SPATIO-TEMPORAL ACTION DETECTION WITH A MOTION SENSE AND SEMANTIC CORRECTION FRAMEWORK

Yong Zhang[1,2], *Chunan Yu*[1], *Chenglong Fu*[1], *Yuanqi Hu*[1], *Ying Zang*[1,†]

[1] School of Information Engineering, Huzhou University, China
[2] School of Computer and Information Technology, Liaoning Normal University, China

ABSTRACT

Accurately distinguishing between action-related features and non-action-related features is crucial in spatio-temporal action detection tasks. Additionally, the calibration and fusion of information across different modalities remain challenging. This paper proposes a novel Motion Sense and Semantic Correction framework (MS-SC) to address these issues. The MS-SC framework achieves accurate detection by fusing features from images (spatial dimension) and videos (spatio-temporal dimension). A Motion Sense Module (MSM) is proposed to significantly increase the feature distance between action and non-action features in the semantic space, enhancing feature discriminability. Considering the complementary nature of information across different modalities, an efficient Semantic Correction Fusion Module (SFM) is introduced to facilitate interaction between features of distinct modalities and maximize their complementary information integration. To evaluate the performance of the MS-SC framework, extensive experiments were conducted on two challenging datasets, UCF101-24 and AVA. The results demonstrate the effectiveness of the MS-SC framework in handling spatio-temporal action detection tasks.

***Index Terms*—** Spatio-temporal action detection, feature interaction, semantic information, one-stage detection

1. INTRODUCTION

Traditional action detection tasks typically only use image features for detection, applying a single action label to the entire video. In contrast, spatio-temporal action detection utilizes temporal and spatial features to perform action labels on each image frame [1, 2]. An excellent spatio-temporal action detection framework aims to deeply understand the information present in every frame of a video and accurately assign action labels to each detected action. It should not only establish temporal connections between adjacent frames but also take into account the coherence of actions. However, previous approaches primarily relied on the 2D Convolutional Neural Network (CNN) [3, 4, 5] framework to extract spatial features from images and store them in a buffer to create correlations between frames, thereby generating the final spatio-temporal features. However, this method has certain shortcomings as it fails to fully capture the continuous nature of actions, resulting in suboptimal detection performance.

In recent years, the introduction of 3D CNN [6, 7, 8, 9, 10] has emerged as a promising approach for extracting spatio-temporal features from video clips, leading to improved accuracy in action detection. However, these fusion methods usually simply stitch together spatial features and spatio-temporal features [11], which suffer from two main shortcomings. First, the extraction of 3D spatio-temporal features cannot accurately determine the presence of an action within a specific period in a video clip. Secondly, when extracting 3D spatio-temporal features, there is a lack of feature focus. This means that the similarity between action features and non-action features is often very strong. To sum up, if the features used are not adequately spatio-temporal and if semantic information correction and full fusion among different modal features are not performed, a significant amount of redundant information may be mixed, which limits the optimal performance of the model.

Fig. 1: Jump-labeled video segments. The person's jump is the action feature, while the rest is background features, that is, non-action features are background features.

Fig. 1 illustrates one of the motivations behind our research, where recognizing actions may seem straightforward to humans but poses a challenge for model-based feature extraction due to the high similarity between extracted action and background features. To address this issue, we propose a novel Motion Sense and Semantic Correction framework (MS-SC) framework. Our primary contributions are as follows: (1) Introducing the MS-SC framework that combines spatial and spatio-temporal features for action detection. (2) Deploying a Motion Sense Module (MSM) that preprocesses spatio-temporal features to maximize the differentiation between action and background features. (3) Introducing an efficient Semantic Correction Fusion Module (SFM) that serves as a feature correction and fusion module. SFM leverages the spatial features from keyframes and the spatio-temporal features to mutually correct each other, allowing for a thorough assessment of the presence of actions within a specific time frame. Additionally, it learns valuable action representations within different modalities and extensively combines the corrected features to retain the most informative features from each modality. (4) Our framework excelled in action detection on two challenging datasets, UCF101-24 and AVA. Notably, on the UCF101-24 dataset, this framework achieved state-of-the-art results with frame mAP of 88.28 and video mAP of 53.74. On the AVA dataset, it achieved frame mAP of 22.3.

