Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

M^3Net: Movement Enhancement with Multi-Relation toward Multi-Scale video representation for Temporal Action Detection

Zixuan Zhao, Dongqi Wang, Xu Zhao [*]

School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China

ARTICLE INFO

Keywords:
Temporal action detection
Temporal action proposal
Video understanding
Feature representation
Multi-scale detection

ABSTRACT

Locating boundary is very important for Temporal Action Detection (TAD) and is a key factor affecting the performance of TAD. However, two factors lead to inaccurate boundary localization: the movement feature submergence and the existence of multi-scale actions. In this work, to address the submergence of movement feature, we design the Movement Enhance Module (MEM), in which the Movement Feature Extractor (MFE) and Multi-Relation Module (MRM) are used to highlight short-term and long-term movement information respectively. To address the characteristic of multi-scale actions, we propose a Scale Feature Pyramid Network (SFPN) to detect multi-scale actions and design a two-stage training strategy that makes each layer focus on a specific scale action. These tow modules are integrated as M^3Net, and extensive experiments demonstrate its effectiveness. M^3Net outperforms other representative TAD methods on ActivityNet-1.3 and THUMOS-14.

1. Introduction

Recently, considerable attention has been paid to video action understanding, due to the huge number of videos and their widespread in society. As a significant task in this field, Temporal Action Detection (TAD) aims to localize the boundary of each action segment in untrimmed videos and label it with a certain action class. The precise determination of action boundaries presents an enduring challenge within the field of TAD.

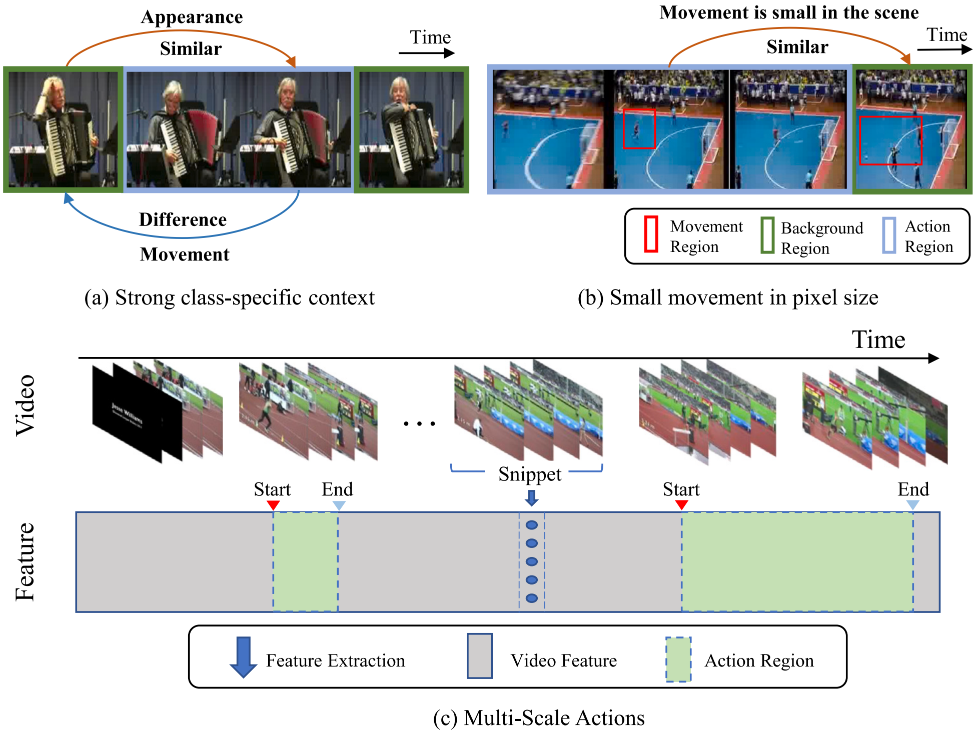
Detecting action boundaries reliably within untrimmed video is impeded by two challenges. The initial challenge, as mentioned in RefactorNet [1], action component will be affected by background and context component, denoted as movement feature submergence, which occurs in short-term and long-term temporal relations of the video. In short-term temporal relation, where either context but not movement itself dominates feature expression, or movement is small in pixel size. For the strong context case, as shown in Fig. 1(a), the presence of an *Accordion* instrument effortlessly implies the action of "Playing Accordion", while this strong class-specific context obfuscates the underlying movement information within the action region. For the small pixel case, as shown in Fig. 1(b), for *Futsal*, the vast proportion of pixels occupied by the stadium scene relegates the movement information to insignificance. This movement feature submergence induces the feature in background to be similar with the feature in action area, which obscures the action boundaries. However, RefactorNet [1] attempts to decouple action component and context component to amplify movement information in a short-term temporal, ignoring long-term temporal correlations.

