Driver Action Recognition Based on Dynamic Adaptive Transformer

Junqi Li, Tao Peng, Junjie Huang, Junping Liu, Xinrong Hu, Zili Zhang and Yu Mao

August 25, 2023

Driver action recognition based on Dynamic Adaptive Transformer

Junqi Li,[1][0000-0002-7833-4970] ✉ Tao Peng,[1,2,3] Junjie Huang,[1]✉ Junping Liu,[1,2] Xinrong Hu,[1,2,3] Zili Zhang [1,2] and Yu Mao[4] [0009-0000-9093-6264]

[1]School of Computer Science and Artificial Intelligence, Wuhan Textile University, Wuhan 43 0200

[2]Hbei Provincial Engineering Research Center for Intelligent Textile and Fashion, Wuhan 430200

[3]Engineering Research Center of Hubei Province of Clothing Information, Wuhan 430200

[4]School of life science, Hubei University, Wuhan 430200

Abstract. In industrial-grade applications, the efficiency of algorithms and models takes precedence, ensuring a certain level of performance while aligning with the specific requirements of the application and the capabilities of the underlying equipment. In recent years, the Vision Transformer has been introduced as a powerful approach to significantly improve recognition accuracy in various tasks. However, it faces challenges concerning portability, as well as high computational and input requirements. To tackle these issues, a dynamic adaptive transformer (DAT) has been proposed. This innovative method involves dynamic parameter pruning, enabling the trained Vision Transformer to adapt effectively to different tasks. Experimental results demonstrate that the dynamic adaptive transformer (DAT) is capable of reducing the model's parameters and Gmac with minimal accuracy loss.

Keywords: spatiotemporal attention, computer vision, driver action recognition, dynamic adaptive network, deep learning.

1 Introduction

Action recognition pertains to the utilization of computer vision and machine learning techniques to identify and comprehend human or object actions within video sequenc es. It is primarily applied in the domain of video analysis, with the goal of automatical ly detecting and classifying various actions or behaviors from video data. Action reco gnition has wide-ranging applications in various fields, including Video surveillance, Human-computer interaction, Health and medical applications, and Sports analysis. A mong the areas of machine learning is action recognition. Its purpose and significance are to determine the types of actions that the entities in the video do over time. To not ify other staff, the driver must make different gestures based on the functioning of the train during driving. Whenever an action recognition algorithm is employed in a drive system, drivers may learn normal actions and detect abnormal behaviors. Action recognition's potential applications in video surveillance, media analysis, and machine vision are also gaining traction. In recent years, action recognition technology has seen significant development in the areas of human-computer interaction, possible routes, human action analysis, and abnormal behavior detection [1-4].

Traditional and deep learning methods are the two kinds of action recognition methods. Manual feature extraction, coding, and classification are used in traditional methods. Traditional methods extract interest points, trajectories, and improved dense trajectories. A point of interest is an area with the largest increase in a particular value during video playback. Trajectories and improved dense trajectories are concepts proposed by Wang et al[1]. It is a method in use in combination with an action boundary histogram. Traditional methods, on the other hand, have poor applicability and robustness. As a result, this method is time-consuming and has problematic applications in practical problems. In recent years, deep learning has emerged as one of the most essential methods for solving problems in computer vision and other fields. Scholars developed and enhanced DNN following the first deep neural network (DNN)Alexnet[5] was successfully used in the field of image classification. After that, many 2D convolutional neural network (CNN) and 3D CNN[6-9]models were proposed and successfully applied in the field of action recognition, with excellent results.

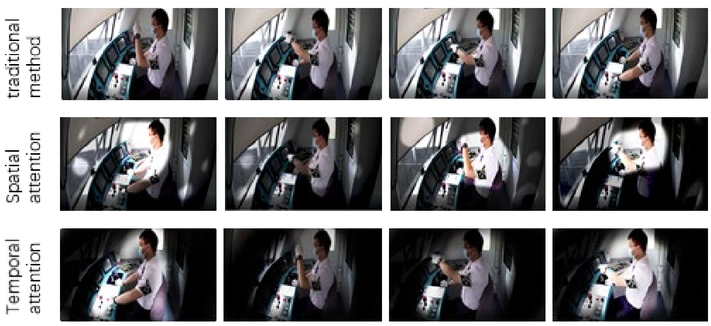


Fig. 1. Traditional methods often focus on the overall image(top row). Our method first uses a spatial attention module(second row) to process spatial information and then uses a temporal attention module (bottom row)to obtain more abundant features.

