Contents lists available at YXpublications

International Journal of Applied Mathematics in Control Engineering

Journal homepage: http://www.ijamce.com

Improved Algorithm of Spatio-Temporal Action Localization Based on YOWO

Yue Zhuo, Xingguo Song*, Zhongqing Cao, Sheng Jin

Mechanical Engineering School, Southwest Jiaotong University, Chengdu 610031, China

ARTICLE INFO

Article history:
Received 12 October 2023
Accepted 24 December 2023
Available online 5 January 2024

Keywords:
YOWO
Spatio-temporal action localization
One-stage
Feature pyramid

ABSTRACT

Spatio-Temporal action localization is a task of classifying the corresponding action category in a sequence frame and locating its position in each frame. You only watch once (YOWO) is an excellent one-stage algorithm for action classification and location based on 2D CNN and 3D CNN, which has fast inference speed. However, its identification precision needs to be further improved in some practical applications. In order to improve the identification precision. We introduce the multi-scale idea of the feature pyramid. In terms of sampling method, we propose a method that randomly select the key frame to obtain local features in consecutive frames, and jointly predict the each frame with global features extracted from the 3D network. The proposed algorithm's Frame-mAP can achieve 81.6% and 87.0% on jhmdb-21 and UCF101-24 respectively, which has an impressive improvement of 7.2% and 6.6% compared with YOWO algorithm. Besides, the inference speed can reach 15 fps only on the GPU of GTX1660Ti. Compared with other state-of-the-art architectures, our method also achieves competitive performance.

1. Introduction

Spatio-temporal action location is a complex task which needs classify the action and locate the actor by bounding boxes on each frame of the input video stream. The popular object detection algorithm, like Faster-RNN[1] and Yolov3[2], which can classify and locate objects from video stream. These models can be roughly divided into two-stage mode and one-stage methods. The two-stage method, like Faster-RNN, which generates dense region proposal in each frame and intercepts features for classification, and then uses the action tube algorithm to obtain the best action pipeline. However, its inference speed is so slow that it is difficult to meet the needs of real-time detection. One-stage algorithm, like Yolov3, which generates location and classification information at one time, can be trained by end-to-end. Its inference speed is much faster than the two-stage algorithm. However, action recognition from single frame based on 2D CNN cannot represent association information between actions, which leads to poor recognition effect.

In order to learn more effective feature, 3D CNN needs to be considered. The YOWO[3] is a more efficient one-stage algorithm for spatio-temporal action location, which is based on the idea that the previously received data needs to be considered first when located and classified the current frame. As shown in Figure 1, the author proposed a method that extract the global information on continuous frames of fixed length by 3D-CNN, and extract local information on the last frame by 2D-CNN, then fuse the local and global information by attention modules, finally, predict the location and classification information of the last frame just like the Yolov2[4] algorithm. YOWO is well-known for its high accuracy and fast inference speed.

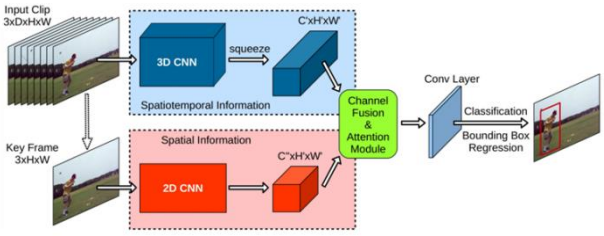

Fig. 1. YOWO network structure

However, the algorithm also has the following disadvantages:

1. Only take the last frame of the clip as key frame. This strategy is not friendly to the prediction of the frame at the front of video streaming. Because of the lack of timing information in front of the key frame, the current frame can only be predicted by using a cyclic sequence.
2. Only the feature information of the last layer of the 2D

* Corresponding author.
E-mail addresses: songxg@swjtu.edu.cn (X. Song)
Digital Object Identifiers: https://doi.org/10.62953/IJAMCE.590525

network is used. This layer is obtained by 32 times down sampling of the original image, the resolution of the feature map is low, so that it is difficult to regress accurate parameters of bounding box.

3. The feature maps of 2D-CNN and 3D-CNN are directly stacked, and then the channel attention module in Danet[5] is used to strengthen the feature extraction. However, the single-scale fusion strategy loses multi-scale information.

In view of the above disadvantages, we propose the following improvement methods.

1. Firstly, we set any frame of the current clip as the key frame, then we can predict the information of any frame in one video without destroying the timing of the video. For a continuous N frames of video stream, the prediction result of each frame can be obtained as follows: input N frames into the 3D-CNN to obtain global features, and use the 2D-CNN as a sliding window, obtain local features frame by frame and jointly predicted the result of the current frame with global features.
2. Secondly, we select the last three feature layers of 2D-CNN to achieve multi-scale feature, and replace single head prediction with multi-scale prediction.
3. Thirdly, we propose a method called secondary fusion. for the three 2D-features, they are separately fused with 3D features to gain features of three scales, and then FPN[6] is used to further aggregate the multi-scale features. This structure decrease network parameters and maintain fast inference speed.

The remainder of the paper is organized as follows. In Section 2, we review the related work for action recognition and spatio-temporal action localization. In Section 3, we introduce the network structure and the improvement methods in detail. In Section 4, the experimental results are presented with additional discussions. Finally, the paper is concluded in Section 5.

2. Related works

2.1 Action recognition

With the development of deep learning, CNN as a powerful feature extractor is gradually applied to action recognition. Frame-by-frame extraction of CNN information is used in action classification[7,8], while two-stream structure[9] combined RGB image and optical flow features can further improve the accuracy of classification. Recently, 3D ConvNets[10,11,12], extended by 2D networks, have been proved to be more effective in extracting spatio-temporal features. The I3D network[13] further improved the 3D convolution technique by expanding the convolution kernel of the 2D-CNN network pre-trained on ImageNet into a 3D convolution kernel, and proposed a kinetics dataset to solve the transfer in action recognition learning problems.

In order to extract more effective features, Wang et al. [14] even combine 3D-CNN and optical flow features. Although 3D convolution network is more efficient in feature extraction, it also has some disadvantages, such as large amount of parameters and expensive training cost. Some researchers have begun to explore ways to improve the 3D network structure: P3D[15] proposes three different space and time separation convolution methods to reduce the parameter of the 3D convolution kernel, R(2+1)D[16] shows that 2D convolution and 1D convolution are sufficient to learn the discriminative features of action recognition, S3D[17] tries to separate 2D and 3D convolution operations and add attention modules.