In complex long-term temporal relation, temporal and semantic information can complement and highlight the submerged feature. Temporal 1D convolution is widely adopted to associate locally adjacent snippets and build the local temporal relations. Graph convolution network is applied to model relations between arbitrary snippets [2] or segments [3]. More recently, MMnet [4] and DRN [5] utilize self-attention to grasp semantic relations between distant snippets. In fact, relations in the long-term of video are complicated but central to accurate detection. As shown in Fig. 2(a), based only on limited local information cannot sufficiently enhance submerged movement features. If the model is allowed to look forward and backward and perceive the segmental temporal relations, submerged movement in consecutive similar frames can be detected much easier. Besides, semantic relations are also critical for TAD. For example, in Fig. 2(b), semantic relations help to generate more expressive action feature and complement submerged movement information. Overall, despite the above beneficial attempts at video relations, a unified framework that considers and exploits multiple relations simultaneously is still absent in TAD.

The subsequent challenge arises from the multi-scale of actions in an untrimmed video. As shown in Fig. 1(c), the richness of features exhibits significant variations across different scales. Action segments that represent a small proportion of the entire video, referred to as small-scale actions, exhibit scarce features. Conversely, segments that

[*] Corresponding author.
E-mail addresses: zhaozixuan@sjtu.edu.cn (Z. Zhao), wangdq0124@sjtu.edu.cn (D. Wang), zhaoxu@sjtu.edu.cn (X. Zhao).

https://doi.org/10.1016/j.patcog.2024.110702
Received 27 October 2023; Received in revised form 22 April 2024; Accepted 16 June 2024
Available online 27 June 2024

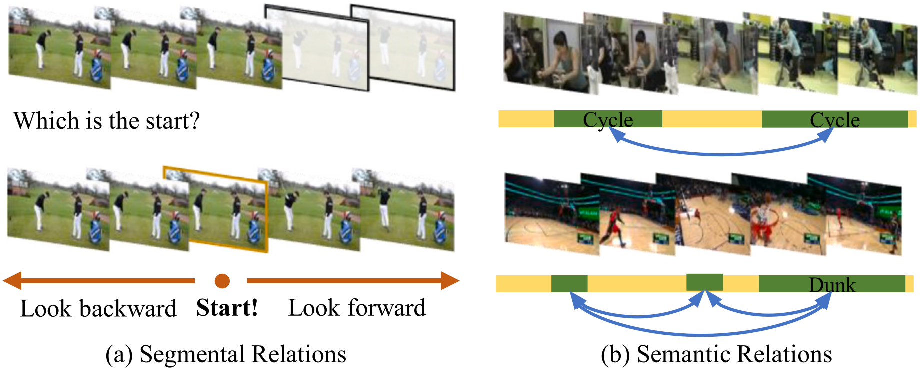


(a) Strong class-specific context

(b) Small movement in pixel size

(c) Multi-Scale Actions

Fig. 1. (a) Strong class-specific context of the *accordion*. (b) Small movement of *Futsal*. (c) Multi-scale actions have different feature richness and action pattern.



(a) Segmental Relations

(b) Semantic Relations

Fig. 2. (a) Temporal relations help the localization of boundary, and (b) Semantic relations benefit the expression of action feature.

account for a larger proportion, referred to as large-scale actions, display abundant features. As mentioned in [6], there are different action patterns between different scales. Specifically, compared with small-scale action, large-scale action contains more obviously action process (*i.e.* start phase, action phase and end phase). MD-TAPN [6] uses dilation module to handle different scale actions and AFDS [7] resorts feature pyramid network (FPN) to solve the problem. However, there are two aspects that are ignored. Firstly, these works ignore the ability of each layer to adaptively learn expression of actions at different scales. Secondly, the information flow in FPN is insufficient, lacking multi-scale receptive fields between different pyramid layers.