Currently, the mainstream models for action recognition include 2D CNN, 3D CNN, and transformer encoder, with the transformer encoder model gaining the most popularity. Our method is based on the spatiotemporal attention module (Fig.1.). However, the general transformer encoder's frame sequence calculation is redundant, increasing the calculation difficulty and training time. This study advances the DAT model and improves the spatiotemporal attention model. It's a Transformer encoder layout. It improves by about 4% points as compared to Timesformer and other traditional methods.

2 Related Work

IDT[10-11] and other early traditional methods are samples. The disadvantage of this method is that it has poor timeliness in processing large datasets and is challenging to apply to applications that have significant real-time requirements. The 2D CNN proposed by Karpath[12] and others did not completely deal with action time domain information. To compensate for this flaw, Simonyan et al.'s[13] dual flow structure is a popular expansion and upgrade.

Conv3d and attention[14]invented an attention technique to better consider temporal information, however, it is difficult to compute in parallel. Some [15-17] methods based on the vit [18] model and transformer (multi-head self-attention mechanism, MSA) have been proposed in recent years as a result of the successful use of attention mechanisms in computer vision. The calculation of these methods, however, is redundant. Some patch fragments have little effect on prediction results in the actual computation process, but they increase the number of calculations considerably.

In an industrial-scale task, the paper by Hou et al[19]. used a BP neural network model to classify and recognize basketball movements as an application to the project. Tieet al[20]. applied HOF and FLM and their improved algorithms to a head movement recognition system. Although these models are not novel approaches in academics, they provide some practical ideas for engineers in industrial-scale projects. In this paper, we propose and improve the better explanatory and higher recognition accuracy, Vision Transformer-based driver action recognition model, and successfully apply it in in dustrial-level projects, which provides certain solutions for subsequent academics and engineers.

Inspired by DynamicViT[21] and AdaptFormer model [22], this paper proposes a DAT driver action recognition method. The driver action model is improved in this method, and the predictor is used to reduce computational complexity and Gmac to ensure maximum accuracy. Finally, it is successfully applied to the action recognition task. Experiments show that the DAT model outperforms C3D and other methods in both the public and our datasets.

3 Dynamic Adaptive Transformer

3.1 Overview

Fig. 2. depicts the overall framework of our Dynamic Adaptive Transformer. DAT is made up of various modules, including the ones mentioned below. After layering these modules, the whole becomes a Transformer encoder.

3.2 Predictor

The predictor is the first component, and it may predict and evaluate the input patch sequence, producing a set of pathways with the highest probability for the next attention computation. As shown in Fig. 3. , the Predictor module processes the input video patch sequence to produce a lightweight output.

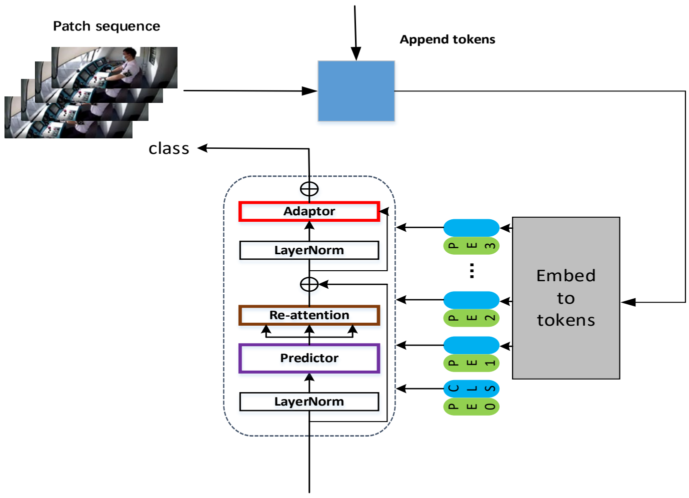


Fig. 2. The overall framework of DAT.

