outputs of these layers are combined by the concatenation layer.

Figure 1 Fire layer (SqueezeNet) (see online version for colours)



Source: Iandola et al. (2016)

Figure 2 The structure and specific parameters of the fire module (see online version for colours)



Through calculation, we can understand more intuitively that the introduction of fire module can significantly reduce the number of parameters. For example, for a convolution layer, the number of the input channels is c_i, the kernel size is k, and the number of the output channels is c_o. Then, the number of parameters of the convolution layer is calculated as outlined in equation (1). For the fire module, the overall number of input channels is c_i. In the squeeze part, the kernel size of the squeeze part is k_{s_1}, and the number of the output channels is s_1. When k_{s_1} is 1, a large number of parameters of the squeeze part can be reduced. Maintaining the setting of $s_1 < e_1 + e_3$ at the same time, the number of the input channels in the expand part is reduced greatly. For the expand part, the number of input channels are both s_1, the kernel size is k_{e_1} and k_{e_3} respectively, and the numbers of output channels are e_1 and e_3 respectively. The number of parameters is calculated as shown in equation (2). The structure and specific parameters of the fire module are shown in Figure 2.

$$P_{conv} = (c_i * k^2 + 1) * c_o \tag{1}$$

$$P_{fire} = (c_i * k_{s_1}^2 + 1) * s_1 + (s_1 * k_{e_1}^2 + 1) * e_1 + (s_1 * k_{e_3}^2 + 1) * e_3 \tag{2}$$

The network structure of YOLO-PC has 24 convolution layers. S-YOLO-PC replaces the sixteenth, eighteenth and twenty-fourth convolution layers with three fire modules. The number of input channels c_i is not limited, but more input channels can result in higher reduction of the number of parameters. As shown in Table 1, the convolution layer of 1,024 input channels is replaced by the fire layers, the number of parameters can be reduced by 12 times compared to 6.5 times of 512 input channels. Replacing the convolution layer of 1,024 input channels leads to the loss of many parameters. For this reason, the sixteenth and eighteenth layers, which input channels are 512, are replaced with the fire layers in S-YOLO-PC to ensure accuracy.

Table 1 Comparison of the number of parameters number with different s_1 and c_i

Layer type	*Parameter* (c_i *512)*	*Parameter* (c_i *1,024)*
Convolutional	4,719,616	9,438,208
Fire (s_1 = 128)	722,048	787,584
Fire (s_1 = 96)	541,792	590,944

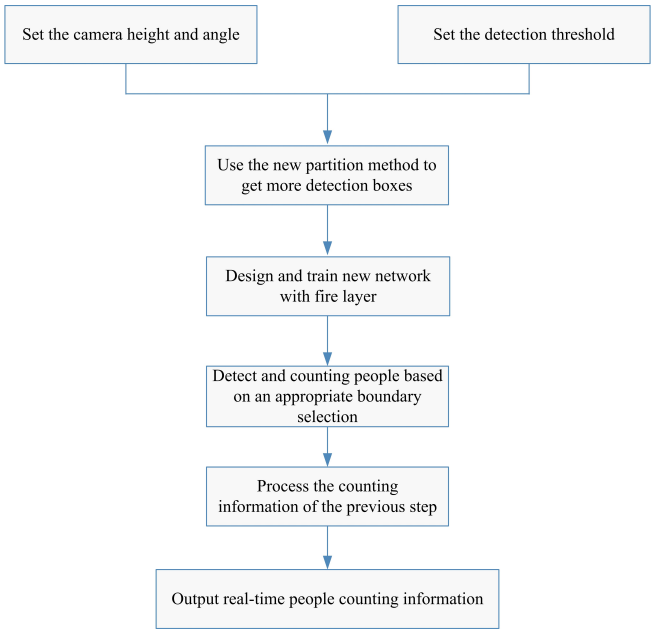
The output channel number of fire layer is e_1+e_3. In SqueezeNet, e_1=e_3=4s_1. We further reduce the output channel number s_1 of the Squeeze part, e_1=e_3>4s_1. Thus, the number of input channels in the expand section is reduced, and the number of the parameters of 1×1 and 3×3 convolution layer of the expand part will be reduced too. The model size will be further reduced as well. The number of s_1 is reduced from 128 to 96 and the number of parameters is decreased heavily. As shown in Table 1, it is clear that the number of parameters, when fire layers are introduced, has been reduced by 25% for both 512 and 1,024 inputs.

