

ITC 3/51 Information Technology and Control Vol. 51 / No. 3 / 2022 pp. 485-498 DOI 10.5755/j01.itc.51.3.30540	Human Detection Algorithm Based on Improved YOLO v4	
	Received 2022/01/19	Accepted after revision 2022/05/04
	 http://dx.doi.org/10.5755/j01.itc.51.3.30540	

HOW TO CITE: Zhou, X., Yi, J., Xie, G., Jia, Y., Xu, G., Sun, M. (2022). Human Detection Algorithm Based on Improved YOLO v4. *Information Technology and Control*, 51(3), 485-498. <http://dx.doi.org/10.5755/j01.itc.51.3.30540>

Human Detection Algorithm Based on Improved YOLO v4

Xuan Zhou

School of Electrical engineering, Xi'an Traffic Engineering Institute, Xi'an, 710300, China;
e-mail: 1138845898@qq.com

Jianping Yi

School of Electronics and Information, Xi'an Polytechnic University, Xi'an, 710048, China;
e-mail: 942749578@qq.com

Guokun Xie, Yajuan Jia, Genqi Xu, Min Sun

School of Electrical engineering, Xi'an Traffic Engineering Institute, Xi'an, 710300, China

Corresponding author: 1138845898@qq.com

The human behavior datasets have the characteristics of complex background, diverse poses, partial occlusion, and diverse sizes. Firstly, this paper adopts YOLO v3 and YOLO v4 algorithms to detect human objects in videos, and qualitatively analyzes and compares the detection performance of two algorithms on UTI, UCF101, HMDB51, and CASIA datasets. Then, this paper proposed an improved YOLO v4 algorithm since the vanilla YOLO v4 has incomplete human detection in specific video frames. Specifically, the improved YOLO v4 introduces the Ghost module in the CBM module, aiming to further reduce the number of parameters. Lateral connection is added in the CSP module to improve the feature representation capability of the network. Furthermore, we also substitute MaxPool with SoftPool in the primary SPP module, which not only avoids the feature loss but also provides a regularization effect for the network, thus improving the generalization ability of the network. Finally, this paper qualitatively compares the detection effects of the improved YOLO v4 and primary YOLO v4 algorithm on specific datasets. The experimental results show that the improved YOLO v4 can effectively solve the problem of complex targets in human detection tasks and improve detection speed.

KEYWORDS: Human detection, Improved YOLO v4 algorithm, Ghost module, SoftPool.

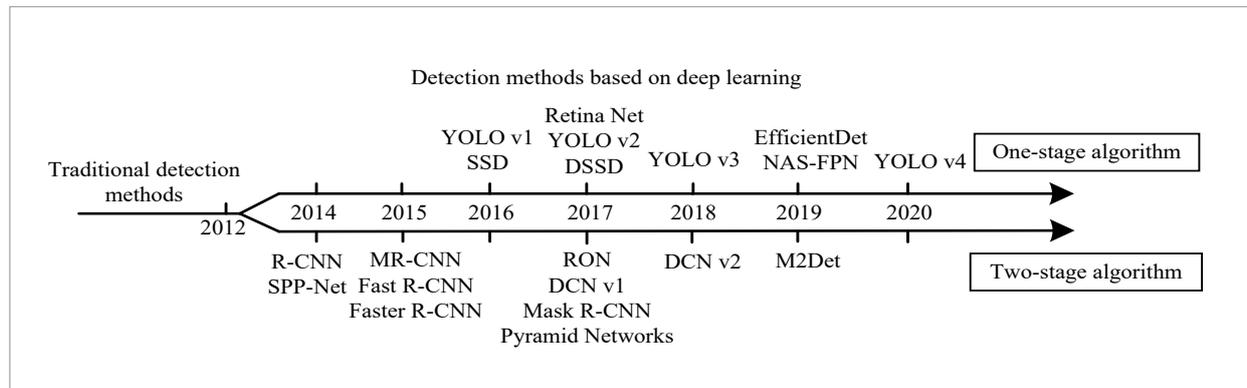
1. Introduction

In recent years, the intelligent video surveillance system has been widely concerned by the government and the public since its application in the fight against crime and the daily safety of the family [12]. Human detection [9] is closely related to personal safety and even social stability. However, human monitoring video requires a lot of manpower, and its efficiency is not satisfactory. Therefore, it is necessary to realize autonomous human detection in video. However, it is still difficult to detect the human body efficiently since

the ambiguity of human movement [16]. With the development of convolutional neural networks (CNNs) [14], the above-mentioned problems have been solved. CNN-based methods use stacked convolution and pooling to extract distinguishable and robust features from a variety of data and have made unprecedented achievements in the field of image recognition [13]. In addition, CNN-based methods have also achieved excellent performance in object detection and even surpassed human beings in some aspects [6].

Figure 1

Development history of general target detection methods



So far, the development process of object detection methods based on CNNs [7] is shown in Figure 1.

As can be seen in Figure 1, existing target detection methods can be divided into two categories according to their principle: two-stage and one-stage. Among them, the two-stage algorithm [3, 20] generally first uses a network to generate candidate regions, then uses the Region of Interest pooling (ROI-pooling) layer to adjust the size of the ROI, and corresponds the candidate targets to the corresponding positions in the feature maps. Finally, the feature information is integrated by the fully connected layer to obtain the target classification and location. This method makes the detection accuracy higher, but its detection speed is slower. Therefore, this type of method is not suitable for some scenarios with high real-time requirements, such as automatic driving, video surveillance,

intelligent security, etc. On the contrary, the workflow of the one-stage algorithm [10, 17, 18, 19, 1] is relatively simple. It does not need to generate candidate regions, but can directly predict all boundary boxes containing category probabilities and positioning coordinates by sending images into the network. This algorithm can train the shared feature completely in a single time, so the detection speed can be improved. The previous one-stage algorithm only considered the detection efficiency of the algorithm, not its accuracy. For example, although YOLO v1 has fast detection speed, its positioning is not accurate enough and its recall rate is low. Although the detection accuracy and speed of YOLO v2 and YOLO v3 are better, the detection performance for small objects is poor. In 2020, the YOLO v4 algorithm was proposed, which adopted various techniques such as data augmentation, Cross Stage Partial connection (CSP), SPP module, and

FPN+PAN module, which significantly improved the performance of human detection. However, there are still problems such as incomplete detection.

Regarding the issue above, we propose the improved YOLO v4 to improve the accuracy and efficiency of YOLO v4. In specially, the proposed improved YOLO v4 introduces the Ghost module in the original CBM module of YOLO v4, and adds lateral connections in the CSP module, in addition, we also substitute MaxPool with SoftPool in the primary SPP module. Notably, three components are computationally lightweight and can significantly enhance the detection performance of the baseline. Finally, our extensive validation experiments on four benchmark datasets (UTL, UCF101, HMDB51, and CASIA) show that the proposed improved YOLO v4 achieves a good trade-off in accuracy and speed of object detection. In the subsequent visualization, it was found that the improved YOLO v4 detection method had the best detection performance compared with the existing detection methods based on YOLO v3 and YOLO v4, which further proved the superiority of the proposed method in extracting valid features.