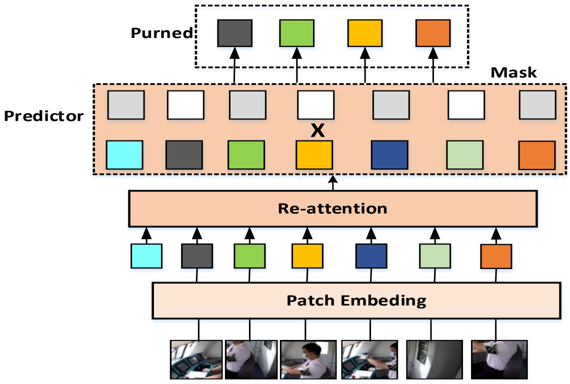


Fig. 3. The working principle and flow chart of the predictor.

It can dynamically determine which token is to be pruned. A binary mask is generated for each input model to determine which token is to be discarded. $\hat{D}$ is the probability mapping to 0 and 1 using the Softmax function, where 0 means no output and 1 means output. This module can be added to multiple layers. N represents the number of patches.

Map $\hat{D}$ and token x as inputs to MLP to obtain local feature z^{local}.

$$z^{local} = \text{MLP}(x) \tag{1}$$

Then obtain the global feature z^{global} with the same formula.

$$z^{global} = \text{aggregate}(\text{MLP}(x), D) \tag{2}$$

The aggregation formula is given by equation (3), where $u \in \mathbb{R}^{NC'}, C' = C/2$ denotes the dimensionality of the input.

$$\text{aggregate}(\hat{D}, u) = \left(\frac{\sum_{i=1}^{N} \hat{D}_i u_i}{\sum_{i=1}^{N} \hat{D}_i}\right) \tag{3}$$

Then the local and global features are spliced, and finally, they are input into MLP to predict which token will be retained or discarded.

$$z_i = [z_i^{local}, z_i^{global}], 1 \le u \le N \tag{4}$$

$$z^{'} = \text{Softmax}(\text{MLP}(z)) \tag{5}$$

3.3 Re-attention

Through layering and residual connection, the re-attention module is composed consis ting of a linear projection layer and a Transformer encoder, and each attention layer c onducts attention calculation in the adjacent patch. The module calculates the feature and outputs it after MLP. The specific formula is given by Formula (6) and Formula (7). Where $l = 1, 2....L$ is the number of layers of attention modules, $a = 1, 2.....A$ i s the number of heads of attention, and D_h is the dimension of heads of attention. p represents the number of patches in N frame images, t represents the current patch f rom which F frame images, SM represents the SoftMax function

When the model reaches a specific depth, the accuracy rate is enhanced again by re-attention calculation, which adds no additional overhead compared to self-attention calculation.

$$\alpha_{(p,t)}^{(l,a)Spatial} = \text{SM}(\frac{q_{(p,t)}^{(l,a)T}}{\sqrt{D_h}}.[\text{k}_{(0,0)}^{(l,a)}\{\text{k}_{(p^{'},t)}^{(l,a)}\}_{p^{'}=1....N}]) \tag{6}$$

$$\alpha_{(p,t)}^{(l,a)Temporal} = \text{SM}(\frac{q_{(p,t)}^{(l,a)T}}{\sqrt{D_h}}.[\text{k}_{(0,0)}^{(l,a)}\{\text{k}_{(p,t')}^{(l,a)}\}_{t'=1....F}]) \tag{7}$$

3.4 Adaptor

Although ViT has had considerable success in the field of computer vision, extending it to video is still difficult. Because of its vast amount of computing and storage, we w ill be far from reaching our existing hardware conditions if we directly fine-tune it an d migrate it to our subway driver action recognition task. To address this problem, a li ghtweight plug-and-play module is provided, which only adds 5% parameters to the model but increases the original model's accuracy by roughly 2%.

The adaptor is comprised of three components: MLP, an activation function, and t wo trainable modules. MLP and parallel trainable modules aggregate features so that s mall-scale parameters can be fine-tuned and transferred to the subway driver's action r ecognition task. Fine-tuned and transferred to the action recognition task of a subway driver. Fig. 4. depicts the Adapter's structure. Formulas (6), (7), and (8) are used to do the specific calculation (8).

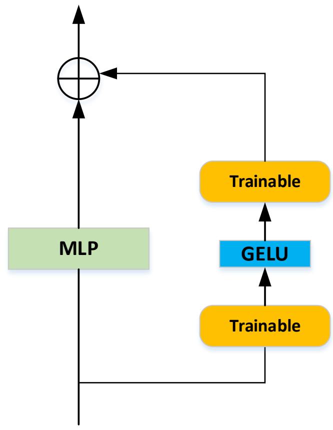


Fig. 4. Adaptor structure diagram.