2.2 Spatio-temporal Action Localization

The task of spatio-temporal action localization is more complicated than action recognition which requires correct classification and accurate location of actors within the time interval when the behavior occurs. Most of the current spatio-temporal action localization algorithms are based on two stages [18,19,20,21,22,23,24,25,26]: RPN(Region Proposal Network) is used to generate dense region proposals in each frame for clips, then Roipooling layer is used to intercept feature map to make further classification, finally, a link algorithm is used to connect the boxes to form action tubes.

For the one-stage algorithm, the method is generally not consistent. Many methods are based on one-stage object detection networks. ACT[27] build upon the SSD[28] framework, which extracts features frame by frame 2D networks and stacks them for regression and classification. MOC[29] build upon the CenterNet[30] framework, which directly predict the position of the moving point, the motion trajectory between adjacent frames and the bounding box parameters.

Some scholars pay attention to the fine-tuning and aggregation of features: CFAD[31] uses 3D features for action classification and coarse-grained action location, then use a time suggests module to fine-tuned the key frame location information to achieve higher location accuracy; STEP[32] performs frame-by-frame classification and location, and uses the time-gradual module to gradually fine-tune features to extend local information to global information. Tang et al.[33] proposed the asynchronous interaction aggregation network (AIA),which used object detector and 3D-CNN to obtain global and local information, and interactively fusion features to achieve better performance in dense actions scene; Ref. [34,35] used graph neural network to get the correlation between the action and the scene.

In addition to using convolution networks, some scholars have found that RNN and its variants perform well in action spatio-temporal location: Akshaya Ramaswamy et al.[36] proposed a method of extracting features by I3D and obtain spatio-temporal modeling by LSTM. Song et al.[37] used BLSTM, a type of bidirectional RNN, to learn the multimodal features that share context information of adjacent clips.

Recently, the self-attention models have become popular, even the transformer[38], which is completely based on the attention mechanism, can replace the convolutional network to achieve good results: Rizard Renanda Adhi Pramono et al proposed a structure that initial action detection by the two-stream CNN[39] or 3D-CNN[40], and then learn the spatio-temporal relationship of actors by hierarchical self-attention module(HiSAN), Rohit Girdhar et al[41] proposed a Transformer-style architecture to aggregate features from the spatiotemporal context.

Compared with other algorithms, YOWO has practicability clear structure and great possibility for improvement. We start with YOWO. YOWO is also the one-stage algorithm, which combines 2D-CNN, 3D-CNN and attention mechanism. Though YOWO can meet real-time inference speed, the prediction accuracy needs to be improved. In this paper, we make improvements based on YOWO so that it can be truly applied to actual engineering projects.

3. Methods

3.1 Video sampling method

In YOWO, continuous frames will be input to the I3D network to obtain the spatio-temporal features first, then take the last frame as

the key frame and is input to the 2D network to obtain spatial features, finally the classification and regression prediction of key frame are performed after the features are fused. We generalize it to a more general form, where any frame in the clips can be regarded as a key frame. However, it can bring a problem that how to ensure the sampling balance, that is, the probability of the network getting any positions of key frame is basically equal.

For a video sequence $[v_1, v_2, ..., v_i, ..., v_{L_{\max}}]$, select any frame as the key frame, $k_{cur} \in [1, L_{\max}]$. Sample the video clips $[x_1, x_2, ..., x_L]$ in this video sequence, where the key frame is the j-th frame of this clip. Besides, the sampling step of the video clip is p. So the clip sequences is described as $clip = [v_{k_{cur}-(j-1)p}, v_{k_{cur}-jp}, ..., v_{k_{cur}+(L-j)p}]$. In order to ensure the validity of the clips, there should be:

$$\begin{cases} k_{cur} - (j-1)p \geq 1 \\ k_{cur} + (L-j)p \geq L_{\max} \end{cases} \tag{1}$$

Then the upper and lower bounds of j can be obtained as:

$$\begin{cases} j_{\min} = L - \left\lfloor \dfrac{L_{\max} - k_{cur}}{p} \right\rfloor \\ j_{\max} = \left\lfloor \dfrac{k_{cur} - 1}{p} \right\rfloor + 1 \end{cases} \tag{2}$$

Restrict the upper and lower bounds of j to be within the subscript of the clip sequence, we can get:

$$\begin{cases} j_{\min} = \max(1, j_{\min}) \\ j_{\max} = \max(1, j_{\max}) \end{cases} \tag{3}$$

During the training process, we set $j = random(j_{\min}, j_{\max})$; During the testing process, we set $j = j_{\max}$. Clip sequence is $clip \in [x_1, x_2, ..., x_L]$, and its length is L. The prediction result for j-th frame is described as below:

$$p_j = f_{fuse}(f_{2d}(x_j) + f_{3d}(x_1, x_2, ..., x_L)), j \in [1, L] \tag{4}$$

where f_{fuse} represents the fusion network, f_{2d} represents the 2D backbone extraction network, and f_{3d} represents the 3D backbone extraction network.

In this way, any frame in one clip can be used as a key frame, the balance of sampling can be guaranteed in the random way. At the same time, this method also indirectly expands the data field which can increase the robustness of the network. For example, for 16-frames clips, any frame can be input into the network as a key frame. Compared with taking only the last frame, the data field is expanded by 16 times. Finally, this method forces 2D-CNN to learn the location information of key frames, while the 3D network essentially affects the classification prediction. The prediction results of key frames are affected by both the front frames and the rear frames, that is in line with the way people watch videos.