3.3 The propsoed people counting approach based on YOLO

As shown in Figure 3, S-YOLO-PC includes six steps. The first step consists in setting the threshold of detection and adjusting the angle and height of the camera. The threshold is usually between 0.2 and 0.4. If the confidence value of detected people is below the threshold, the people will not be counted. In this paper, the default threshold value of 0.2 is used for the sake of simplicity.

In the second step, new partition method is introduced to generate more bounding boxes of people. The algorithm is designed to be more efficient in identifying people with high accuracy. YOLO divided the image into a 7×7 grid and two bounding boxes were predicted for each grid cell. Each bounding box included the coordinate information of box and the confidence of the people. By performing a set of experiments, we found that people counting works better with more detected boxes and higher confidence values. As a result, S-YOLO-PC divides the image into 9×9 grid instead of 7×7 and detects three bounding boxes.

Figure 3 Process of people counting (see online version for colours)



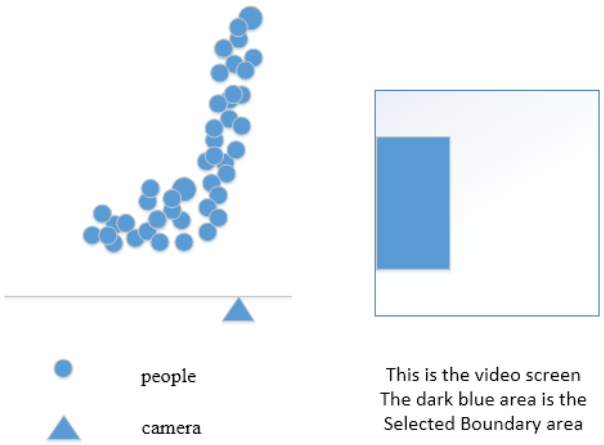
In the third step, new network structure combined with the fire layer is designed and trained. The second step is to make more partitions, which results in an increase in the number of model parameters. As introduced in Section 3.2, the combination of fire layer can produce a small model with high accuracy. The new network structure is composed of 21 convolutional layers and three fire layers, and the new model of people detection is trained on the data sets of VOC-2007 and VOC-2012.

The fourth step aims to detect and count people based on an appropriate boundary selection. S-YOLO-PC selects one or more grid cells from 243 (9×9×3) cells and chooses a boundary according to the actual situation. If people turn left by somewhere, the boundary of the left area of the video is selected. The location of the boundary is about the location of the 113th cell. People who pass through the boundary will be counted. Figure 2 shows the camera position and selected boundary area. Similarly, if people turn right by somewhere, the location of the boundary is about the location of the 129th cell. If people go straight by somewhere, the location of the boundary is about the location of the 121st cell. S-YOLO-PC can be more pertinent and accurate in detecting the flow of people because some irrelevant interference can be avoided. For example, if people are in the billboards and unrelated backs, they can be ignored because they are not in the selected boundary.

The information of the fourth step will be processed in the fifth step. In the selected boundary area, the boxes number accumulates and constantly updates as time goes on, we define the number of boxes as S. The value of S at the moment t represents the total number of detected people from the beginning until time t. Since it takes time for people to move in the boundary area, the same person may be detected repeatedly. According to the statistic result, every person has been detected around 18 times when passing through the selected boundary area, so the predicted number is $S/18$ at the default threshold.

In the last step, the counting information is output. S-YOLO-PC displays the current counting number, FPS and confidence value in the video. S-YOLO-PC can also save real-time information and continue to update the information, and then output them through some interfaces.

Figure 4 Left boundary selection sketch map (see online version for colours)



4 Experimental results

In this section, the average precision, the number of the model parameters of YOLO, YOLO-PC and S-YOLO-PC are compared. Compared to the structure mentioned in Redmon et al. (2016), the previous version mentioned is called YOLO-OLD. YOLO-PC is proposed based on the YOLO-OLD and S-YOLO-PC is proposed based on the updated YOLOV1 version and YOLO-PC.