We summarize our main contributions as follows:

- 1 To reduce the computational costs of the backbone, Ghost module is introduced into the original CBM module of YOLO v4, aiming to further reduce the number of parameters in the network, reduce the coupling degree between channels, and promote the channels of different groups to extract complementary features.
- 2 To improve the feature representation capability of the network, we introduce lateral connection in the CSP module to realize feature interaction between the primary branch and the secondary branch, which integrates the features of different semantic levels, and improves the efficiency of parameter updating during backpropagation.
- 3 From the perspective of feature loss, the Max-Pool in SPP module is replaced by SoftPool, which avoids feature loss and provides certain regularization effect for the network, to improve the generalization ability of the network.

The rest of this paper is arranged as follows. Section 2 presents some related works, while section 3 gives research methodology. Then, Section 4 presents the experimental results and analysis. Lastly, Section 5 concludes the work.

2. Related Work

The target detection technology is applied to the human detection task, and the human is the only target detected in this paper. Human detection methods based on deep learning are mainly divided into two types: Anchor-based and Anchor-Free-based detection methods. The former uses Anchor to generate a large number of Anchor boxes so that it can determine the category and position of the target. The latter does not need to generate Anchor boxes and detect objects directly from images.

2.1. Anchor

The human detection method based on Anchor focuses on marking the human body in the dataset and then uses the coordinates of these Anchors as the starting point to predict the error value of the annotation box. Faster R-CNN is the target detection method based on Anchor and belongs to the two-stage detection model. The working principle of this algorithm is that candidate areas are generated in the images to be tested, then the ROI size is adjusted by using the ROI-pooling layer, and then the network layer is used to further classify and locate the candidate areas, which achieves ideal results in human detection tasks.

Because Faster R-CNN is not ideal for detecting small human bodies in pictures, it will also produce false detections. Because of these problems, Zhang et al. [23] proposed the RPN+BF model, which first modified the RPN to generate candidate areas suitable for the proportion of the human body, then used ROI-pooling to generate features of a certain length, and finally used cascaded Boosted Forest directly trains its features. This method alleviates the problem of false detection and missed detection of small objects, but still does not solve the multi-scale problem. Therefore, Cai et al. [2] proposed the MS-CNN model, which extracts candidate regions on feature maps of different scales, so that it has a better detection effect on human bodies of different sizes. In addition, to further improve the detection accuracy of Faster R-CNN, Zhang et al. [24] also proposed a variety of improvement strategies and solved the problems of severe occlusion and small target well. However, the two-stage human detection models all face the problem of high computational complexity, so they cannot detect human bodies in videos in real-time. However,

the currently popular one-stage detection models all abandon the step of generating candidate regions, reduce the amount of computation, and achieve a good compromise between detection accuracy and detection speed.

Recently, a variety of human detection models based on Anchor and belonging to one-stage have been proposed. For example, Liu et al. [11] proposed the ALFNet model, which introduced the cascading idea based on SSD network, that is, when the prediction frame is generated, the IOU threshold is continuously increased, so that the high-quality prediction boxes near the target increases, thereby improve its positioning accuracy. In addition, Lin et al. [8] proposed the human attention mechanism to effectively identify human regions, and introduced the zoom-in-zoom-out module to integrate local information and context information, thus improving the feature extraction capability of the model.

2.2. Anchor-Free

From two-stage to one-stage, Anchor-based to Anchor-Free-based development route. Now, human detection has entered the era of Anchor-free. Because the algorithm flow of the detection method based on Anchor-Free is relatively simple, it does not require a large number of steps, the computational complexity is small, and has a large space for improvement in detection rate, which promotes its rapid development.

The detection method based on Anchor-Free does not need to define Anchor, and can directly detect the target from the picture. In 2015, Huang et al. [5] proposed the Densebox model, which is a multi-task target detection framework based on Fully Convolutional Network (FCN), which introduces key point information and improved positive and negative sample distinction. The strategy improves the detection accuracy, and at the same time, to improve the recall rate of small targets, an up-sampling operation is also introduced to fuse the features of the shallow and deep layers to obtain a larger-scale output feature map. However, YOLO v1 divides the picture into grids and detects the target through the grid, which can greatly improve the detection speed. In addition, it can directly regression the classification probability and location coordinate information of targets through the full connection layer, but its accuracy is unsatisfactory. Therefore, Tian et al. [22] proposed

the FCOS model, which designed a three-branch network and added Feature Pyramid Networks (FPN) to further improve the detection accuracy. Compared with YOLO v1 and Densebox, this model Uses FPN for multi-scale prediction, the detection effect of small target humans is better. However, compared with the single-branch model YOLO v1, other methods are intensive prediction, so there are too many parameters, and they need to be classified and regressive through two sub-networks, resulting in slower detection speed. With the continuous updating of the network, YOLO v3 and YOLO v4 of the YOLO series of algorithms are both one-stage modes and adopt new methods such as multi-scale prediction, which achieve a perfect balance between detection speed and detection accuracy.

Target detection methods, such as R-CNN and Faster R-CNN, generate many potential boundary boxes around the target to be tested and then use classifiers to determine its specific position and category probability, all of which belong to the two-stage model and the detection speed is slow. However, YOLO adopts the regression idea to detect the target, inputs the whole image into the network, extracts features from the image, and finally directly outputs the category and position of the target, which belongs to the one-stage model, and improves the detection speed without affecting the detection accuracy. Therefore, this paper selects the YOLO series algorithm to detect the human body in the video.