† Corresponding author: Y. Zang. Email: 02750@zjhu.edu.cn. This work was supported by the National Natural Science Foundation of China under Grant 61772252, and the Huzhou Science and Technology Plan Project under Grants 2022GZ01, 2022GZ08 and 2023ZD2004.

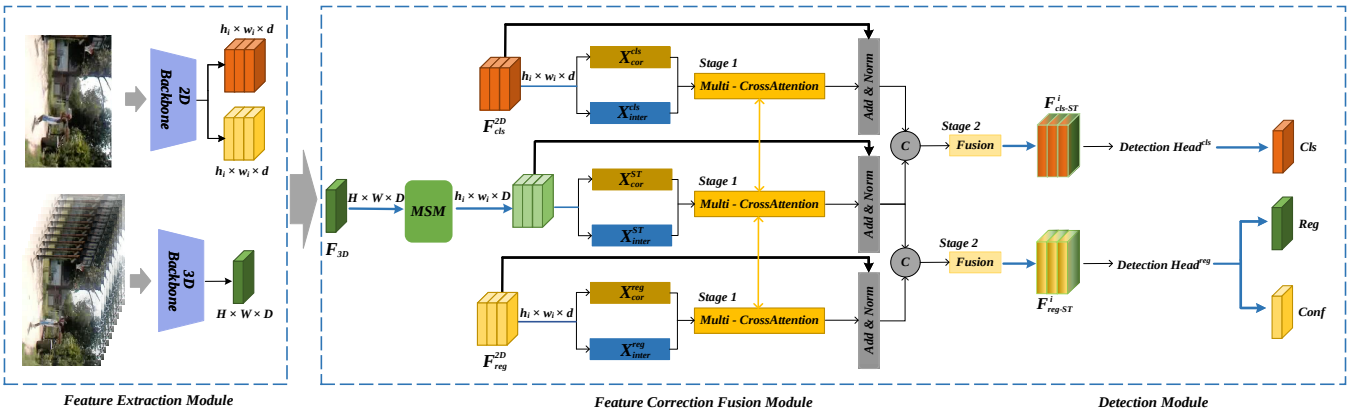


Fig. 2: The MS-SC framework includes three parts. (1) **Feature Extraction Module**: This module utilizes Backbone to extract features from both the current frame and the video segment. (2) **Feature Correction Fusion Module**: The spatio-temporal features pass through the Motion Sense Module first, effectively distinguishing action and background features. Subsequently, classification and regression features are independently fused and calibrated with spatio-temporal features. (3) **Detection Module**: This module consists of two branches for classification detection and regression detection. An anchor-free action detector is used for action classification and confidence prediction.

2. METHODOLOGY

Our framework comprises three parts: feature extraction, feature correction fusion, and detection, as shown in Fig. 2. Given a video segment V_{in}, divided into K frames, where F_k represents the current frame, we extract spatial and spatio-temporal features using 2D and 3D Backbones. Then, we mutually correct and fuse these features and perform action detection using the detection module.

2.1. Feature Extraction Module

We utilize YOLOv7's Backbone and Feature Pyramid Network (FPN) [12] as a 2D Backbone to extract multi-level spatial features $F_{2D}^i \in R^{h_i \times w_i \times d'}$ from the current frame image. We then employ two 3×3 convolutional layers as decoupling heads to produce three-level decoupling features: Classification Feature $F_{cls}^{2D} = [F_{cls}^i]_{i=1}^3$, $F_{cls}^i \in R^{h_i \times w_i \times d}$ and Regression Feature $F_{reg}^{2D} = [F_{reg}^i]_{i=1}^3$, $F_{reg}^i \in R^{h_i \times w_i \times d}$, where h_i, w_i represent image dimensions, and d', d represent latent dimensions.

Next, to ensure real-time detection and reduce computation, we adopt an efficient 3D CNN as the framework for the 3D Backbone. This allows us to extract the spatio-temporal feature $F_{3D} \in R^{H \times W \times D}$ from the video clip, and perform spatio-temporal correlation during the detection process, where H, W represent image dimensions, and D represent latent dimensions.