4.1 Dataset

The training and testing dataset come from the pattern analysis, statistical modelling and computational learning visual object classes project, which is abbreviated as PASCAL VOC. The dataset includes VOC 2007 and VOC 2012. There are 20 types of objects, which are for image classification, object detection and image segmentation. In

this paper, 5,011 images of VOC 2007 and 11,540 images of VOC 2012 are used for training to detect people. 4,952 images of VOC 2007 are chosen as the test set.

4.2 Performance evaluation

The accuracy rate mainly depends on the accuracy of the people detection and the accuracy of counting. People detection is a type of object detection problem and the reference point of measuring accuracy in object detection is the mean average precision (mAP). In object detection, a curve can be drawn based on recall and precision for each category. AP is the area under the curve, and mAP is the average of multiple categories of AP. S-YOLO-PC is designed for people counting, so it is necessary to calculate the AP value of people detection as the standard of measuring accuracy. To get the average precision of person, we refer to the equation of Apte et al. (2017) and R-CNN mAP evaluation script (Girshick, 2016).

$$AP = \sum_{i=1}^{n} P(i)\Delta r(i) \tag{3}$$

$P(i)$ refers to precision at a given threshold i and $\Delta r(i)$ refers to the change in recall between k and $k-1$. This corresponds to the area under the precision recall curve. AP value of several methods is shown in Table 2, the AP of people of S-YOLO-PC is 72.0%, which is 6.6% higher than YOLO and 9.1% higher than YOLO-OLD.

Table 2 AP values of four compared approaches

Methods	*Train set*	*Test set*	*AP of person(%)*
YOLO-OLD	2007+2012	2007	62.9
YOLO	2007+2012	2007	65.4
YOLO-PC	2007+2012	2007	72.8
S-YOLO-PC	2007+2012	2007	72.0

In terms of counting, more detection boxes and higher confidence are obtained for counting, and combining boundary selection achieves better counting results. To this end, S-YOLO-PC uses 9×9 grid and three bounding boxes. We set up three sets of experiments of a 4 min video by using different thresholds (i.e., 0.2, 0.3 and 0.4). The obtained results, shown in Table 3, show that S-YOLO-PC detects more boxes and achieves higher confidence values for those boxes compared to YOLO and YOLO-OLD.

Improvement of more partitions results in an increase in parameters and storage space for the model. S-YOLO-PC makes improvement according to this problem. The method of calculating parameters number is given in Section 3.2, and the percentage of reduced storage space is used to compare the effect of model compression. The change of parameter numbers of the three fire layers is shown in Table 4. The number of parameters is reduced by 17,202,912 on the original basis, and the model size is reduced by 36 % compared with YOLO-OLD, 11.5% compared with YOLO, and 9% compared with YOLO-PC.

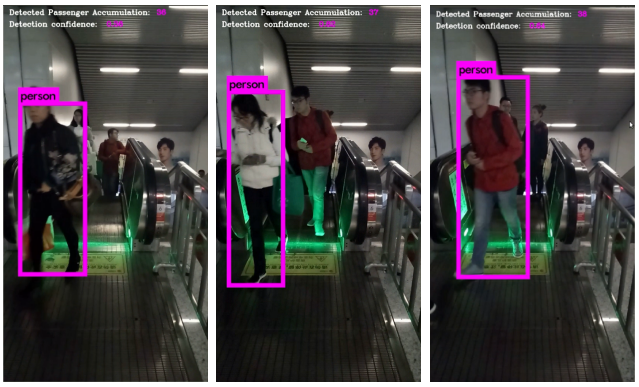
The reference point of measuring real time performance is FPS. The reference point depends on the resolution of the test video or the image. In order to be more practical, this paper uses 1,280×720 continuous high definition videos for testing. With the support of NVIDIA Titan X, S-YOLO-PC can detect and count people with 41 fps. This is a good real time performance compared to the 30 fps shown in video.