3. Research Methodology

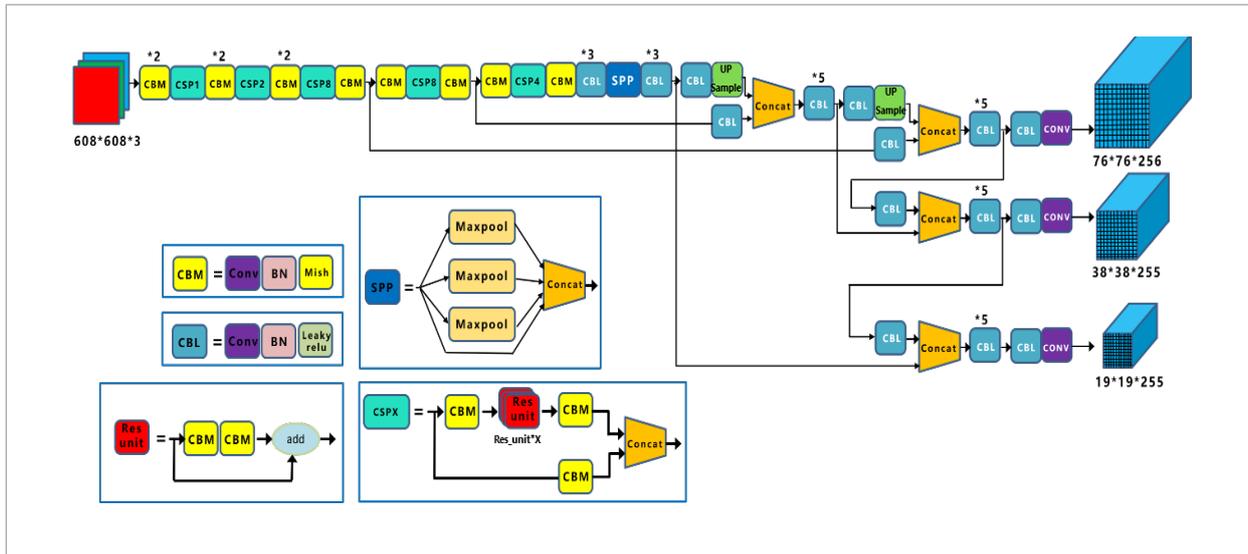
In this section, we mainly describe the details of the improved YOLO v4 human detection algorithm. First, we review the overall network structure of YOLO v4. Then, the design principles of Backbone and Neck are described, respectively, and the advantages and disadvantages of modules are discussed. Finally, we provide the implementation details of human detection using the improved YOLO v4 algorithm.

3.1. Network Architecture

The YOLO v4 has made all-around improvements to YOLO v3, significantly improving the accuracy and speed of object detection, its overall architecture is shown in Figure 2. The YOLO v4 includes four parts:

Figure 2

The overall architecture of YOLO v4



input, backbone, neck, and output. In specially, YOLO v4 integrates mosaic data augmentation, self-adversarial training (SAT), and cross mini-batch normalization (CmBN) at the input, so that the network can leverage smaller batch size during training, thus reducing the memory usage. Its backbone adopts CSPDarkNet53 to enhance the feature extraction ability. Furthermore, the YOLO v4 uses the spatial pyramid pooling (SPP) module to increase the spatial receptive field. Finally, the path aggregation network (PANet) is applied to integrate features of different semantic levels to improve the performance of small object detection.

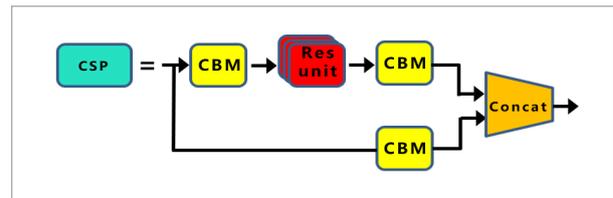
3.1.1. Backbone

The role of the backbone is feature extraction. Its innovations include CSPDarknet-53, Mish activation function, and DropBlock regularization. Specifically, CSPDarknet-53 is mainly based on the Darknet-53 of YOLO v3 by adding a CSP module to alleviate the problem of duplication of gradient information generated during training. The architecture of the CSP module is shown in Figure 3.

It can be seen from Figure 3, the CSP module uses two branches to operate on input features to further improve feature reuse. Meanwhile, the first CBM module reduces the number of channels by half to reduce

Figure 3

The architecture of the CSP module



the computational costs of the module. Then, the output feature maps of the two branches are concatenated according to the channel dimension by the cross-stage hierarchical structure. In this manner, the CSP module helps to integrate low-level and high-level semantic features, thereby effectively improving the feature learning capacities of the backbone.

Supposing the input is 416x416, the architecture of CSPDarkNet53 is shown in Table 1. The numbers of RB1 to RB5 modules are 1, 2, 8, 8, and 4, respectively.

YOLO v4 uses the Mish activation function in the backbone to ensure that the gradient of neuron activation near 0 is smooth, thus improving the convergence speed of the network. The expression of the Mish function is shown in Equation (1).

$$f(x) = x \cdot \tanh(\ln(1+e^x)). \tag{1}$$

Table 1

Architecture of CSPDarkNet53

Input	Type	Stride	Padding	Upsample	Output
416*416*3	Conv, 3*3	1	1		416*416*32
416*416*32	Conv, 3*3	2	1		208*208*64
208*208*64	RB1, 1*1	1	0		208*208*32
208*208*32	RB1, 3*3	1	1		208*208*64
208*208*64	Conv, 3*3	2	1		104*104*128
104*104*128	RB2, 1*1	1	0		104*104*64
104*104*64	RB2, 3*3	1	1		104*104*64
104*104*64	Conv, 3*3	2	1	√	52*52*256
52*52*256	RB3, 1*1	1	0		52*52*128
52*52*128	RB3, 3*3	1	1		52*52*128
52*52*128	Conv, 3*3	2	1	√	26*26*512
26*26*512	RB4, 1*1	1	0		26*26*256
26*26*256	RB5, 3*3	1	1		26*26*256
26*26*256	Conv, 3*3	2	1	√	13*13*1024
13*13*1024	RB5, 1*1	1	0		13*13*512
13*13*512	RB5, 3*3	1	1		13*13*512

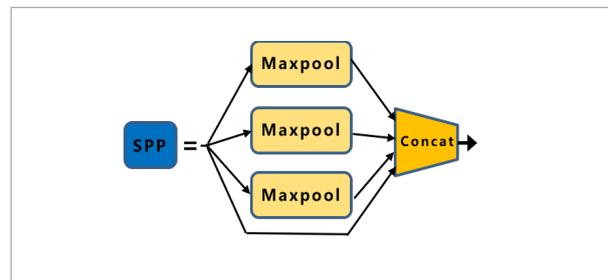
In addition, to alleviate the over-fitting, YOLO v4 uses DropBlock regularization. The Dropout [15] is to randomly delete some neurons to make the network structure sparser. Its effect is particularly good in the fully connected layer, but the effect is poor in the convolutional layer of the backbone. The DropBlock method discards the features of continuous regions, thus making the network model simpler, and can learn new weights of the network to compensate for the feature matrix during the training process to prevent over-fitting.

3.1.2. Neck

The neck is located between the backbone and the output, and mainly performs feature aggregation to improve the detection accuracy of small objects. The Neck structure of YOLO v4 includes SPP and PANet modules. The architecture of the SPP module is shown in Figure 4. First, the module processes the intermediate feature maps through identity mapping and maximum

Figure 4

The architecture of the SPP module



pooling of different kernel sizes. Then, the multi-scale output feature maps are concatenated according to the channel dimensions. Finally, the CBL unit is used to map the channels of the concatenated feature maps to the original channel dimensions. In this manner, it is possible to effectively capture multi-scale spatially salient information without affecting the inference

speed of the network, thus increasing the local spatial receptive field of the network.