In this paper, as an effort to overcome the above challenges, we design the M^3Net, which features two crucial designs: (1) In order to overcome the movement feature submergence, and enlarge the difference between foreground and background snippets, we propose the Movement Enhance Module (MEM). MEM contains two important designs. Firstly, we propose the Movement Feature Extractor (MFE), which leverages the dynamic information of the frame sequence and the static information of the frame to extract movement feature in a snippet. Secondly, Multi-Relation Module (MRM) is proposed to consider and exploit multiple relations simultaneously. In MRM, several basic units are used flexibly to build three paths with different responsibilities: 1D convolution for local information aggregation, bidirectional GRU for segmental temporal relations, and the self-attention mechanism for global semantic relations. Multi-scale feature learning is only used for local path. MRM also includes segmental path and semantic path, so we call it multi-relation module. (2) In order to obtain specific representations for actions at different scales, we propose the Scale FPN (SFPN). SFPN employs a U-shape network to produce multi-scale video features and facilitate the information flow between different layers. Moreover, to ensure that each layer in SFPN focuses on action of the corresponding scale, we design a two-stage learning strategy. In the former Generalization stage, each layer is trained with all action segments; in the latter Specialization stage, each layer is biased toward actions in a specific scale range.

In summary, this work explores how to enhance submerged movement features and cope with the property of multi-scale in TAD. Its contributions are summarized as follows.

1. To alleviate the movement feature submergence,we design the MEM to highlight the movement feature in a video snippet, and

explore multi-relations between snippets. MFE is designed to extract movement feature in a short range and MRM is designed to build long range relations.

2. For the multi-scale actions in an untrimmed video with different feature patterns, we design the SFPN to learn different scale actions respectively, where targeted training and inference strategies are adopted. Consequently, each layer in the SFPN specializes in actions at a certain scale range.
3. Extensive experiments conducted on two datasets verify the effectiveness of our proposed method. On ActivityNet1.3, M^3Net promotes the best average mAP from 36.6% to 38.0%, and boosts the mAP@0.7 from 31.8% to 36.8% on THUMOS-14.

2. Related work

Temporal Action Detection. Benefiting from the successful practice of image object detection, the two-stage pipeline prevails in TAD task. This pipeline consists of a first stage for localizing candidate action segments within the video and a second stage that employs an action classifier to classify these proposals. The first stage has three main paradigms. (1) Anchor-Based method: Methods [8,9] involve placing anchors of varying scales onto the video feature and subsequently determining the final confidence of these anchors. However, they cannot produce flexible boundaries. (2) Boundary-Base method: Methods [10,11] predict boundary scores on the video feature, which are combined to generate proposals. However, the confidence of the proposal lacks global information. (3) Combined method: Methods [12,13] combine these two methods to generate precise confidence and flexible boundaries. In this work, we follow the combined method to integrate the start point and end point as a proposal and generate multi-scale anchor maps to predict anchor confidence.

Video Feature for TAD. The pre-trained TAC model is often used as the feature extractor for Temporal Action Detection which receives several frames as a snippet to distill both appearance and motion information from raw video frames. TSN [14], I3D [15] and TSM [16] are the most common feature extractors. However, they primarily focus on the action categories, leading to some actions that are strongly related to the scene having quite similar representations to background segments. To overcome the drawback, TSP [17] adds additional background supervision in the pretraining model to generate the background-sensitive feature. BSP [18] artificially synthesizes different video boundaries and conducts boundary learning in the pretrained model to generate boundary-sensitive feature. Com-STAL [19] proposes to construct the interaction between objects and actions in the video and capture the motion information. However, these works cannot fundamentally address the problem of movement feature submergence. In this paper, M^3Net uses the Movement Enhance Module (MEM) to magnify the movement feature and the difference between action and background.

Long-term temporal relations within the video are very important for accurate action detection. Many researches [5,20] have pointed out the importance of contextual information. PGCN [3] and GTAD [2] use the graph convolution to connect arbitrary frames or segments, and thus accumulate information as required. Lately, RTD-Net [21] manages to assist action detection with semantic relations inside video. Despite these above helpful tries, there is still lack of a unified framework that can explicitly establish multiple relations together. In this paper, the Multi-Relation Module (MRM) is designed to tackle it.