2.2. Feature Correction Fusion Module

In this section, we thoroughly calibrate and fuse the extracted spatial features with the spatio-temporal features. Before correction fusion, we preprocess the spatio-temporal features. Since the 3D Backbone struggles to effectively differentiate action features from background features during feature extraction, we introduced the MSM (section 2.2.1) to maximize the distinction of action features, consequently eliminating irrelevant features. Additionally, we established the SFM (section 2.2.2) which consists of two branches. These branches independently calibrate and comprehensively fuse the classification and regression features with the spatio-temporal features, enhancing the effectiveness of the features.

2.2.1. Motion Sense Module

This module comprises two main branches: the perceptual branch and the space-time branch, as shown in Fig. 3. In the perceptual branch, our objective is to enhance the distinguishability between action features and background features by increasing their feature distance from the average features at the video level. In the space-time branch, the objective is to incorporate semantic content from a wider receptive field, which helps in dynamically focusing on features of a specific scale. The formulation of this process is represented by Eq (1), (2), and (3).

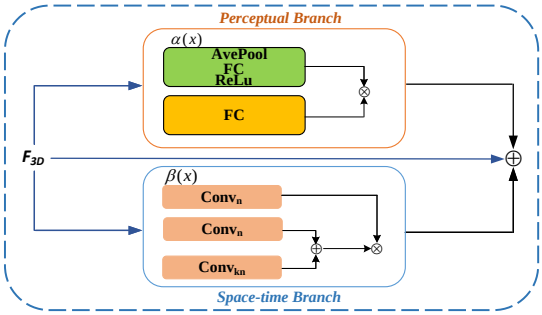


Fig. 3: The motion sense module.

$$F_{PER} = \alpha(F_{3D}) \times FC(F_{3D}) \quad (1)$$

$$F_{ST} = \beta(F_{3D})(Conv_n(F_{3D}) + Conv_{kn}(F_{3D})) \quad (2)$$

$$F_{MSM} = F_{PER} + F_{ST} + F_{3D} \quad (3)$$

where $\alpha(x)$ consists of average pooling (Avepool), fully connected layers (FC), and the ReLU function. $\beta(x)$ is composed of a one-dimensional deep convolutional layer $Conv_n$ with a window size of n. k is a window expansion factor that enlarges the receptive field of features, enabling the capture of larger-scale temporal information.

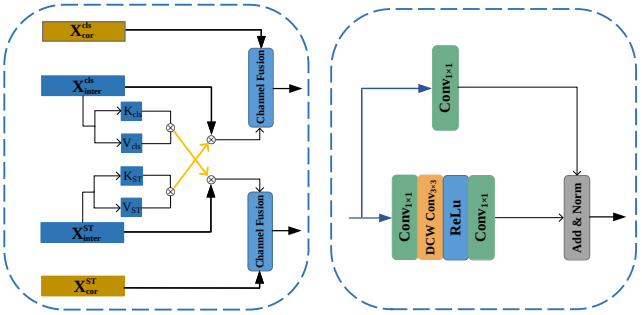
After processing the spatio-temporal features through the MSM, the distinguishability between action and background features is significantly enhanced. To facilitate fusion with the 2D decoupled features, we utilize an upsampling operation on F_{MSM} using upsampling, yielding $[F_{3D}^i]_{i=1}^3 \in R^{h_i \times w_i \times D}$, as described in Eq. (4).

$$F_{3D}^{i} = Up_{2^{i-1}}(F_{MSM}) \tag{4}$$

where the Up refers to the Upsampling operation, which can align spatio-temporal features with spatial features.

2.2.2. *Semantic Correction Fusion Module*

As spatial and spatio-temporal features represent different dimensions of action features, how to mutually correct and fuse them is critical in action detection. To address this, our SFM is designed with two stages to enhance information interaction, correction, and fusion, as shown in Fig. 4.



(a) Stage 1 (b) Stage 2

Fig. 4: The semantic correction fusion module.