To improve the detection performance of small objects, the YOLO v4 proposes the PANet module to achieve feature aggregation at different semantic levels. The PANet transfers the location information contained in the low-level semantic features to the high-level semantic features through bottom-up and top-down. Meanwhile, it utilizes up-sampling, down-sampling, and concatenate operations to aggregate multi-scale semantics, and the aggregated features are used for multi-stage prediction. In this manner, it is beneficial to utilize the position information of the bottom-level features and the semantic information of the top-level features at the same time, thus significantly improving the detection effect of the model for small objects.

3.1.3. Head

The innovations of YOLO v4 in Head mainly include CIOU_loss loss function and DIOU_nms non-maximum suppression (NMS). Among them, CIOU_loss is a parameter to measure the length-width ratio of boundary frame added based on the DIOU_loss loss function, and its function expression is shown in Equation (2).

$$CIOU_loss = 1 - CIOU = 1 - \left(IOU - \frac{Distance_2^2}{Distance_C^2} - \frac{v^2}{(1 - IOU) + v} \right), \quad (2)$$

where v denote a parameter used to measure the similarity of aspect ratio, its calculation formula is shown in Equation (3).

$$v = \frac{4}{\pi^2} \left(\arctan \frac{w^{gt}}{h^{gt}} - \arctan \frac{w^p}{h^p} \right)^2 \quad (3)$$

According to Equations (2)-(3), the CIOU_loss loss function takes aspect ratio, IOU, and center point distance into account to improve the detection rate of the network.

Most object detection networks adopt the primary NMS method to screen the prediction frames, while YOLO v4 is inspired by the idea of DIOU_loss to further suppress redundant frames by considering the IOU and the center point distance of adjacent prediction frames, so that it can obtain a high detection accuracy even under occlusion.

3.2. Improved YOLO v4

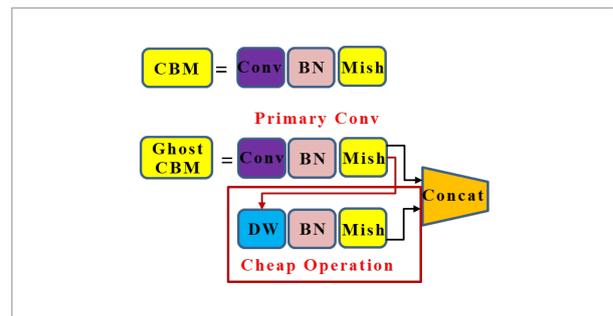
To further reduce the number of parameters and computational costs of YOLO v4, in this paper, we proposed the Ghost CBM module and improved CSP modules, aiming to improve the performance of object detection on small datasets. Furthermore, we proposed the Soft-SPP (S-SPP) module to alleviate the feature loss, this module can provide a regularization effect for the network, thus improving the generalization performance of the network.

3.2.1. Modified CBM Module (Ghost CBM)

Inspired by Han et al. [23], there are redundant Ghost features in the middle layer of deep neural networks. The Ghost features are an important component of the feature layer, and other features can be obtained by the linear mapping of Ghost features. Therefore, to further reduce the number of parameters in YOLO v4, we proposed the Ghost CBM module. The structure diagram of the original CBM module and the Ghost CBM module proposed in this paper is shown in Figure 5.

Figure 5

The architecture of the CBM module and the modified CBM module



It can be seen from Figure 5, the Ghost CBM first uses primary convolution to generate real feature maps, and then carries out the depth-wise convolution to complete the linear transformation of the real feature maps, to obtain Ghost feature maps. Finally, the real feature maps and Ghost feature maps are concatenated according to the channel dimension to obtain the output feature maps. Given the input feature maps $\mathbf{X} \in H \times W \times C_p$, and the output feature maps $\mathbf{Y} \in H' \times W' \times C_o$, the kernel size of the convolution layer is $k \times k$. The computation of primary convolution is formulated as:

$$H' \times W' \times C_0 \times k \times k \times C_i. \quad (4)$$

The computational costs of the Ghost module include two parts, the first part is primary Conv and the second part is Cheap Operation. The output feature maps are divided into n parts, among which 1 part performs *Primary Conv* and $(n-1)$ part performs *Cheap Operation*. Meanwhile, given the convolution kernel of the first part is $k \times k$ and the second part is $d \times d$. The computation costs of the Ghost CBM is formulated as:

$$H' \times W' \times \frac{C_0}{n} \times k \times k \times C_i + (n-1) \times \frac{C_0}{n} \times H' \times W' \times d \times d. \quad (5)$$

Comparing Equations (4)-(5), when $n \ll C_{in}$, compared with original CBM, the number of parameters of Ghost CBM module was reduced by n . In subsequent experiments, n is set to 2. By introducing the Ghost module, the overall parameters and computation of YOLO v4 can be significantly reduced, which is beneficial to alleviate the over-fitting problem of the model and improve the feature extraction ability of the network.

3.2.2. Improved CSP Module (I-CSP)

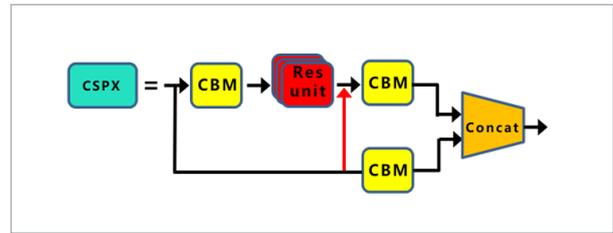
As can be seen from Section 3.1.1, the CSP module applies the idea of a cross-stage local network to extract multi-level features. Meanwhile, it can be seen from Figure 3 that the primary and secondary branches of the original CSP module perform two kinds of asymmetric convolution operations in parallel, so the output feature maps of the two branches have different semantic levels. Based on the above observation, we proposed the improved CSP module, its architecture is shown in Figure 6. In this module, this module performs element-wise addition on the feature maps of the two branches through lateral connection to realize the feature interaction of the two branches. In this manner, when the next convolution is performed, high-level and low-level semantic information can be effectively aggregated. In addition, the parameter updating efficiency of backpropagation can be improved by lateral connection, it is beneficial to extract rich and multi-level features, thus improving the feature representation ability of the network.

3.2.3. Modified SPP (Soft SPP)

Notably, the maximum pooling of the SPP module will lead to the risk of loss of important information, thus affecting the detection effect to a certain extent. To retain the original feature activation as much as possible, we proposed the Soft SPP(S-SPP) module to re-

Figure 6

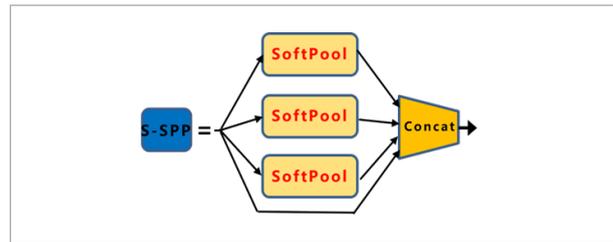
The architecture of the I-CSP module



place the original maximum pooling. Its architecture is shown in Figure 7. Among them, the red marks in the figure represent the improvement of the original network in this paper.