3.2 Network structure

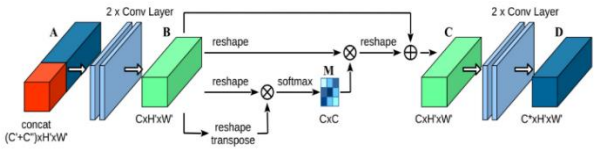

Fig. 2. the fusion module in YOWO

As shown in Figure 2, in YOWO, the last layer of features of the 2D network and the 3D network are used for stacking, and the prediction results are obtained after several convolutions and attention mechanisms. This method is simple and the information extracted is limited, so we designed a secondary fusion module to replace it. This structure can also learn multi-scale prediction which is helpful to obtain more accurate location information. We try to add its attention mechanism to our structure, but it reduces the performance of the network, so we abandoned the attention mechanism in the fusion module. Although the structure is more complex, the network uses many 1×1 convolution to compress channels, so the overall parameters will not increase. The overall network structure is shown in Figure 3.

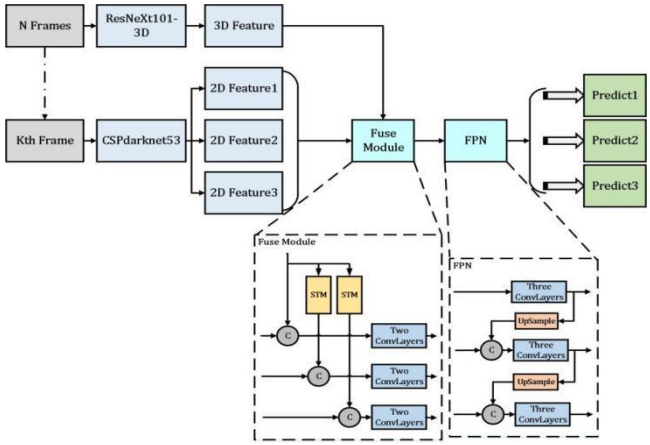

Fig. 3. Overall network structure

As shown in Figure 3, the C represents concatenation, the ConvLayer is composed of convolution and LeakyReLu and Batchnormal, the UpSample is composed of a 1x1 convolution and adjacent upsampling, and the STM represents scale transfer module.

In the feature extraction part, the I3D network is used to extract spatio-temporal information which is pretrained on the kinetics dataset. For the 2D network, we replace the darknet19 in YOWO with the CSPdarknet53. CSPdarknet53 has a CSP structure[42], which can greatly reduce the amount of network parameters and increase the inference speed.

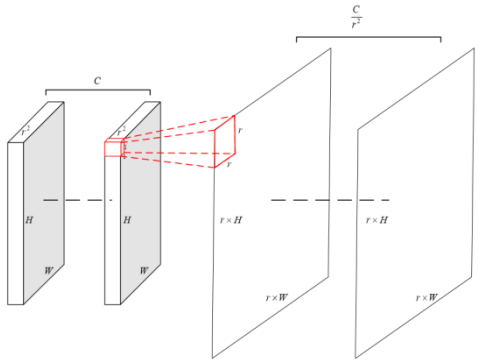

Fig. 4. STM (scale transfer)

For the 224×224 input, the 3D network mainly contributes to the classification information and doesn't require accurate location information, so we only take the low-resolution feature map obtained by the last convolution, which has higher semantic information. For the 2D network, because the accurate location information of the key frame needs to be obtained, we take the last three layers of 2D network as effective features, which are the results of 8 times, 16

times, and 32 times downsampling of the original image. For the three feature maps, the high-resolution feature map has a small receptive field due to its low downsampling ratio, which is suitable for detecting small object, while the low-resolution feature map has a large receptive field due to its high downsampling ratio, which is suitable for detecting large object.

Fuse module is applied in the feature fusion, as shown in Fig. 3. The 3D feature needs to be sampled to the same size as the 2D features first, where a STM[43] structure is implemented. As shown in Figure 4, STM is a feature rearrangement technology, each point of the channel is rearranged on the $r \times r$ plane, so as to achieve the effect of magnifying the resolution of the feature map by r times, and it won't introduce parameters without loss of information. For the upsampled 3D features, we stack them with 2D features separately, then fuse the stacked features after two convolutions. In the second stage of fusion, we use the feature pyramid to further integrate the spatio-temporal features at different resolutions. As show in "FPN" of Fig. 3, low-resolution features are stacked with high-resolution feature maps after three convolutions and nearest upsample. Finally, we learn from the prediction layer of yolov3, 3×3 convolution is performed to integrate the features, and 1x1 convolution is performed to change the channels to $L = 3(5 + n_c)$,where n_c represents the number of classes. The location information and classification information of the key frame will be predicted in the same time.

3.3 Loss function

In YOWO, the smoothL1loss is used as the regression loss, MSEloss as the confidence loss, and the multi-class Facolloss[44] as the classification loss. We make the following improvements: We use IOU as the regression loss. Compared with the smoothL1loss, IOU loss has scale invariance to the bboxes, which is helpful to train better detectors. so the ciouloss[45] is used which is an improved version of IOU loss. We try to apply muti-class Focalloss to our multi-scale prediction structure, the classification loss will be very small during the training process, so that the percentage of each loss are extremely imbalanced, this leads to poor training results. on the other hand, softmax function and multi-classes loss are used in YOWO, which is conducive to the prediction of single label categories. Although UCF101-24 and jhmdb-21 datasets are all single-label scenarios, but in actual scenarios, a person often has multiple actions at the same time. Therefore, we all use binary focalloss as the confidence loss and classification loss, and the sigmoid function is used to convert the output into probability distribution. The definition of each loss will be described in detail below.

For the regression loss, the distance between the center point and the aspect ratio are considered in ciouloss by comparing with the iouloss, which can make bounding box regression more accurately and faster. It is defined as:

$$L_{ciou} = 1 - IOU + \frac{\rho^2(b, b^{gt})}{c^2} + \alpha\nu \tag{5}$$

where *IOU* represents the intersection ratio of the truth box and the predicted box, and $\rho(b, b^{gt})$ represents the center point distance between the truth box and the predicted box, c represents the diagonal distance of the smallest circumscribed rectangle, and ν is the aspect ratio penalty which can be described as:

$$\nu = \frac{4}{\pi^2}(\arctan\frac{w^{gt}}{h^{gt}} - \arctan\frac{w}{h})^2 \tag{6}$$

where w^{gt} , h^{gt} epresents the width and height of the truth box, w and h represents the width and height of the predicted box.