Table 3 The number of detected boxes and the average confidence values at different threshold values (T)

Method	*Boxes number*			*Average confidence value*		
	T=0.2	*T=0.3*	*T=0.4*	*T=0.2*	*T=0.3*	*T=0.4*
YOLO-OLD	13,400	9,423	5,856	0.39	0.45	0.51
YOLO	11,908	7,642	4,320	0.37	0.44	0.51
YOLO-PC	14,664	13,612	12,514	0.59	0.62	0.64
S-YOLO-PC	14,559	12,954	10,402	0.50	0.53	0.58

Figure 5 shows the experimental screen in the real escalator scene, the real-time accumulative number of people is displayed at the top of the screen. S-YOLO-PC can count people not only at escalators, but also at the entrance or exit of some similar places, such as scenic spots, etc. The three pedestrians passing through in Figure 5 are accurately detected with bounding boxes and accurately counted. S-YOLO-PC uses fire layers to reduce a large number of redundant parameters and improve the network structure. Smaller and faster models are more suitable for practical applications. In summary, S-YOLO-PC can quickly count people with high accuracy with a smaller model in real-time.

Figure 5 The experimental results of the left boundary selection, (a) the 36th person, (b) the 37th person (c) the 38th person (see online version for colours)



(a) (b) (c)

Table 4 Differences of the number of parameters

Layer name	*Output size*	*Filter size/strid (conv layer)*	*Depth*	*s1 (1×1 squeeze)*	*e1 (1×1 expand)*	*e3 (3×3 expand)*	*Parameter (conv layer)*	*Parameter (fire layer)*
Convolutional	28x28x512	1x1/1	1					
Fire1	28×28×1,024		2	96	512	512	4,719,616	541,792
Ma×pool	14×14×1,024		1					
Convolutional	28×28×512	1×1/1	1					
Fire2	14×14×1,024		2	96	512	512	4,719,616	541,792
Convolutional	14×14×1,024	3×3/1	1					
Convolutional	14×14×1,024	3×3/1	1					
Convolutional	7×7×1,024	3×3/2	1					
Convolutional	7×7×1,024	3×3/1	1					
Fire3	7×7×1,024		2	96	512	512	9,438,208	590,944

5 Conclusions and future work

In this paper, we have introduced a compressed real-time people counting approach named S-YOLO-PC. S-YOLO-PC improves the original convolutional structure of YOLO, and uses the optimised fire layer to replace the 3×3 convolutional layer and the compressed model with fewer parameters is obtained through training. Through more divisions of cells, S-YOLO-PC achieves more bounding boxes and higher detection confidence. Combined with a boundary selection method, people counting becomes more pertinent with higher detection and counting accuracy. In summary, S-YOLO-PC reduces the number of required parameters and compresses the models while achieving high accuracy and real-time performance, making it more suitable for practical application scenarios. As a future work, we plan to undertake further research on designing low power objects detection algorithms with high accuracy and high real-time performance.

Acknowledgements

This work was partially supported by the National Key R&D Program of China (Project Nos. 2017YFC1601800, 2017YFC1601000), National Natural Science foundation of China (Grant No. 61673194), Key Research and Development Program of Jiangsu Province, China (Grant No. BE2017630), the Postdoctoral Science Foundation of China (Grant No. 2014M560390).

References

Apte, M., Mangat, S. and Sekhar, P. (2017) *YOLO Net on iOS*, Technical report, Standford University, Standford [online] http://101.96.10.75/cs231n.stanford.edu/reports/2017/pdfs/135.pdf.

Cai, X., Gao, X-Z. and Xue, Y. (2016) 'Improved bat algorithm with optimal forage strategy and random disturbance strategy', *International Journal of Bio-inspired Computation*, Vol. 8, No. 4, pp.205–214.

Cai, X., Wang, H., Cui, Z., Cai, J., Xue, Y. and Wang, L. (2018) 'Bat algorithm with triangle-flipping strategy for numerical optimization', *International Journal of Machine Learning and Cybernetics*, Vol. 9, No. 2, pp.199–215.

Cui, Z., Cao, Y., Cai, X., Cai, J. and Chen, J. (2017) 'Optimal LEACH protocol with modified bat algorithm for big data sensing systems in internet of things', *Journal of Parallel and Distributed Computing – 2017*, DOI:10.1016/j.jpdc.2017.12.014.

Gao, M-L. et al. (2016) 'A novel visual tracking method using bat algorithm', *Neurocomputing*, Vol. 177, pp.612–619.

Girshick, R. (2015) 'Fast R-CNN', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.1440–1448.

Girshick, R. (2016) *Faster R-CNN* [online] https://github.com/rbgirshick/py-faster-rcnn/blob/master/lib/datasets (accessed 23 February 2016).