Figure 7

The architecture of the S-SPP module



Specifically, SoftPool [21] uses the smoothed maximum approximation of activation in the region of the kernel, that is, each activation with index i applies a weight, which is calculated as the ratio of the activation's natural index to the sum of all activated natural indexes in the neighborhood, as shown in Equation (6). In this manner, it can ensure that each pixel is assigned a weight and that large activations are more dominant than small ones. Highlighting more efficient activation is a more balanced approach than average pooling and max pooling.

$$w_i = \frac{e^{a_i}}{\sum_{j \in \mathbf{R}} e^{a_j}}, \quad (6)$$

where, a_i denotes the activation value whose index is i in the neighborhood \mathbf{R} of the pooled kernel. The output value of SoftPool is summed by weighted activation in neighborhood \mathbf{R} , as shown in Equation (7).

$$\alpha = \sum_{i \in \mathbf{R}} w_i * a_i. \quad (7)$$

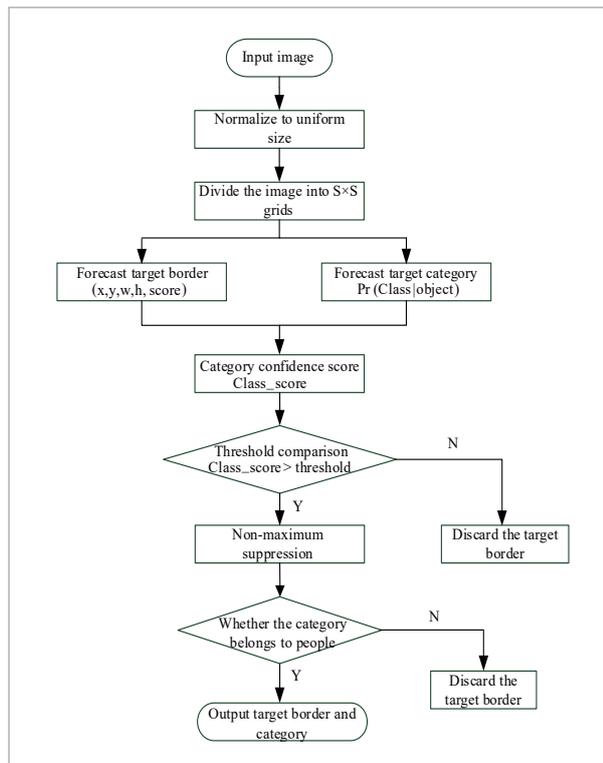
To sum up, this paper has carried on the improvement of the Backbone and Neck of YOLO v4. First of all, from the perspective of computational costs, the idea of the Ghost module is introduced into the CBM module to further reduce the number of parameters and improve the performance of the model on small sample datasets. Then, from the perspective of feature and parameter learning, the feature interaction between the primary branch and the sec branch in the CSP module is realized through lateral connection, which not only integrates the features of different semantic levels but also improves the efficiency of parameter update in the back-propagation. Finally, from the perspective of feature loss, the MaxPool in the SPP module is replaced by SoftPool, which can avoid feature loss, and provides a certain regularization effect for the network, to improve the network generalization ability.

3.3. Design of Improved YOLO v4 Human Detection Algorithm

The flow for detecting human by applying the improved YOLO v4 algorithm is shown in Figure 8.

Figure 8

Improved YOLO v4 algorithm flow design



In this paper, the improved YOLO v4 algorithm is used to detect human in the video. Therefore, ROI_index is set to 0 corresponding to the person category to label only human. In addition, to overcome the problem of target overlap, the IOU threshold is set to 0.45, and the category confidence is set to 0.25 to weaken the phenomenon of missed detection, to further improve the detection effect of the network.

4. Experiment

In this section, we first describe the benchmark databases and implementation details. Then, we conduct extensive ablation experiments on the UTI dataset to verify the effectiveness of the proposed improved YOLO v4 algorithm and seek its optimal setting. Finally, we compare the improved YOLO v4 with previous methods on the UCF101, HMDB51, UTI, and CASIA datasets.

4.1. Datasets

UCF101. The UCF101 dataset contains 101 classes and a total of 13,320 videos with 320×240 pixels. The dataset has various behavior categories, including individual sports, human-object interaction, and person-to-person interaction. However, since the goal of this paper is to detect the people in the video, we use the part of UCF101 to display the behavior of the whole human, such as baseball pitch, basketball, squatting, fencing, throwing discus, etc.

HMDB51. HMDB51 contains 51 classes with a total of 6,849 videos with a pixel of 320×240 . This dataset includes five categories, such as body movement (hand clapping, handstand, etc.), facial movement (laughing, chewing, etc.), body-object interaction movement (shooting, combing hair, etc.), face-object interaction movement (eating, drinking, etc.), and individual movement (fencing, hugging, etc.). Similarly, this part only uses part of HMDB51 to display the behavior of the whole human, such as dribbling, fencing, jumping, picking up the ball, punching, etc.

UTI. The UTI dataset consists of 20 video clips, each containing multiple moving persons, it contains six behaviors such as punching, kicking, shaking, pointing, hugging, and pushing. Each video contains at least one of these six interactions, resulting in an average of eight human movements per video clip. The resolution is 720×480 and the frame rate is 30FPS

(frames per second). In this paper, various behaviors in the video are cut out separately, and the same behaviors are placed in the same folder.

CASIA. The CASIA dataset is a behavioral analysis of parking lots released by the Chinese Academy of Sciences, which includes single-person behaviors such as stoop walking, running, and smashing cars, and includes two-person interactions such as fighting, robbery, and stalking. The video resolution is 320×240 , the frame rate is 25FPS, and the video format is avi.

4.2. Implementation Details

Experimental setup. CPU is Inter(R) Core (TM) I5-4210M, the frequency is 2.60GHz. The software environment is configured on this hardware: Ubuntu 16.04, Cuda9.0, Python3, Opencv, etc. We implement the proposed model on Tensorflow and Keras framework.