For confidence loss and classification loss, we start out with the binary cross-entropy loss, but classification loss is difficult to decrease in the middle of training, and the performance is poor, so we use binary FocalLoss instead. We further compare the effects of the two loss functions, we train 40 epochs for 8-frames-clip on jhmdb-21 dataset. Except for the loss function, the other parameters remain the same. The confidence threshold is 0.005 and the IOU threshold is 0.4 in non maximal suppression, and the evaluated IOU threshold is 0.5 to distinguish between positive and negative samples, the experimental results are shown in table 1, we reported the metrics of recall and Frame-mAP. The recall of BCE loss is relatively low, which further affects the Frame-mAP.

Tab. 1. recall and Frame-mAP@0.5 between BCE loss(binary cross-entropy loss) and binary Facol Loss on jhmdb-21.

Loss fuction	NMS		mIOU	recall	Frame-mAP
	conf thresh	iou thresh			
BCE Loss	0.005	0.4	0.5	84.3	67.7
BFocal Loss	0.005	0.4	0.5	92.9	77.4

The FocalLoos is defined as Equation (7):

$$FL(p_t) = -\alpha_t(1 - p_t)^{\gamma}\log(p_t) \tag{7}$$

where α_t and γ are all hyperparameters, γ is used to mine difficult samples, α_t control the balance of positive and negative samples, p_t represents the predicted score.

Equation (7) can be written as binary Facolloss:

$$\begin{aligned} BFL(p_t, g_t, \alpha_t, \gamma) = &-\alpha_t(1 - p_t)^{\gamma} g_t \log(p_t) \\ &-(1 - \alpha_t) p_t^{\gamma}(c)(1 - g_t)\log(1 - p_t) \end{aligned} \tag{8}$$

where g_t represents the true label. For positive samples, $g_t = 1$; for negative samples, $g_t = 0$.

According to Equation (8), the classification loss and confidence loss can be described as Equation (9) and Equation(10):

$$L_{cls} = \sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathrm{I}_{i,j}^{obj} \sum_{c \in classes} BFL(p_t(c), g_t(c), \alpha_t, \gamma) \tag{9}$$

$$\begin{aligned} L_{conf} = &\sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathrm{I}_{i,j}^{obj} BFL(c_t, 1, \alpha_t, \gamma) \\ &+\sum_{i=0}^{S^2}\sum_{j=0}^{B} \mathrm{I}_{i,j}^{nobj} BFL(c_t, 0, \alpha_t, \gamma) \end{aligned} \tag{10}$$

where S represents the grid, B represents the number of priori boxes, $p_t(c)$ represents the predicted category probability of the grid point, and $g_t(c)$ represents the true category probability, c_t represents the probability that the grid point contains an object. For the classification loss, we use the default parameters in the paper of Facolloss, that is, $\alpha_t = 0.25, \gamma = 2$ .For the confidence loss, our prediction mode is similar to yolov3, which already have the methods to balance the positive and negative samples, so we set $\alpha_t = 0.5$ so that the weight ratio of positive and negative samples is 1:1. For the final conv layer of the classification and confidence subnet, the bias should be initialization to $b = -\log((1 - \pi)/\pi)$, where $\pi = 0.01$ [44].

in this way, all prediction results will be negative samples at the beginning of training, which is conducive to the convergence of the network.

Furthermore, the input of the spatio-temporal action location network is not a single RGB image, but a time sequence of a certain length, so the training process is expensive. In order to speed up the convergence, we also fine-tuned the method of positive and negative sample division strategy for training based on yolov3. In yolov3, each ground truth box can only be assigned to one priori box which has the highest iou among the three feature layers, this sampling rule is too strict for action recognition. So we change this rule: for the topmost feature map, only one box with the highest iou with the nine priori boxes is matched; for the second layer feature map, look for the two boxes with the highest iou with the nine priori boxes for prediction in turn; for the third layer, look for the three boxes with the highest iou with 9 priori boxes in turn, if they are not present, this layer will not be responsible for prediction. In this way, the number of positive samples for each training step increase which can effectively speed up training. and the features closer to the bottom layer have a greater probability of making more predictions, which is in line with the law that the bottom layer will get more semantic information in top-down pyramid. What's more, the resolution of the bottom feature map is higher, the original image is divided into more grid cells, and it is difficult to have intensive labels for action recognition, so it will not increase the label overwrite[46], which means that center points of truth box coincide when image downsample as feature map, so that the prediction is made by the same grid and the previous boxes is ignored.

3.4 Other improvements

The parameters of network can reach nearly 500M in YOWO when we use resnext101 as a 3D network, it is easy to overfit during the training process. In the process of training the original network of YOWO, the 10th iteration of the network has basically reached the desired effect, the following training will not bring the significant improvement. Therefore, some regularization methods to prevent over-fitting are necessary. In addition to basic data augmentation, we use dropblock to prevent over-fitting. At the same time, we add the attention mechanism to the detection head to further improve the performance of the network.

3.4.1 Attention mechanism

The attention mechanism is widely used in sequence models, which can significantly improve the accuracy of multi-classification networks. In recent years, the attention mechanism has also begun to be applied to convolutional neural networks, such as the SE module[47], CBAM module[48], which are used in image classification and object detection. Here we have added the SE module to the detection head to perform channel selection and channel suppression on the fused features to obtain more delicate features, and only increase the amount of 1M parameters. The structure of the SE module can be divided into two steps: compression and excitation. As show in Figure 5, compression is obtained by performing global average pooling on the feature layer to obtain the global compressed feature of the current feature map; excitation is obtained the weight of each channel in the feature map through a bottleneck structure of two-layer fully connection, which is weighted to the original feature map next.

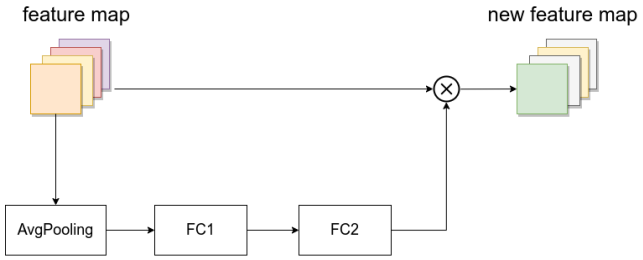

Fig. 5. SE attention model

3.4.2 Dropblock

Dropblock[49] is a strong regularization method for preventing over-fit. we add it to the structure of FPN. As show in Fig. 6, compared with dropout discarding points in the feature map, dropblock discards a region in the feature map which can effectively prevent overfitting.