Multi-scale Action Detection. The scale of actions varies dramatically in a video. In comparison to larger actions, smaller actions suffer from limited samples and insufficient feature representation. MD-TAPN [6] uses multi-scale dilation module to grab different scale features and AFDS [7] focuses on feature pyramid network (FPN) to generate different temporal resolution features. However, the FPN structure fails to explicitly learn specific feature representations tailored to individual action scales and lacks inherent information flow. TSI [22] proposes a scale-invariant loss, which balances large and small actions in quantity, but fails to fundamentally solve the scarce feature of small actions. In this work, we propose SFPN to cope with these problems, which establishes associations between different scale features and learns scale-specific feature representations.

3. Method

3.1. Overview

Problem Definition. Input of the M^3Net is an untrimmed video denoted as $V = \{v_i\}_{i=1}^{L_v}$, where v_i represents the ith frame of the video and L_v is the total frames. The duration of the video is T_v. Due to redundancy between video frames, several consecutive frames are usually regarded as a snippet. With the sampling interval σ, the whole video can be defined as snippet sequence $S = \{s_i\}_{i=1}^{L_s}$, $L_s = L_v/\sigma$ representing the number of total snippets. The output of the M^3Net is $\{\psi_i|\psi_i = (t_{i,s}, t_{i,e}, c_i, score_i)\}$ where $t_{i,s}, t_{i,e}$, c_i and $score_i$ are start time, end time, action category and confidence score, respectively. The annotations of untrimmed video are action instances $\{\Psi_i|\Psi_i = (t^*_{i,s}, t^*_{i,e}, c^*_i)\}$.

Pipeline. The architecture of M^3Net is shown in Fig. 3. M^3Net is mainly composed of two parts: Movement Enhance Module (MEM) and Scale FPN (SFPN). Firstly, video V is sent to the Movement Enhance Module (MEM) to extract movement enhanced feature and explore different video relations between snippets. Next, the U-Shape module is employed to transform the feature into F_v^1, F_v^2 and F_v^3 with length L_1, L_2 and L_3, respectively. These features are combined to form a three-layer feature pyramid. Subsequently, the Detection Head at each layer generates boundary probabilities for each snippet, as well as IoU score, center offset and duration offset of each anchor. All the outputs are fused to generate action proposals. Finally, the Action Classifier tags every proposal with a certain action label.

3.2. Movement enhance module

To obtain the feature that contains action category information and is sensitive to foreground and background, simultaneously, we propose the Movement Enhance Module, which contains two vital stages: Movement Feature Extractor (MFE) and Multi-Relation Module (MRM). MFE mainly extracts movement information in a snippet. Additionally, MRM explores multi-relations between snippets in a long-term temporal relation.

Movement Feature Extractor. In order to amplify the movement information, MFE is proposed. As shown in Fig. 4, MFE uses a siamese network to highlight the movement information in a video snippet. We select R(2+1)D-34 [23] as backbone with the TAC pretrained weight on Kinetics-400, and two networks share the weight. MFE has two input paths. One path is video snippet path composed of consecutive frames, and the other is frame duplication path, which selects a frame from the snippet and copies it to the length of the snippet. The feature from the video snippet path contains both static scene and dynamic movement information, which is $F_{Total} \in \mathbb{R}^{C'\times T}$, but feature from the frame duplication path only contains static scene information, which is $F_{Static} \in \mathbb{R}^{C'\times T}$. And then, MFE extracts the movement information in the snippet using the difference between the two features, which is defined as $F_{Movement} \in \mathbb{R}^{C'\times T}$. Finally, $F_{Movement}$ and F_{Total} are concatenated as $F_{MT} \in \mathbb{R}^{2C'\times T}$. In the Table 10, we select three static frame selection methods and verify the impact of different selection methods.