Stage 1: Information exchange stage. In this stage, spatial and spatio-temporal features undergo information correction and interaction through a symmetrical dual-path structure. For simplicity, we focus on the interaction between classification features and spatio-temporal features. First, we flatten the input classification feature F_{cls}^{i} and spatio-temporal feature F_{3D}^{i} to $R^{N\times d}$ and $R^{N\times D}$, respectively, where $N = h_i \times w_i$. Due to the symmetrical structure, we will only elaborate on the classification features here. Subsequently, we employ linear embedding on the classification feature to generate two equally sized vectors: the correction vector X_{cor} and the interaction vector X_{inter}.

Due to the limitations of the previous cross-modal attention, which lacked sufficient information interaction between the two modalities and resulted in high memory usage, we flexibly adjusted and restructured the multi-head cross-modal attention based on this foundation. Specifically, we embed the interaction vectors of each head into K and V, both sized $R^{N\times C_{head}}$. For instance, we embed X_{inter}^{cls} into K_{cls} and V_{cls}, and employ X_{inter}^{ST} as Q. We then perform element-wise multiplication between K_{cls} and V_{cls}, followed by a softmax operation. Finally, we perform another element-wise multiplication with X_{inter}^{ST}, as shown in Eq. (5).

$$CA_{ST} = X_{inter}^{ST} Softmax(K_{cls}^{T} V_{cls}) \tag{5}$$

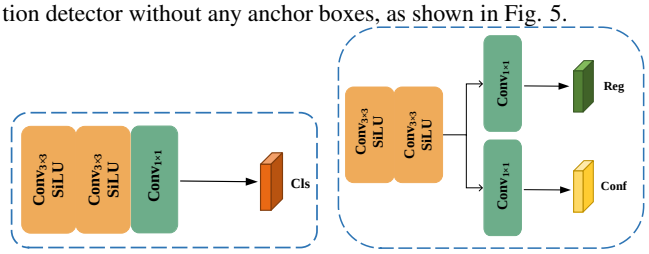
where CA denotes the outcome of participation. To achieve attention from different representation subspaces, we retain a multi-head mechanism. Then, we concatenate the participating result vector CA and the correction vector X_{cor}. Finally, we apply two linear embeddings and adjust the features to $R^{h_i\times w_i\times d}$ to output F_{CA}^{cls} and F_{CA}^{ST}.

Stage 2: Fusion stage. We use a simple channel embedding to merge the features of the two paths, which is achieved by a 1×1 convolutional layer. In addition, we believe that incorporating information from surrounding regions is crucial for robust CLS-ST segmentation. To address this, we introduce an additional deep convolutional layer $DWConv_{3\times3}$ [13] to implement the skip connection structure. In this way, merged features of size $R^{h_i\times w_i\times d\times 2}$ are fused into the final output, and feature decoding of size $R^{h_i\times w_i\times d}$ is performed, as shown in Eq. (6).

$$F_{cls-ST}^{i} = Fusion(F_{CA}^{cls} F_{CA}^{ST}) \tag{6}$$

2.3. Detection Module

After the feature correction fusion, we deploy a detection head on each of the two branches for final detection, namely classification detection and regression detection. It is worth noting that the dynamic label assignment mechanism we employ is an anchor-free action detector without any anchor boxes, as shown in Fig. 5.



(a) Classification Detection (b) Regression Detection

Fig. 5: The detection module.

Classification detection: The input is denoted as F_{cls-ST}^{i}. The detection head consists of two 3×3 convolutional layers followed by the SiLU function and a 1×1 convolutional layer followed by the sigmoid function. The output is $Y_{cls}^{i} \in R^{h_i\times w_i\times N_C}$, where N_C is the number of action classes.

Regression detection: The input is denoted as F_{reg-ST}^{i}. The design of the detection head is the same as that of the classification detection. However, there is an additional 1×1 transformation layer in parallel for action confidence prediction. The outputs are denoted as $Y_{reg}^{i} \in R^{h_i\times w_i\times 4}$ and $Y_{conf}^{i} \in R^{h_i\times w_i\times 1}$.