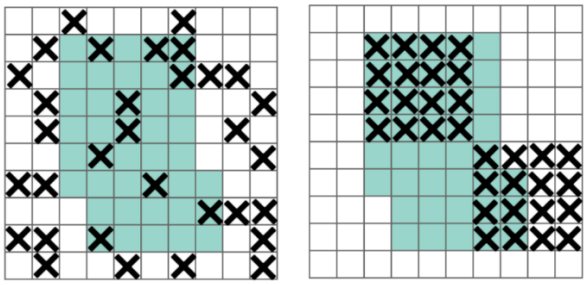

Fig. 6. (a) is the ordinary dropout, which represents discarding points randomly; (b) is the dropblock, which represents discarding a region of the feature map randomly.

4. Experimental Details

4.1 Dataset and Experimental parameters

We evaluate the results on the benchmark datasets jhmdb-21 and UCF101-24. For the jhmdb-21, it contains 22712 training labels and 9126 test labels, and a total of 21 actions. For the UCF101-24, it contains 337835 training labels and 137558 test labels, and a total of 24 actions.

The experimental platform is driven by a single GPU of Nvidia 1660ti. We use the SGDM optimizer with an initial learning rate of 0.0001 and the momentum of 0.937. For the jhmdb-21, a total of 40 epochs are performed. We freeze 2D and 3D network at the beginning of training, unfreeze the 3D network in the 10th epoch, and unfreeze the 2D network in the 25th epoch. The learning rate is halved in the 10th, 18th, 25th, and 35th epoch; For the UCF101-24, we train a total of 20 epochs, and we freeze 2D and 3D network at the beginning of training, unfreeze the 3D network in the 5th epoch, and unfreeze the 2D network in the 15th epoch. For all the datasets, the data augmentation we used include random horizontal flipping, random zoom, spatial jitter and color jitter. The range of random zoom value for images is between 0.5~1.5 and color jitter is between 0.7~1.3. It should be noted that for each RGB images in the same clip, the parameters used for data augmentation should be consistent, otherwise the coherence of the action in spatio-temporal dimension will be destroyed. The RGB images input to the network are all resized to 224×224.

4.2 Comparison with YOWO

In order to prove that our method is effective, we did the

experiment on the dataset jhmdb-21. In this experiment, we conduct experiments with 8-frames clips. We use the YOWO as baseline, we get an improved version of YOWO by replacing the backbone network and loss function first, and then gradually add new modules to improve the network for experimental control. Furthermore, we take Frame-mAP@0.5 as evaluation index, the IOU threshold is 0.4 and the confidence threshold is 0.005 in non maximum suppression. The experimental results are shown in table 2.

Tab. 2. YOWO* is the result of replacing YOWO's 2D feature extraction network darknet19 with cspdarknet53 and introducing ciouLoss and binary FacolLoss. KFRS represents the random sampling of key frames designed in this paper. SecondFuse is the secondary fusion module, Frame- mAP is the test result of jhmdb-21 test dataset.

Methods	Frame-mAP(%)	Parameters
YOWO	64.9	462M
YOWO*	71.1	372M
+KFRS	73.6	372M
+SecondFuse	76.3	373M
+dropblock	77.0	373M
+SE	77.4	373M

As shown in the table 2, YOWO's original network Frame-mAP is 64.9%, YOWO* replaces the 2D backbone network and loss function, gains 6.2% improvement, Frame-mAP reaches 71.1%; introduce KFRS , that is to improve from the sampling rules during training, Frame-mAP increased by 2.5% to 73.6%; the single-head fusion structure of YOWO was replaced with a secondary fusion module , Frame-mAP increased by 2.7%; the dropblock module continued to be introduced, to prevent network overfitting, Frame-mAP increased by 0.7%; finally the SE attention mechanism was introduced, the parameter increased by 1M, and Frame-mAP increased by 0.4%. From the above analysis, we can see that for 8-frames clips, we have achieved a 12.5% improvement compared to YOWO on the jhmdb-21.

In order to verify the effectiveness of the improved positive and negative division strategy, we take 8-frames clips for training, and train 40 epoch in total. With the growth of epoch, we separately recorded the changes of Frame-mAP@0.5 under the yolov3's division strategy and our division strategy. As shown in Figure 7, our strategy is obviously faster than yolov3's.

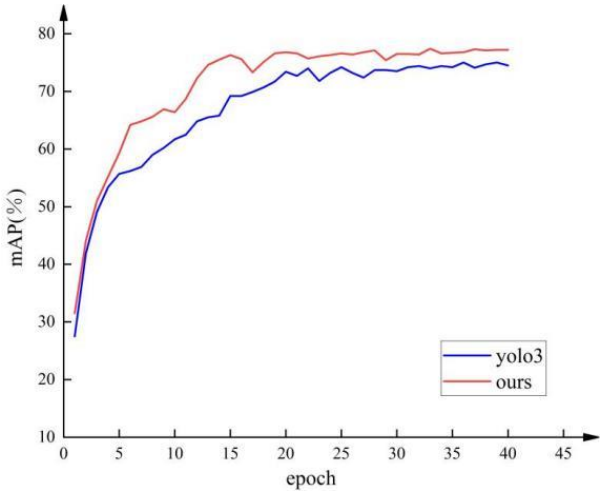

Fig. 7. The blue and red lines represent the growth curve of Frame-mAP with the number of iterations under yolo3's and the improved positive and negative sample division strategy, respectively.

We also compare Frame-mAP of our network with the best result of YOWO on the datasets jhmdb-21 and UCF101-24, the input image size is still 224×224. The results are shown in table 3.