Multi-Relation Module. To establish long-range temporal relations between snippets, the Multi-Relation Module (MRM) is proposed. As shown in Fig. 5, there are three well-designed paths in MRM. (1) Local Path: 1D convolution can directly establish relations in a local scale k, where k is the kernel size. In order to obtain multi-scale

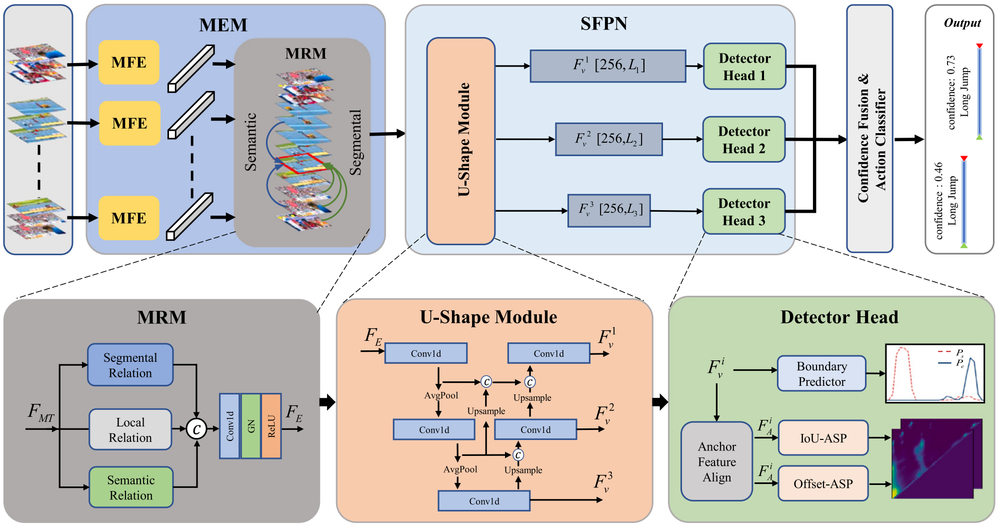


Fig. 3. Overview of our M^3Net and the structure of some sub-modules. There are two crucial parts in the framework (*Top*): Movement Enhance Module (MEM) and Scale FPN (SFPN). MEM is used to generate movement enhanced feature. SFPN is designed to generate multi-scale feature pyramid. The structure of the sub-modules is shown below (*Bottom*) with the same color in the framework. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

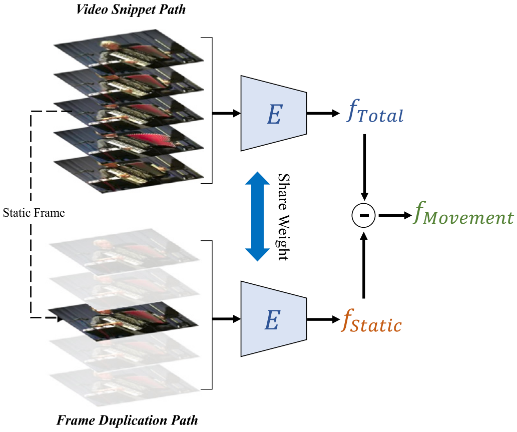


Fig. 4. Illustration of MFE. This module constructs the movement feature in a snippet by the difference between dynamic video snippet and static picture.

receptive field, inspired by spatial pyramid pooling [24], we design the Temporal Pyramid Pooling (TPP) in local path. The TPP module uses the kernel size of 3, 5 and 7 to perform average pooling on the feature sequence. (2) Segmental Path: the recurrent neural network processes each snippet successively according to the input order, with an outstanding ability to model the temporal relation. Furthermore, GRU is capable of selective memory and forgetting. In MRM, bidirectional GRU is applied to establish long-range temporal relations, which makes the receptive field of the model extend bidirectionally to the past and future. (3) Semantic Path: self-attention [25] mechanism is able to build global relations between any semantically similar segments, exchanging and aggregating information as needed. Therefore, to effectively grasp global semantic relations, four cascaded self-attention modules constitute this path. Multi-head attention is applied and the head count is set as 4.

3.3. Scale feature pyramid network

We contend different scale actions have different patterns. Therefore, the SFPN is designed to learn patterns of specific scales. SFPN contains U-Shape Module and Detector Head, which aims to generate feature pyramid and final detection results, respectively.