2.4. Loss

We define the loss as follows.

$$\mathcal{L}^{C} = \frac{1}{N_{pos}} \sum_{cls} \beta_{(\widehat{P}_{cls}>0)} \mathcal{L}_{cls}(\widehat{P}_{cls}, P_{cls}) \tag{7}$$

$$\mathcal{L}^{R} = \frac{\lambda}{N_{pos}} \sum_{reg} \beta_{(\widehat{P}_{cls}>0)} \mathcal{L}_{reg}(\widehat{P}_{reg}, P_{reg}) \tag{8}$$

$$\mathcal{L}^{F} = \frac{1}{N_{pos}} \sum_{conf} \mathcal{L}_{conf}(\widehat{P}_{conf}, P_{conf}) \tag{9}$$

$$\mathcal{L} = \mathcal{L}^{C} + \mathcal{L}^{R} + \mathcal{L}^{F} \tag{10}$$

where $\mathcal{L}_{cls}$, $\mathcal{L}_{conf}$ are the binary cross entropy, $\mathcal{L}_{reg}$ is the GIoU loss. P_{cls}, P_{reg}, and P_{conf} represent classification prediction, regression prediction, and confidence prediction, respectively. $\widehat{P}_{cls}$, $\widehat{P}_{reg}$, and $\widehat{P}_{conf}$ represent groundtruths. N_{pos} is the number of positive samples. $\beta_{(\widehat{P}_{cls}>0)}$ is an indicator function when $\widehat{P}_{cls} > 0$ is 1, otherwise it is 0. λ is the loss balance factor, and the default value is 5.

3. EXPERIMENTS SETUP AND RESULTS

3.1. Experimental Setup

The MS-SC framework conducts experiments on two challenging datasets, UCF101-24 and AVA[14]. The UCF101-24 is a subset of the UCF101 [15] dataset containing 24 action classes. The AVA dataset uses version 2.2 containing 80 class labels. We report the mean Average Precision (mAP) at an Intersection over Union (IoU) threshold of 0.5 of 60 classes out of 80 classes. For the experiments, the initial learning rate is set at 0.0001.

3.2. Results

Comparison between different modules in the MS-SC framework. The MS-SC framework introduces two key modules: the MSM and the SFM. To evaluate the impact of these modules on the network architecture, ablation experiments were conducted on two datasets. The results are presented in Table 1.

Table 1: Comparison between different modules on two datasets

Modules		UCF101-24		AVA
MSM	SFM	fAP(0.5)	vAP(0.5)	fAP(0.5)
		87.00	52.80	21.70
✓		87.54	53.51	21.94
	✓	88.07	53.68	22.14
✓	✓	**88.28**	**53.74**	**22.30**

As shown in Table 1, after adding only MSM, the frame-mAP (fAP) at an IoU threshold of 0.5 (fAP(0.5)) showed an improvement of 0.54 on the UCF101-24 dataset.The video-mAP (vAP)(0.5) also improved by 0.71 on the same dataset. On the AVA dataset, the fAP(0.5) increased by 0.21. Furthermore, after incorporating only SFM, the model's performance witnessed additional enhancements. On the UCF101-24 dataset, fAP(0.5) improved by 1.07, vAP(0.5) improved by 0.88, and on the AVA dataset, fAP(0.5) improved by 0.44. Finally, when both MSM and SFM were combined in the final model framework, further improvements were observed. On the UCF101-24 dataset, fAP(0.5) improved by 1.28, vAP(0.5) improved by 0.94, and on the AVA dataset, fAP(0.5) improved by 0.6.

Comparison with state-of-the-art works. We first compare our results with other methods on the challenging UCF101-24 dataset, as shown in Table 2.