First of all, like the traditional Transformer, the attention of token x_l is calculated first, and then the residual connection is performed.

$$x_l = \text{Re-attention}(Q,K,V)\text{=SoftMax}(\frac{QK^T}{\sqrt{d}})V \tag{8}$$

$$x_l^{'} = \text{MLP}(\text{LN}(x_l)) + x_l \tag{9}$$

Secondly, in the trainable modules, we have the feature $x_l^{''}$ formally via:

$$x_l^{''} = \text{GELU}(\text{LN}(x_t) \cdot W_{Trainable}) \cdot W_{Trainable^{'}} \tag{10}$$

Finally, both features $x_l^{''}$ and x_l are fused with $x_l^{'}$ by residual connection.

$$x_l^{'} = MLP(LN(x_l)) + x_l^{''} + x_l \tag{11}$$

4 Experiments

4.1 Experiments datasets

The experimental data comes from the subway cab's monitoring video. Preprocess the data by cutting the five categories of behaviors to be recognized into small segments ranging from 1 to 5 seconds, and then using the script to cut each little segment into a n 8-frame-per-second frame sequence. Since setting the batchsize too large will preve nt our device from operating, our experiment uses the Adam optimizer and sets the ba tchsize to 8.

There are about 2000 training samples, where the specific information of the datase t is given by Table 1. Car (pointing to the driving screen, It means the driver signals to drive.), Signal (pointing to the signal screen, It means that the driver signals the instr uctions), Null (no action, It means the driver doesn't make any moves), Double (Car & Signal), and Out are the five types of actions to be recognized (pointing out of the c ar, It means the driver gestures out the window). As shown in the set of pictures in Fig . 1. , there are several displays and a windshield below the driver's hand, both of whic h are objects the driver is pointing at

Table 1. Details of the dataset.

Action Category/Type	*train*	*Validation*	*test*
null	195	143	55
double	200	138	51
car	215	130	60
out	206	145	59
signal	220	148	57

4.2 Experiments Settings

Experiments are used to evaluate the efficacy and feasibility of DAT. It primarily asse sses Predictor and Adaptor's ability to improve and migrate model efficiency. Second, it simply assesses the viability of Re-attention in the deep network.

Fig. 5. first compare our method to several popular methods in Gmac and Parameters. The size of the legend indicates the value of the horizontal axis intuitively. Furthermore, more specific values are provided in Table 2. Following that, we evaluated the effectiveness of the Re-attention and Self-attention modules as the network depth

increased. The experiment (Fig. 6.) discovered that Re-attention can indeed solve the attention collapse problem of our subway driver's action dataset.

Then we have choose 3D CNN and Transformer encoder representatives for the pruning effect experiment, and the results are shown in Table 3. Except for a few met

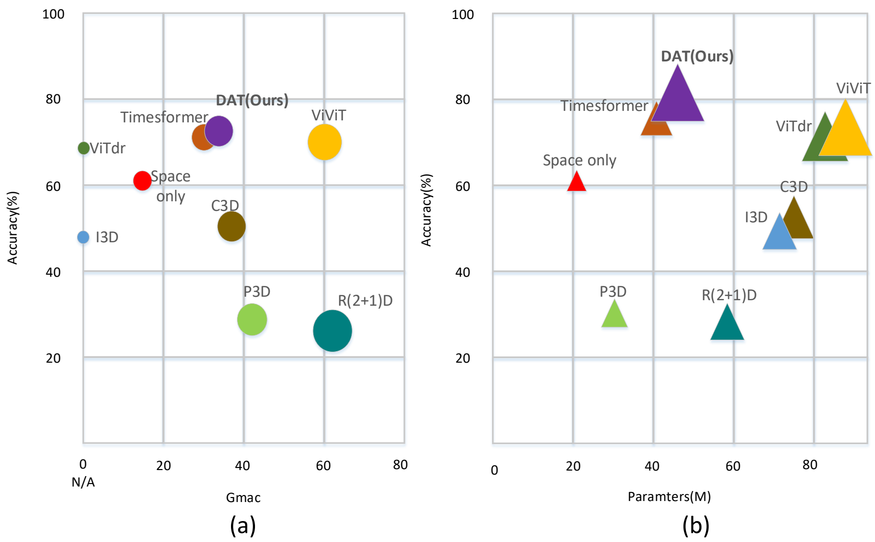


Fig. 5. Performance comparison of several methods, in our subway driver datasets: (a) shows the relationship between Gmac and Accuracy, and (b) shows the relationship between Parameters and Accuracy.