Girshick, R., Donahue, J., Darrell, T. and Malik, J. (2014) 'Rich feature hierarchies for accurate object detection and semantic segmentation', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp.580–587.

He, K., Gkioxari, G., Dollár, P. et al. (2017) 'Mask R-CNN', *2017 IEEE International Conference on Computer Vision (ICCV)*, pp.2980–2988, IEEE.

Howard, A.G., Zhu, M., Chen, B. et al. (2017) *MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications* [online] https://arxiv.org/abs/1704.04861 (accessed 17 April 2017).

Hsieh, J.W., Peng, C. and Fan, K.C. (2008) 'Grid-based template matching for people counting', *Proc. of IEEE 9th Workshop on Multimedia Signal Processing*, pp.316–319, IEEE Computer Society, Washington DC, USA.

Iandola, F.N., Han, S., Moskewicz, M.W. et al. (2016) *SqueezeNet: AlexNet-Level Accuracy with 50x Fewer Parameters and <1MB Model Size*, arXiv preprint arXiv:1602.07360.

Kang, K. et al. (2018) 'A hybrid gravitational search algorithm with swarm intelligence and deep convolutional feature for object tracking optimization', *Applied Soft Computing*, Vol. 66, pp.319–329.

Kim, B., Lee, G.G. and Yoon, J.Y. (2008) 'A method of counting people in crowded scenes', *Proc. of the 4th International Conference on Intelligent Computing*, pp.1117–1126, Springer, Berlin, Germany.

Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) 'Imagenet classification with deep convolutional neural networks', *NIPS*.

Lee, G.G., Kim, B. and Kim, W.Y. (2007) 'Automatic estimation of pedestrian flow', *First ACM/IEEE International Conference on Distributed Smart Cameras, 2007. ICDSC'07*, pp.291–296, IEEE.

Li, M. et al. (2008) 'Estimating the number of people in crowded scenes by MID based foreground segmentation and head-shoulder detection', *Proc.of IEEE 19th International Conference on Pattern Recognition*, pp.1–4, IEEE Computer Society, Washington DC, USA.

Lin, T.Y., Goyal, P., Girshick, R. et al. (2017) *Focal Loss for Dense Object Detection*, arXiv preprint arXiv:1708.02002.

Liu, W., Anguelov, D., Erhan, D. et al. (2016) 'SSD: single shot multibox detector', *European Conference on Computer Vision*, pp.21–37, Springer, Cham.

Ma, H., Lu, H. and Zhang, M. (2008) 'A real-time effective system for tracking passing people using a single camera', *Proc. of World Congress on Publication Intelligent Control and Automatio*, pp.6173–6177, IEEE Computer Society, Washington DC, USA.

Rabaud, V. and Belongie, S. (2006) 'Counting crowded moving objects', *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp.705–711, IEEE.

Redmon, J. and Farhadi, A. (2017) *YOLO9000: Better, Faster, Stronger*, arXiv preprint.

Redmon, J. and Farhadi, A. (2018) *Yolov3: An Incremental Improvement*, arXiv preprint arXiv:1804.02767.

Redmon, J., Divvala, S., Girshick, R. and Farhadi, A. (2016) 'You only look once: unified, real-time object detection', *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Las Vegas, USA, pp.779–788.

Ren, P., Fang, W., Djahel, S. (2017) 'A novel YOLO-based real-time people counting approach', *2017 International Smart Cities Conference (ISC2)*, 14–17 September, Wuxi, pp.1–2.

Ren, S., He, K., Girshick, R. et al. (2015) 'Faster R-CNN: towards real-time object detection with region proposal networks', *Advances in Neural Information Processing Systems*, pp.91–99.

Zhang, C. et al. (2015) 'Cross-scene crowd counting via deep convolutional neural networks', *Computer Vision and Pattern Recognition*, pp.833–841, IEEE.

Zhang, E. and Chen, F. (2007) 'A fast robust people counting method in video surveillance', *Proc. of International Conference on Computational Intelligence and Security*, pp.339–343, IEEE Computer Society, Washington DC, USA.

Zhang, M., Wang, H., Cui, Z. and Chen, J. (2018) 'Hybrid multi-obective cuckoo search with dynamical local search', *Memetic Computing*, Vol. 10, No. 2, pp.199–208.