Table 2: Comparison results on the UCF101-24 dataset

Method	Input	fAP(0.5)	vAP(0.5)
T-CNN[9]	F	41.4	—
ACT[3]]	V + F	67.1	51.4
TacNet[4]	V + F	72.1	52.9
HISAN[16]	V + F	73.7	49.5
I3D[14]	V + F	76.6	59.9
MOC[5]	V + F	78	53.8
AIA[17]	V	78.8	—
PCSC[18]	V + F	79.2	61
SAMOC[19]	V + F	79.3	52.5
YOWO[10]	F	80.4	48.8
TubeR[8]	V + F	83.2	58.4
ACAR[20]	V	84.3	—
HIT[21]	V	84.8	74.3
YOWOv2-L(K=32)[11]	F	87	52.8
YOWO+LFB[10]	F	87.3	53.1
STAR/B[22]	V	87.3	66.2
MS-SC(our)	F	**88.28**	**53.74**

As shown in Table 2, our model framework achieved 88.28 on fAP(0.5), ranking first in revenue, and vAP(0.5) achieved 53.74 on the frame input, also ranking first in revenue. The vAP is a map that calculates the dimension of the video, and our input is only frame, without the assistance of video input, but our frame still ranks high compared to other advanced works.

We also compared each label of the YOWO series on the UCF101-24 dataset, as shown in Fig. 6. It is stated that our model framework catches up with the YOWO series on most labels. Particularly, the model demonstrates prominent performance in recognizing labels that are considered difficult, class 5, 11, 12, 21, and so on.

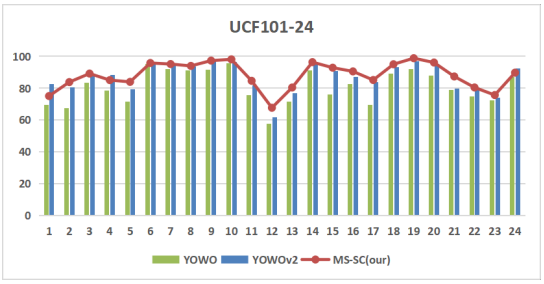
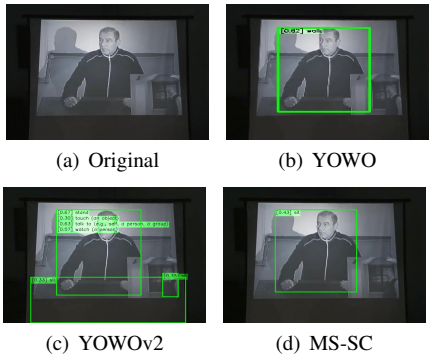


Fig. 6: Comparison of the fAP (0.5) performance for each label between the proposed method and the YOWO series on the UCF101-24 dataset.

Furthermore, we compared the performance of several methods on the AVA dataset, as shown in Table 3. Due to the highly challenging nature of the AVA benchmark, with dynamic data scenarios and multiple annotations for each action instance, our model framework prioritizes low latency and efficient real-time detection. Compared to the YOWO series, our framework achieves higher performance with minimal GFLOPs cost. Fig. 7 illustrated a visual comparison with the YOWO series on the AVA dataset. The MS-SC model demonstrates higher detection performance and produces cleaner recognition results compared to the YOWO series.

Table 3: Comparison results on the AVA dataset.

Method	Backbone	Input	fAP(0.5)	GFLOPs
I3D[14]	I3D-VGG	32 × 2	14.5	—
SlowFast[1]	SlowFast-R50	16 × 4	24.2	308
X3D-XL[6]	X3D-XL	16 × 5	26.1	290
WOO[23]	SlowFast-R101	8 × 4	28.3	252
Tuber*[8]	CSN-152	32 × 2	31.7	120
YOWO[10]	3D-ResNeXt-101	32 × 4	19.1	82
YOWOv2-L[11]	3D-ResNeXt-101	32 × 4	21.7	92
MS-SC(our)	3D-ResNeXt-101	32 × 4	**22.3**	**94.5**



(a) Original (b) YOWO

(c) YOWOv2 (d) MS-SC

Fig. 7: Visual comparison with the YOWO series on the AVA dataset.

4. CONCLUSION

In the task of spatio-temporal action detection, effectively fusing features from different modalities is crucial. In this paper, we propose a novel framework based on Motion Sense and Semantic Correction for spatio-temporal action detection. Since spatio-temporal features are not sufficient for distinguishing meaningful features, a motion sense module was proposed to differentiate action features from background features within the spatio-temporal features. Then, an efficient semantic correction fusion module was introduced, aiming to fuse spatial features and spatio-temporal features to guide each other and correct each other. Meanwhile, our framework exhibits beneficial results on both datasets.