Tab. 3. Compare Frame-mAP@0.5 between ours and YOWO under 8-frames and 16-frames input.

method	Jhmdb-21		UCF101-24	
	8	16	8	16
YOWO	64.9	74.4	79.2	80.4
ours	77.4	81.6	85.9	87.0

As shown in the table 3, on the jhmdb-21, YOWO's Frame-mAP reached 64.9% for the input of 8-frames clips, while our Frame-mAP can increase by 12.5% to 77.4%; for 16-clip input, YOWO's mAP reached 74.4%, it can be seen that it is even smaller than ours with 8-frame as input. The best result we achieved with 16-frames clips is 81.6% Frame-mAP, which has increased by 7.2% compared to YOWO. on the UCF101-24, the Frame-mAP for YOWO reached 79.2% for 8-frames clips while reached 80.4% for 16-frames clips, and we reached 85.9% and 87.0% with an impressive improvement of 6.7% and 6.6%.

Finally, we compared the inference speed with YOWO, all results are run on a single GPU of GTX1660Ti. As shown in table 4, through the improvement of the network structure, the network parameters reduced from 462M to 373M, the FLOPs of network will increase slightly, and the inference speed of our network can reach 10 frames/s for 16-frames input while reach 15 frame/s for 8-frames input, which is almost equal to YOWO.

Tab. 4. Compare inference speed between ours and YOWO under 8-frames and 16-frames input.

	Clip	2d-backbone	3d-backbone	Parameters	GFLOPs	Run time
YOWO	8	darknet19	resnext101	462M	24.9	63.4ms
ours	8	cspdarknet53	resnext101	373M	27.0	63.8ms
YOWO	16	darknet19	resnext101	462M	43.6	98.1ms
ours	16	cspdarknet53	resnext101	373M	45.8	95.0ms

4.3 Comparison with state-of-the-art

We have compared our method with other state-of-the-art architectures on jhmdb-21 and UCF101-24. Using the standard metrics, we report the frame-mAP at IOU threshold 0.5 and the video-mAP at various IOU thresholds, our method achieves competitive performance under different tested threshold criterion. In some respects, our method achieves state-of-the-art.

4.3.1 Comparison on dataset J-HMDB-21

As shown in table 5, on dataset J-HMDB-21, in term of Frame-mAP@0.5, Video-mAP at IOU threshold 0.2 and Video-mAP at IOU threshold 0.5, we all lag slightly behind HiSAN. while the threshold is 0.75, Video-mAP still has 68.3%, which is 5.6% higher than HiSAN, but 6.7% lower than MOC. In general, compared with other state-of-the-art architectures, our method is still competitive.

Tab. 5. Performance on dataset J-HMDB-21 and comparison with SOTA results by frame-mAP under IOU threshold 0.5 and video-mAP under different IOU thresholds.

Method	Frame-mAP	Video-mAP		
		0.2	0.5	0.75
ROAD(2017)[19]	-	73.8	72.0	44.5
ACT(2017)[27]	65.7	74.2	73.7	52.1
TPnet(2018)[20]	-	74.8	74.1	61.3
YOWO(2019)[3]	74.4	87.8	85.7	58.1
CA RCNN(2020)[25]	79.2	-	-	-
CFAD(2020)[错误!未找到引用源。]	-	84.8	83.7	62.4
MOC(2020)[29]	74.0	80.7	80.5	**75.0**
HiSAN(2021)[40]	**85.4**	**93.9**	**91.8**	62.7
ours	81.6	87.7	87.5	68.3

4.3.2 Comparison on dataset UCF101-24

Tab. 6. Performance on dataset UCF101-24 and comparison with SOTA results by frame-mAP under IOU threshold 0.5 and video-mAP under different IOU thresholds.

Method	Frame-mAP	Video-mAP		
		0.2	0.5	0.75
ROAD(2017)	-	73.5	46.3	15.0
ACT(2017)	65.7	77.2	51.4	22.7
YOWO(2019)	80.4	75.8	48.8	-
CFAD(2020)	-	79.4	62.7	-
MOC(2020)	78.0	82.8	53.8	**29.6**
HST-LSTM(2020)[36]	82.4	87.2	-	-
HiSAN(2021)	80.3	**88.6**	66.4	29.3
ours	**87.0**	84.7	**74.0**	26.6

As shown in table 6, on dataset UCF101-24, we have achieved SOTA in most indicators. For the Frame-mAP@0.5, Our method outperforms the state-of-the-art results. For the Video-mAP, our method achieves SOTA at the IOU thresh of 0.5, it is worth mentioning that we are 7.6% ahead of the second place HiSAN, our method is slightly behind the HiSAN at the thresh of 0.75.

4.4 Discussion on shortcomings

Fig. 8. Visualization of some key frames' detection results on jhmdb-21 test dataset

Fig. 9. Visualization of some key frames' detection results on UCF101-24 test dataset

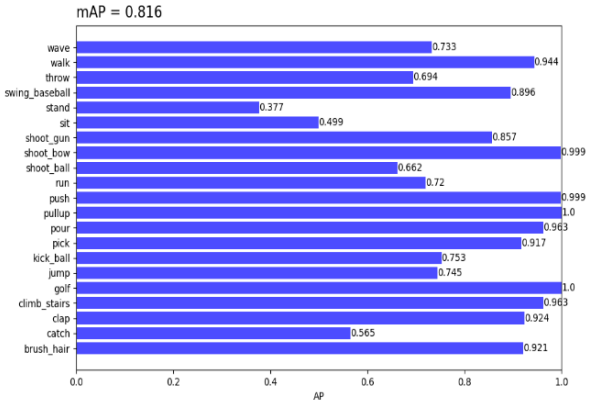

Fig. 10. 21 AP distribution of actions on the jhmdb-21

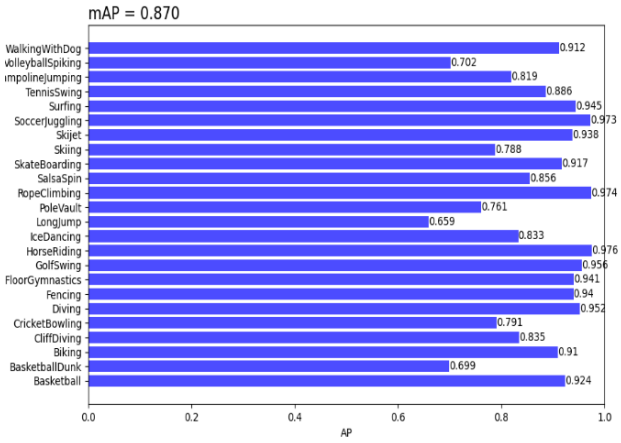

Fig. 11. 24 AP distribution of actions on the UCF101-24

Normally, our network has high location accuracy and classification accuracy, which is illustrated in Figure 8 and Figure 9. We also report the AP of each category on jhmdb-21 and UCF101-24 in Figure 10 and Figure 11. Due to the high similarity between classes in jhmdb-21, the AP of some categories in jhmdb-21 is relatively low, such as sit and stand.