Since the object of this paper is to detect the person in the video, thus the class of {yolo} layer in the file is set to 1, the corresponding category is person, and the filters convolved at the bottom layer are changed to 18, whose calculation formula is as follows: $1 \times 1 \times [3 \times (4 + 1 + 1)] = 18$, i.e. $N \times N \times [\text{num} \times (\text{cords} + \text{classes} + 1)]$, where N is the kernel size of the convolu-

tion layer, num is 3 (number of predicted prior boxes), cords are 4 (coordinate value of prediction box). The following 1 denotes a single classification, and the last 1 is confidence. After that, modify some network parameters, set stride and padding to 1, filter to 10, mask to (6,7,8), The anchor is set to (10, 13, 16, 30) (33, 23) (30, 21) (50119) (116), living (156198) (373326), etc. Finally, some other parameters were modified according to their requirements. The input image was an RGB image with the size of 416×416 . The momentum used in the training process was 0.9, decay was 0.0005. The test batch size is 1, and the training batch is 32, the batch indicates the number of training samples. For a fair comparison, the same training and testing protocols were used for each method.

4.3. Algorithm Testing

After setting the network parameters, we fine-tune the YOLO v3 and YOLO v4 algorithms, and set the momentum, weight decay, and learning rate to 0.9, 0.0005, and 0.0001, respectively. The two methods were evaluated on UCF101, HMDB51, UTI, and CASIA datasets, and the pre-training model is loaded to detect the persons in the datasets. The detection results are shown in Figures 9-10, respectively.

Figure 9

The detection results of YOLO v3

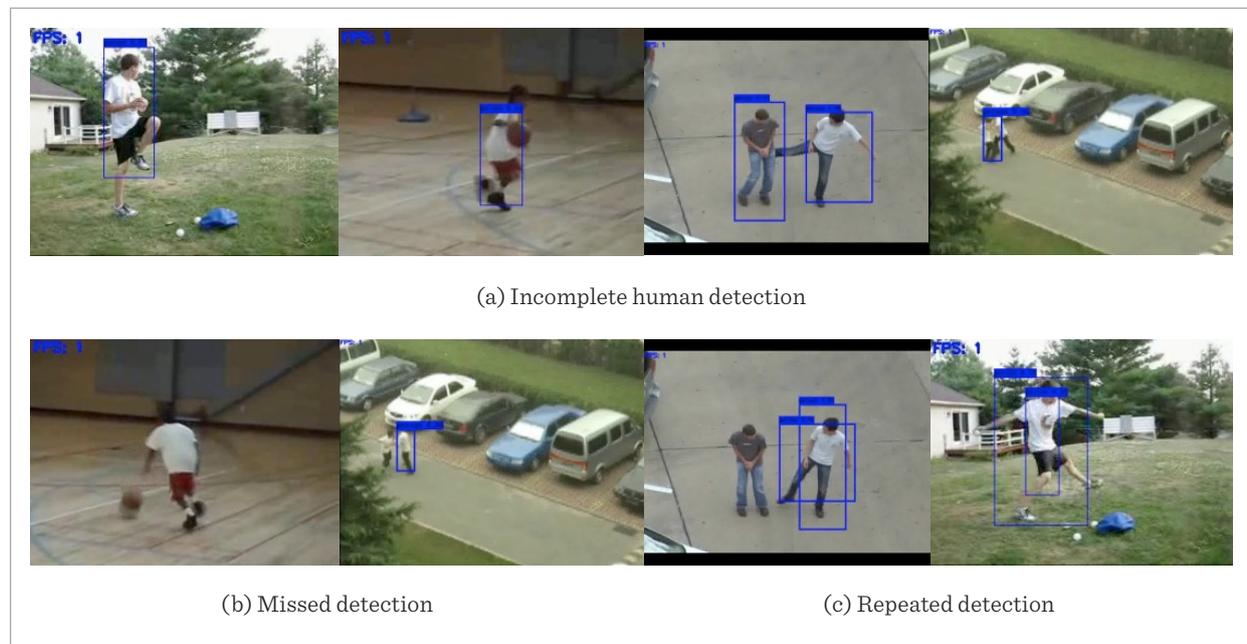
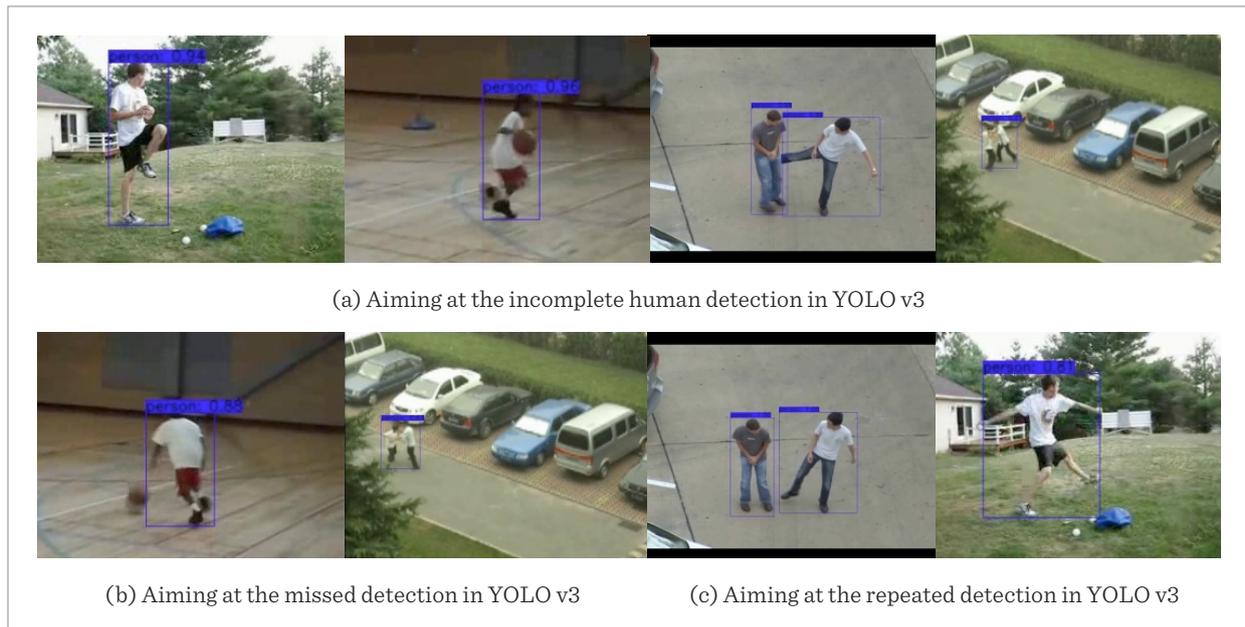


Figure 10

The detection results of YOLO v4



By comparing Figures 9-10, it can be seen that YOLO v3 has poor human detection effects on the UCF101, HMDB51, UTI, and CASIA datasets. There are three main problems: incomplete human detection, missed detection, and repeated detection. To alleviate the above problems, YOLO v4 has significantly improved the performance of human detection by adopting various techniques such as data augmentation, cross-stage partial connection, SPP module, and FPN+PAN module.