Table 2. Detailed data constituting.

Method	*GMac*	*Params*	*Accuracy*
C3D	38.67	78.02M	50.17%
P3D	40.81	33.18M	29.28%
R(2+1)D	62.68	51.99M	26.96%
Space	17.45	21.90M	60.17%
I3D	N/A	71.44M	48.55%
Timesformer	32.08	40.82M	75.51%
ViViT	40.42	88.90M	73.81%
ViTdr	N/A	81.79M	69.79%
DAT(Ours)	33.57	43.14M	78.33%

Table 3. Effect of pruning and its influence on accuracy.

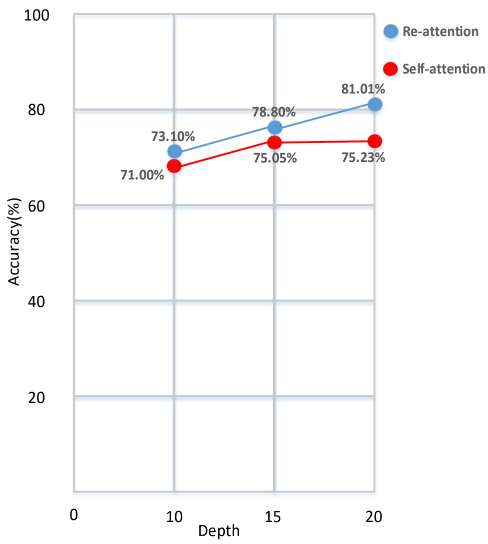
Method	*Pretrain*	*Predictor(M)*	*Non-Predictor(M)*	*Accuracy(%)*
C3D	ImageNet-1K	78.02M	85.11M	52.26%(↑ 2.09%)
P3D	ImageNet-1K	33.18M	35.05M	31.45%(↑ 2.17%)
R(2+1)D	ImageNet-1K	51.99M	55.48M	30.76%(↑ 3.8%)
I3D	ImageNet-1K	71.44M	77.86M	50.45%(↑ 1.9%)
Timesformer	ImageNet-21K	40.82M	43.35M	76.47%(↑ 0.96%)
ViViT	ImageNet-21K	88.90M	94.10M	72.29%(↓ 1.52%)
ViTdr	ImageNet-21K	81.79	85.36M	68.51%(↓ 1.28%)
DAT(Ours)	ImageNet-21K	43.14M	45.60M	77.52%(↓ 0.81%)

Table 4. Influence of Adaptor on Model Parameters and Accuracy.

Method	*Adaptor(M)*	*Non-Adaptor (M)*	*Tuning Parameter(%)*	*Accuracy(%)*
C3D	91.48M	85.11M	7.0%	50.79% (↑ 0.62%)
P3D	37.11M	35.05M	5.6%	30.71%(↑ 1.43%)
R(2+1)D	57.68M	55.48M	4.0%	29.22%(↑ 2.26%)
I3D	82.97M	77.86M	6.2%	47.84%(↓ 0.71%)
Timesformer	46.77M	43.35M	7.4%	74.94%(↓ 0.57%)
ViViT	98.23M	94.10M	4.3%	74.06%(↑ 0.25%)
ViTdr	89.57M	85.36M	4.8%	70.16%(↑ 0.37%)
DAT(Ours)	47.93M	45.60M	5.3%	79.59%(↑ 1.26%)

hods, the accuracy of the others has improved, and the number of parameters has been reduced by about 8%. Our model ensures the highest level of accuracy rate stability while also reducing model parameters. At the same time, we compared the Adaptor, which is used to migrate it to different models for experiments. Table 4 shows that the adapter only adds about 10% of the parameters to the network, but its fine-tuning par ameters are much lower than those of the Full tuning method, and our model accuracy has improved as a result.

Finally, we have chosen several public datasets for DAT comparative experiments. Fig. 7. shows that our model has some advantages and is feasible in public datasets.