In order to further analyze and evaluate the performance of our algorithm, and find out the problems in the classes of low AP, we report the confusion matrix on jhmdb-21 and UCF101-24. The detected results are based on the confidence threshold of 0.5 and the IOU thresh of 0.4. As show in Fig.11 and Fig.12, the vertical axis represents the real category and the horizontal axis represents the predict category. On jhmdb-21, 24% of the actual prediction results are stand for groundtruth boxes of sit, and 54% of the actual

prediction results are sit for groundtruth boxes of stand. There are the same problem for catch and shoot ball, it is difficult to distinguish between the two actions, while there is no similar opposite action sequence in UCF101-24, the classification of each category works well. So the network may be difficult to distinguish between these two sequences with opposite timing.

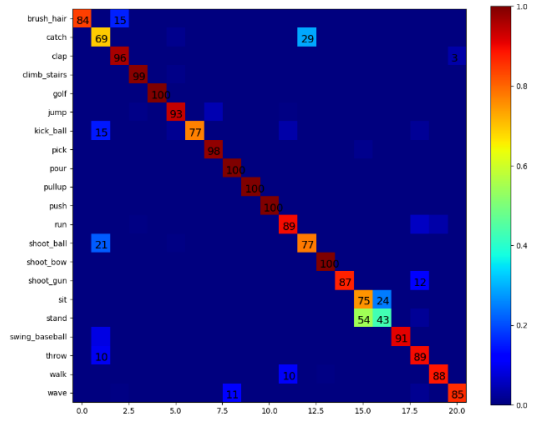

Fig. 12.. confusion matrix on jhmdb-21

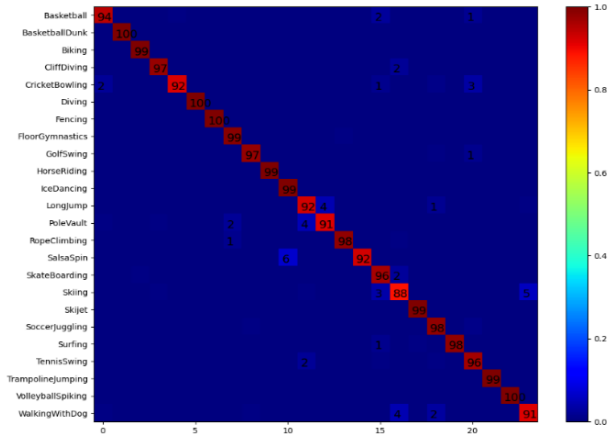

Fig. 13. confusion matrix on UCF101-24

5. Conclusion

Based on the idea of YOWO, the following improvements are proposed in this paper. For sampling method, we have extended the key frame to any frame to increase the robustness of the network. We also have improved the network structure, and proposed a secondary fusion module that fuses 2d feature and 3d feature to obtain spatio-temporal characteristics by STM first, then fuses the features again to obtain multi-scale features by FPN. Finally, the loss function and the method of positive and negative samples division strategy are redesigned. We have evaluated the improved network on the jhmdb-21 and UCF101-24, the accuracy is greatly improved and inference speed is not decreased by comparing with YOWO algorithm, which can be applied to some practical projects for spatio-temporal action localization in real time. However, our algorithm has some shortcomings which is not very sensitive to temporal information, and will be improved in the future work.

References

[1] Faster R. Towards real-time object detection with region proposal networks[J]. Advances in neural information processing systems, 2015, 9199(10.5555): 2969239-2969250.

[2] Redmon J, Farhadi A. Yolov3: An incremental improvement[J]. arXiv preprint arXiv: 1804.02767, 2018.

[3] Köpüklü O, Wei X, Rigoll G. You only watch once: A unified cnn architecture for real-time spatiotemporal action localization[J]. arXiv preprint arXiv: 1911.06644, 2019.

[4] Redmon J, Farhadi A. YOLO9000: better, faster, stronger[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 7263-7271.

[5] Fu J, Liu J, Tian H, et al. Dual attention network for scene segmentation[C]// Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 3146-3154.

[6] Lin T Y, Dollár P, Girshick R, et al. Feature pyramid networks for object detection[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2017: 2117-2125.

[7] Karpathy A, Toderici G, Shetty S, et al. Large-scale video classification with convolutional neural networks[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2014: 1725-1732.

[8] Yue-Hei Ng J, Hausknecht M, Vijayanarasimhan S, et al. Beyond short snippets: Deep networks for video classification[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 4694-4702.

[9] Simonyan K, Zisserman A. Two-stream convolutional networks for action recognition in videos[J]. arXiv preprint arXiv:1406.2199, 2014.

[10] Tran D, Bourdev L, Fergus R, et al. Learning spatiotemporal features with 3d convolutional networks[C]//Proceedings of the IEEE international conference on computer vision. 2015: 4489-4497.

[11] Hou R, Chen C, Shah M. Tube convolutional neural network (t-cnn) for action detection in videos[C]//Proceedings of the IEEE international conference on computer vision. 2017: 5822-5831.

[12] Saha S, Singh G, Cuzzolin F. Amtnet: Action-micro-tube regression by end-to-end trainable deep architecture[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 4414-4423.

[13] Carreira J, Zisserman A. Quo vadis, action recognition? a new model and the kinetics dataset[C]//proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017: 6299-6308.

[14] Wang X, Gao L, Wang P, et al. Two-stream 3-d convnet fusion for action recognition in videos with arbitrary size and length[J]. IEEE Transactions on Multimedia, 2017, 20(3): 634-644.

[15] Qiu Z, Yao T, Mei T. Learning spatio-temporal representation with pseudo-3d residual networks[C]//proceedings of the IEEE International Conference on Computer Vision. 2017: 5533-5541.