In specially, YOLO v4 uses the Mosaic data augmentation and the SPP module to alleviate the problem of incomplete human detection. On the one hand, the Mosaic data augmentation enriches the background of the target to be detected and improves the stability of the network. The SPP module uses the spatial pyramid pooling layer module to separate the most important features and extract them efficiently and makes CNN not immune to the limitation of input size is fixed to further alleviate the problem of incomplete human detection.

The YOLO v4 introduced the FPN+PAN method to solve the missed detection problem. Combining FPN and PAN modules, not only can better aggregate the features of the network layer, but also further improve the feature extraction capabilities of the YOLO v4 network.

The YOLO v4 introduces CIOU_loss loss function and DIOU_nms non-maximum suppression to alleviate the repeated detection. The CIOU_loss loss function takes the aspect ratio, IOU value, and center point distance into account, alleviating the problem of repeated detection. The DIOU_nms method further suppresses redundant frames by considering the IOU value and the distance between the center points of adjacent prediction frames, so that it can obtain a higher detection rate even under occlusion, and further alleviate the problem of repeated detection.

In summary, the YOLO v4 algorithm alleviates these three problems and significantly improves the effect of human detection. However, it can be seen from Figure 10(b) that this algorithm still has the problem of incomplete human detection. In response to this problem, this article will further improve the YOLO v4 algorithm, and use the improved YOLO v4 algorithm to efficiently detect the human body in the video.

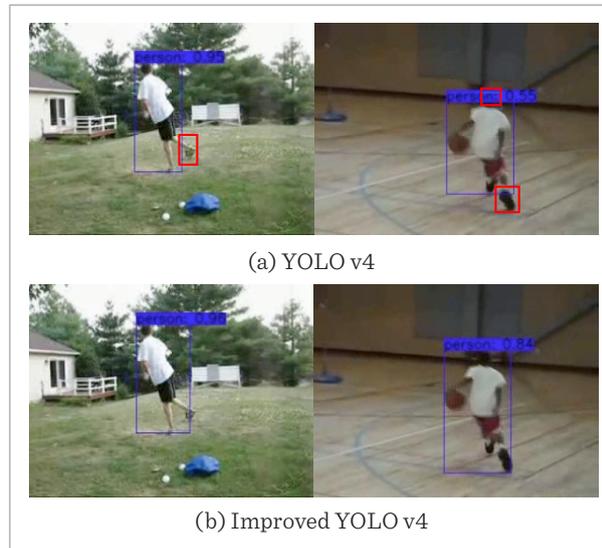
4.4. Performance Analysis

To demonstrate the effectiveness of the proposed improved YOLO v4, we conduct the performance comparison of the proposed algorithm with the traditional YOLO v4 algorithm on the UTI, UCF101, HMDB51,

and CASIA datasets. The qualitative comparison results are shown in Figure 11. For a fair comparison, we use the same training and testing protocol as YOLO v4 in our improved YOLO v4 (see section 4.2).

Figure 11

Comparison of detection performance before and after improvement



It can be seen from Figure 11, for the same frame in the video, the human object in the prediction frame of the improved YOLO v4 is more complete and takes up a larger proportion, so the detection effect is better than the original YOLO v4 algorithm. This performance improvement is mainly due to the three optimization strategies of YOLO v4 in this article, that is, the introduction of the Ghost module, the addition of horizontal connections, and the replacement of pooling methods. Specifically, the proposed Ghost CBM module further reduces the computational costs of the network and improves the performance of the model on small datasets. Meanwhile, adding lateral connections not only integrates the features of different semantic levels, but also improves the efficiency of parameter learning during backpropagation. Finally, replace MaxPool in the SPP module with SoftPool, which not only avoids feature loss, but also provides a certain regularization effect for the network, thereby improving the generalization ability of the network.

To further confirm the effectiveness of the proposed improved YOLO v4, we conducted experiments on

the UCF101, HMDB51, UTI, and CASIA datasets. Table 2 shows the performance comparison of YOLO v3, YOLO v4, and improved YOLO v4 on the four human behavior datasets.

Table 2

Comparison of detection performance before and after improvement

Methods \ Datasets	Datasets			
	UTI	UCF101	HMDB51	CASIA
YOLO v3	87.4	74.9	72.8	60.5
YOLO v4	92.3	90.4	89.6	82.6
Improved YOLO v4	93.8	91.5	90.6	84.1

As can be seen from Table 2, the improved YOLO v4 has improved the detection accuracy compared with the YOLO v3 and YOLO v4 on four benchmark databases. In specially, compared with YOLO v4, the average accuracy (AP) of the proposed method on UTI, UCF101, HMDB51, and CASIA datasets is improved by 1.5%, 1.1%, 1.0%, and 1.5%, respectively. Through the lateral connection of the proposed improved CSP module, cross-level features will be effectively integrated, thus significantly enhancing the feature extraction ability of the backbone. In addition, the proposed S-SPP module can reduce feature loss, thus improving the generalization performance of the network.

In addition, to further evaluate the detection speed of the proposed improved YOLO v4, table 3 compared the detection speed of YOLO v3, YOLO v4, and improved YOLO v4 on four datasets.

Table 3

The detection speed comparison of different algorithms (seconds/frame)

Methods \ Datasets	Datasets			
	UTI	UCF101	HMDB51	CASIA
YOLO v3	0.785	0.909	0.881	0.912
YOLO v4	0.652	0.791	0.789	0.806
Improved YOLO v4	0.627	0.763	0.765	0.781

As can be seen from Table 3, the inference speed of improved YOLO v4 is better than YOLO v3 and YOLO v4 on all datasets. The experimental results show that the improved YOLO v4 improves the learning ability of the backbone and reduces computational costs, thus improving the detection speed of baseline. It can be seen from Table 3 that the detection speed of three algorithms on the UTI dataset is faster than that on other datasets, due to the background of the UTI dataset is simple, with little interference and easy to distinguish the human, while the interference of other datasets is relatively large and difficult to distinguish the human. In summary, the verification experiments on four datasets show that the proposed improved YOLO v4 has excellent detection performance and strong generalization performance.

5. Conclusion

To extract distinguishing features to improve the detection performance of the YOLO v4, we proposed the improved YOLO v4 algorithm. In specially, we proposed Ghost CBM to reduce network parameters, thus improving inference speed. Meanwhile, we presented the improved CSP module to enhance the feature extraction ability of the backbone, thus improving the detection accuracy of the baseline. Furthermore, we proposed the Soft SPP module to alleviate the feature loss caused by the max pooling. Our improvements to YOLO v4 are comprehensive, and the improved YOLO v4 presented focuses on both efficiency and accuracy. Finally, extensive verification experiments on four

datasets show that the proposed improved YOLO v4 significantly improves the speed and accuracy of human detection.

Appendix A

The download addresses of the four datasets used in this article are as follows:

UCF101:

<http://dataju.cn/Dataju/web/datasetInstanceDetail/134>.