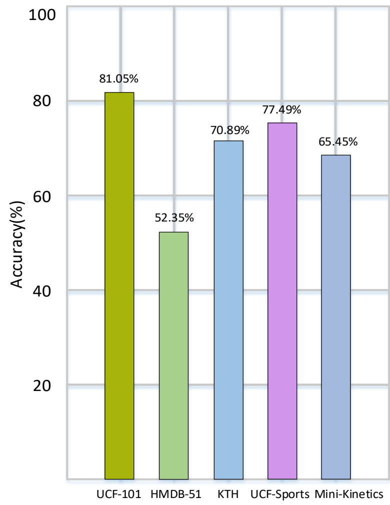


Fig. 6. Comparison between Re-attention and Self-attention.

Fig. 7. results of our model in the open datasets.

5 Conclusion

In this paper, we apply the DAT model to the task of recognizing subway driver actio ns. It can dynamically prune the model parameters. At the same time, the Adaptor mo dule increasing the portability. The experiment shows that our method achieves traditi onal methods in terms of accuracy, parameters, and other indicators, proving its feasib ility and effectiveness. We also discovered that the overfitting issue occasionally surfa ced at the start of the experiments. This issue was resolved after the datasets was recre ated with distinct action features, and we hypothesize that this may be owing to the ac tions' high repeat rate and shoddy production—however, the precise reason for this ha s to be established in further research.

References

1. Planamente, Mirco, et al. "Domain generalization through audio-visual relative norm alignment in first person action recognition." *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*. 2022.

2. Zhang, Weichen, et al. "Collaborative and adversarial network for unsupervised domain adaptation." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2018.
3. Wang, Fangxin, Jiangchuan Liu, and Wei Gong. "WiCAR: WiFi-based in-car activity recognition with multi-adversarial domain adaptation." 2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS). IEEE, 2019.
4. Olabiyi, Oluwatobi, et al. "Driver action prediction using deep (bidirectional) recurrent neural network." arXiv preprint arXiv:1706.02257 (2017).
5. Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton. "Imagenet classification with deep convolutional neural networks." Communications of the ACM 60.6 (2017): 84-90.
6. Carreira, Joao, and Andrew Zisserman. "Quo vadis, action recognition? a new model and the kinetics dataset." proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2017.
7. Tran, Du, et al. "A closer look at spatiotemporal convolutions for action recognition." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2018.
8. Peizhen, Xu, et al. "Action Recognition by Improved Dense Trajectories." Journal of System Simulation 29.9: 2053.
9. Sudhakaran, Swathikiran, Sergio Escalera, and Oswald Lanz. "Gate-shift networks for video action recognition." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020.
10. Wang, Heng, et al. "Dense trajectories and motion boundary descriptors for action recognition." International journal of computer vision 103.1 (2013): 60-79.
11. Wang, Heng, and Cordelia Schmid. "Action recognition with improved trajectories." Proceedings of the IEEE international conference on computer vision. 2013.
12. Karpathy, Andrej, et al. "Large-scale video classification with convolutional neural networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014.
13. Simonyan, Karen, and Andrew Zisserman. "Two-stream convolutional networks for action recognition in videos." Advances in neural information processing systems 27 (2014).
14. Yao, Li, et al. "Describing videos by exploiting temporal structure." Proceedings of the IEEE international conference on computer vision. 2015.
15. Li, Xiangyu, et al. "Trear: Transformer-based RGB-d egocentric action recognition." IEEE Transactions on Cognitive and Developmental Systems 14.1 (2021): 246-252.
16. Neimark, Daniel, et al. "Video transformer network." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
17. Girdhar, Rohit, et al. "Video action transformer network." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2019.
18. Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020).
19. Hou X, Ji Q. Research on the recognition algorithm of basketball technical action based on BP neural system[J]. Scientific Programming, 2022, 2022.
20. Hong T, Li Y W, Wang Z Y. Real-Time Head Action Recognition Based on HOF and ELM[J]. IEICE TRANSACTIONS on Information and Systems, 2019, 102(1): 206-209.
21. Rao, Yongming, et al. "Dynamicvit: Efficient vision transformers with dynamic token sparsification." Advances in neural information processing systems 34 (2021): 13937-13949.
22. Chen, Shoufa, et al. "AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition." arXiv preprint arXiv:2205.13535 (2022).