[16] Tran D, Wang H, Torresani L, et al. A closer look at spatiotemporal convolutions for action recognition[C]//Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2018: 6450-6459.

[17] Xie S, Sun C, Huang J, et al. Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification[C]//Proceedings of the European Conference on Computer Vision (ECCV). 2018: 305-321.

[18] Gkioxari G, Malik J. Finding action tubes[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 759-768.

[19] Gkioxari G, Malik J. Finding action tubes[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2015: 759-768.

[20] Singh G, Saha S, Cuzzolin F. Predicting action tubes[C]//Proceedings of the European Conference on Computer Vision (ECCV) Workshops. 2018: 0-0.

[21] Peng X, Schmid C. Multi-region two-stream R-CNN for action detection[C]//European conference on computer vision. Springer, Cham, 2016: 744-759.

[22] Saha S, Singh G, Sapienza M, et al. Deep learning for detecting multiple space-time action tubes in videos[J]. arXiv preprint arXiv:1608.01529, 2016.

[23] Zhao J, Snoek C G M. Dance with flow: Two-in-one stream action detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 9935-9944.

[24] Hou R, Chen C, Shah M. An end-to-end 3d convolutional neural network for action detection and segmentation in videos[J]. arXiv preprint arXiv:1712.01111, 2017.

[25] Wu J, Kuang Z, Wang L, et al. Context-aware rcnn: A baseline for action detection in videos[C]//European Conference on Computer Vision. Springer, Cham, 2020: 440-456.

[26] Gleason J, Ranjan R, Schwarcz S, et al. A proposal-based solution to spatio-temporal action detection in untrimmed videos[C]//2019 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE, 2019: 141-150.
[27] Kalogeiton V, Weinzaepfel P, Ferrari V, et al. Action tubelet detector for spatio-temporal action localization[C]//Proceedings of the IEEE International Conference on Computer Vision. 2017: 4405-4413.
[28] Liu W, Anguelov D, Erhan D, et al. Ssd: Single shot multibox detector[C]//European conference on computer vision. Springer, Cham, 2016: 21-37.
[29] Li Y, Wang Z, Wang L, et al. Actions as moving points[C]//European Conference on Computer Vision. Springer, Cham, 2020: 68-84.
[30] Zhou X, Wang D, Krähenbühl P. Objects as points[J]. arXiv preprint arXiv:1904.07850, 2019.
[31] Li Y, Lin W, See J, et al. CFAD: Coarse-to-Fine Action Detector for Spatiotemporal Action Localization[C]//European Conference on Computer Vision. Springer, Cham, 2020: 510-527.
[32] Yang X, Yang X, Liu M Y, et al. Step: Spatio-temporal progressive learning for video action detection[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 264-272.
[33] Tang J, Xia J, Mu X, et al. Asynchronous interaction aggregation for action detection[C]//European Conference on Computer Vision. Springer, Cham, 2020: 71-87.
[34] Ji J, Krishna R, Fei-Fei L, et al. Action genome: Actions as compositions of spatio-temporal scene graphs[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 10236-10247.
[35] Ehsanpour M, Abedin A, Saleh F, et al. Joint learning of social groups, individuals action and sub-group activities in videos[J]. arXiv preprint arXiv:2007.02632, 2020.
[36] Ramaswamy A, Seemakurthy K, Gubbi J, et al. Spatio-temporal action detection and localization using a hierarchical LSTM[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops. 2020: 764-765.
[37] Song Y, Kim I. Spatio-temporal action detection in untrimmed videos by using multimodal features and region proposals[J]. Sensors, 2019, 19(5): 1085.
[38] Vaswani A, Shazeer N, Parmar N, et al. Attention is all you need[J]. arXiv preprint arXiv:1706.03762, 2017.
[39] Pramono R R A, Chen Y T, Fang W H. Hierarchical self-attention network for action localization in videos[C]//Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019: 61-70.
[40] Pramono R R A, Chen Y T, Fang W H. Spatial-Temporal Action Localization with Hierarchical Self-Attention[J]. IEEE Transactions on Multimedia, 2021.
[41] Girdhar R, Carreira J, Doersch C, et al. Video action transformer network[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019: 244-253.
[42] Wang C Y, Liao H Y M, Wu Y H, et al. CSPNet: A new backbone that can enhance learning capability of CNN[C]//Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops. 2020: 390-391.
[43] Zhou P, Ni B, Geng C, et al. Scale-transferrable object detection[C]// proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 528-537.
[44] Lin T Y, Goyal P, Girshick R, et al. Focal loss for dense object detection[C]// Proceedings of the IEEE international conference on computer vision. 2017: 2980-2988.
[45] Zheng Z, Wang P, Liu W, et al. Distance-IoU loss: Faster and better learning for bounding box regression[C]//Proceedings of the AAAI Conference on Artificial Intelligence. 2020, 34(07): 12993-13000.
[46] Hurtik P, Molek V, Hula J, et al. Poly-YOLO: higher speed, more precise detection and instance segmentation for YOLOv3[J]. arXiv preprint arXiv:2005.13243, 2020.
[47] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C]//Proceedings of the IEEE conference on computer vision and pattern recognition. 2018: 7132-7141.
[48] Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]// Proceedings of the European conference on computer vision (ECCV). 2018: 3-19.
[49] Ghiasi G, Lin T Y, Le Q V. Dropblock: A regularization method for convolutional networks[J]. arXiv preprint arXiv:1810.12890, 2018.

Yue Zhuo, currently studying mechanical and electronic engineering at Southwest Jiaotong University, is mainly involved in research related to computer vision and robot human-computer interaction.

Xingguo Song, Ph.D., graduated from Harbin Institute of Technology, School of Mechanical and Electrical Engineering, majoring in Mechanical Design and Theory, is a visiting scholar at Rice University and a postdoctoral fellow at Johns Hopkins University, USA. His main research interests are intelligent robotics, UAV path planning, bionic robotics, and computer vision.

Zhongqing Cao, Ph.D., graduated from Southwest Jiaotong University with a Master's degree in Computational Mechanics and a Ph.D. in Solid Mechanics. Mainly engaged in research related to mechanics and robotics.

Sheng Jin, graduated from Southwest Jiaotong University majoring in mechanical and electronic engineering, mainly researching computer vision related topics