HMDB51:

<http://dataju.cn/Dataju/web/datasetInstanceDetail/125>.

UTI:

<http://dataju.cn/Dataju/web/datasetInstanceDetail/144>.

CASIA:

<http://www.cbsr.ia.ac.cn/china/Action%20Databases%20CH.asp>.

Acknowledgements

This work was supported by the Young and middle-aged fund project of Xi'an Traffic Engineering Institute (2022KY-02).

Ethical Permit

The authors declare that they have no conflict of interest. All procedures performed in studies involving human participants were in accordance with the ethical standards of the institutional and national research committee. This article does not contain any studies with animals performed by any of the authors. Informed consent was obtained from all individual participants included in the study.

References

1. Bochkovskiy, A., Wang, C., Liao, H. YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv preprint, 2020, arXiv:2004.10934.
2. Cai, Z., Fan, Q., Feris, R., Vasconcelos, N. A Unified Multi-scale Deep Convolutional Neural Network for Fast Object Detection. Proceedings of the European Conference on Computer Vision, (ECCV 2016), Scottsdale, AZ, USA, November 3-7, 2016, 354-370. https://doi.org/10.1007/978-3-319-46493-0_22
3. Girshick, R. Fast R-CNN. Proceedings of the IEEE International Conference on Computer Vision, (ICCV 2015), Santiago, Chile, December 11-18, 2015, 1440-1448. <https://doi.org/10.1109/ICCV.2015.169>
4. Han, K., Wang, Y. H., Tian, Q., Guo, J. Y., Xu, C. J. GhostNet: More Features from Cheap Operations. arXiv preprint, 2019, arXiv:1911.11907v2. <https://doi.org/10.1109/CVPR42600.2020.00165>
5. Huang, L., Yang, Y., Deng, Y., Yu, Y. Densebox: Unifying Landmark Localization With End to End Object Detection. arXiv preprint, 2015, arXiv:1509.04874.
6. Kulikajavas, A., Maskeliūnas, R., Damaševičius, R. Detection of Sitting Posture Using Hierarchical Image

- Composition and Deep Learning. *PeerJ Computer Science*, 2021, 7, e442. <https://doi.org/10.7717/peerj-cs.442>
7. Li, Z., Liu, F., Yang, W., Peng, S., Zhou, J. A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects. *arXiv preprint*, 2020, arXiv:2004.02806. <https://doi.org/10.1109/TNNLS.2021.3084827>
 8. Lin, C., Lu, J., Wang, G., Zhou, J. Graininess-aware Deep Feature Learning for Pedestrian Detection. *Proceedings of the European Conference on Computer Vision, (ECCV 2018)*, Munich, Germany, September 8-14, 2018, 745-761. https://doi.org/10.1007/978-3-030-01240-3_45
 9. Liu, C., Szirányi, T. Real-Time Human Detection and Gesture Recognition for On-Board UAV Rescue. *Sensors*, 2021, 21(6), 2180. <https://doi.org/10.3390/s21062180>
 10. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. SSD: Single Shot Multibox Detector. *Proceedings of the European Conference on Computer Vision, (ECCV 2016)*, Amsterdam, Netherlands, October 8-16, 2016, 21-37. https://doi.org/10.1007/978-3-319-46448-0_2
 11. Liu, W., Liao, S., Hu, W., Liang, X., Chen, X. Learning Efficient Single-stage Pedestrian Detectors by Asymptotic Localization Fitting. *Proceedings of the European Conference on Computer Vision, (ECCV 2018)*, Munich, Germany, September 8-14, 2018, 643-659. https://doi.org/10.1007/978-3-030-01264-9_38
 12. Mabrouk, B., Zagrouba, A. Abnormal Behavior Recognition for Intelligent Video Surveillance Systems: A Review. *Expert Systems with Applications*, 2018, 91, 480-491. <https://doi.org/10.1016/j.eswa.2017.09.029>
 13. Mu, H., Sun, R., Yuan, G., Wang, Y. Abnormal Human Behavior Detection in Videos: A Review. *Information Technology and Control*, 2021, 50(3), 522-545. <https://doi.org/10.5755/j01.itc.50.3.27864>
 14. Mu, R., Zeng, X. A Review of Deep Learning Research. *KSII Transactions on Internet and Information Systems*, 2019, 13(4), 1738-1764. <https://doi.org/10.3837/tiis.2019.04.001>
 15. Poernomo, A., Kang, D. K. Biased Dropout and Crossmap Dropout: Learning Towards Effective Dropout Regularization in Convolutional Neural Network. *Neural Networks*, 2018, 104, 60-67. <https://doi.org/10.1016/j.neunet.2018.03.016>
 16. Qian, H., Zhou, X., Zheng, M. Abnormal Behavior Detection and Recognition Method Based on Improved ResNet Model. *Computers, Materials & Continua*, 2020, 65(3), 2153-2167. <https://doi.org/10.32604/cmc.2020.011843>
 17. Redmon, J., Divvala, S., Girshick, R., Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2016)*, Las Vegas, NV, USA, June 26-July 1, 2016, 779-788. <https://doi.org/10.1109/CVPR.2016.91>
 18. Redmon, J., Farhadi, A. YOLO9000: Better, Faster, Stronger. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2017)*, Honolulu, HI, USA, July 21-26, 2017, 6517-6525. <https://doi.org/10.1109/CVPR.2017.690>
 19. Redmon, J., Farhadi, A. YOLOv3: An Incremental Improvement. *arXiv preprint*, 2018, arXiv:1804.02767.
 20. Ren, S., He, K., Girshick, R., Sun, J. Faster R-CNN: Towards Real-time Object Detection With Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(6), 1137-1149. <https://doi.org/10.1109/TPAMI.2016.2577031>
 21. Stergiou, A., Poppe, R., Kalliatakis, G. Refining activation downsampling with SoftPool. *arXiv preprint*, 2021, arXiv:2101.00440v3. <https://doi.org/10.1109/ICCV48922.2021.01019>
 22. Tian, Z., Shen, C., Chen, H., He, T. FCOS: Fully Convolutional One-stage Object Detection. *Proceedings of the IEEE International Conference on Computer Vision, (ICCV 2019)*, Seoul, Korea, Republic of, October 27-November 2, 2019, 9626-9635. <https://doi.org/10.1109/ICCV.2019.00972>
 23. Zhang, L., Lin, L., Liang, X., He, K. Is Faster R-CNN Doing Well for Pedestrian Detection? *Lecture Notes in Computer Science*, 2016, 443-457. https://doi.org/10.1007/978-3-319-46475-6_28
 24. Zhang, S., Benenson, R., Schiele, B. CityPersons: A Diverse Dataset for Pedestrian Detection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, (CVPR 2017)*, Honolulu, HI, USA, July 21-26, 2017, 4457-4465. <https://doi.org/10.1109/CVPR.2017.474>