5. REFERENCES

[1] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6202–6211.

[2] Zhaofan Qiu, Ting Yao, and Tao Mei, "Learning spatio-temporal representation with pseudo-3d residual networks," in *proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 5533–5541.

[3] Vicky Kalogeiton, Philippe Weinzaepfel, Vittorio Ferrari, and Cordelia Schmid, "Action tubelet detector for spatio-temporal action localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 4405–4413.

[4] Lin Song, Shiwei Zhang, Gang Yu, and Hongbin Sun, "Tacnet: Transition-aware context network for spatio-temporal action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11987–11995.

[5] Yixuan Li, Zixu Wang, Limin Wang, and Gangshan Wu, "Actions as moving points," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVI 16*. Springer, 2020, pp. 68–84.

[6] Christoph Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 203–213.

[7] Jianchao Wu, Zhanghui Kuang, Limin Wang, Wayne Zhang, and Gangshan Wu, "Context-aware rcnn: A baseline for action detection in videos," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXV 16*. Springer, 2020, pp. 440–456.

[8] Jiaojiao Zhao, Yanyi Zhang, Xinyu Li, Hao Chen, Bing Shuai, Mingze Xu, Chunhui Liu, Kaustav Kundu, Yuanjun Xiong, Davide Modolo, et al., "Tuber: Tubelet transformer for video action detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 13598–13607.

[9] Rui Hou, Chen Chen, and Mubarak Shah, "Tube convolutional neural network (t-cnn) for action detection in videos," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5822–5831.

[10] Okan Köpüklü, Xiangyu Wei, and Gerhard Rigoll, "You only watch once: A unified cnn architecture for real-time spatiotemporal action localization," *arXiv preprint arXiv:1911.06644*, 2019.

[11] Jianhua Yang and Kun Dai, "Yowov2: A stronger yet efficient multi-level detection framework for real-time spatio-temporal action detection," *arXiv preprint arXiv:2302.06848*, 2023.

[12] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao, "Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 7464–7475.

[13] Jiachen Li, Ali Hassani, Steven Walton, and Humphrey Shi, "Convmlp: Hierarchical convolutional mlps for vision," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6306–6315.

[14] Chunhui Gu, Chen Sun, David A Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, et al., "Ava: A video dataset of spatio-temporally localized atomic visual actions," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 6047–6056.

[15] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah, "Ucf101: A dataset of 101 human actions classes from videos in the wild," *arXiv preprint arXiv:1212.0402*, 2012.

[16] Rizard Renanda Adhi Pramono, Yie-Tarng Chen, and Wen-Hsien Fang, "Hierarchical self-attention network for action localization in videos," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 61–70.

[17] Jiajun Tang, Jin Xia, Xinzhi Mu, Bo Pang, and Cewu Lu, "Asynchronous interaction aggregation for action detection," in *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XV 16*. Springer, 2020, pp. 71–87.

[18] Rui Su, Wanli Ouyang, Luping Zhou, and Dong Xu, "Improving action localization by progressive cross-stream cooperation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12016–12025.

[19] Xurui Ma, Zhigang Luo, Xiang Zhang, Qing Liao, Xingyu Shen, and Mengzhu Wang, "Spatio-temporal action detector with self-attention," in *2021 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2021, pp. 1–8.

[20] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li, "Actor-context-actor relation network for spatio-temporal action localization," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 464–474.

[21] Gueter Josmy Faure, Min-Hung Chen, and Shang-Hong Lai, "Holistic interaction transformer network for action detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 2023, pp. 3340–3350.

[22] Alexey Gritsenko, Xuehan Xiong, Josip Djolonga, Mostafa Dehghani, Chen Sun, Mario Lučić, Cordelia Schmid, and Anurag Arnab, "End-to-end spatio-temporal action localisation with video transformers," *arXiv preprint arXiv:2304.12160*, 2023.

[23] Shoufa Chen, Peize Sun, Enze Xie, Chongjian Ge, Jiannan Wu, Lan Ma, Jiajun Shen, and Ping Luo, "Watch only once: An end-to-end video action detection framework," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 8178–8187.