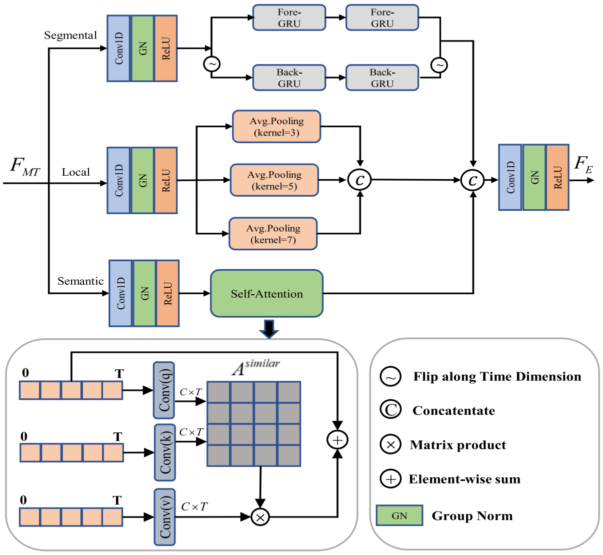


Fig. 5. Multi-Relation Module includes three paths consisting of different basic units, responsible for temporal local relations, temporal segmental relations and global semantic relations respectively.

U-Shape Module. For feature pyramid generating, as shown in Fig. 3, SFPN uses a U-shape architecture inspired by [26] to generate feature sequences in different temporal resolution. From top to down, SFPN uses multiple temporal convolution layers followed by AvgPooling to downsample the temporal resolution. From bottom to up, to restore the lower layer information, SFPN uses several 1D transpose convolution to restore the feature resolution and features in the same resolution are concatenated.

Detector Head. As demonstrated in Fig. 3, each layer in SFPN is equipped with its own detection head and each detector head consists of a Boundary Predictor, an Anchor Feature Align Layer and two Anchor Score Predictor (ASP). The output of the Boundary Predictor and Anchor Score Predictor are used to generate proposals and confidence scores, and the Anchor Feature Align Layer is to generate the feature representation for each anchor.

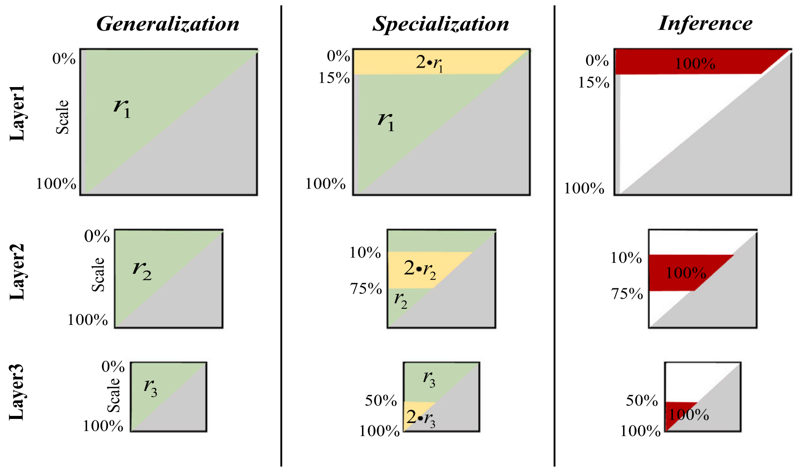


Fig. 6. The two-stage training strategy (left column and middle column) and the inference strategy (right column) of ASP. Each square represents an anchor map, the gray area is the invalid region. Other colors represent different sampling ratios in a certain scale range, with white, green, yellow and red representing 0%, r_i, $2r_i$ and 100% respectively. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Boundary Predictor. Boundary Predictor aims at predicting boundary probabilities $P_s = \{p_s^i\}_{i=1}^{L_i}, P_e = \{p_e^i\}_{i=1}^{L_i}$ of each snippet, indicating the probability that each snippet period is the start or end of action.

Anchor Feature Align Layer. Supposing N_i anchors are randomly sampled to participate in the forward. The Anchor Feature Align Layer adopts SGAlign designed in [2] to generate feature expressions $F_A^i \in \mathbb{R}^{C\times S\times N_i}$ for anchors, where S is the sampling number of snippets within an anchor.

Anchor Score Predictor. IoU-ASP and Offset-ASP have the same structure to recognize the action pattern within an anchor. For the ith layer in SFPN, IoU-ASP generates two maps $M_{cls}^i \in \mathbb{R}^{D_i\times L_i}, M_{reg}^i \in \mathbb{R}^{D_i\times L_i}$ of shape $[D_i, L_i]$, where D_i is predefined maximum anchor duration. In Offset-ASP, two output maps are denoted as $M_{cent}^i \in \mathbb{R}^{D_i\times L_i}, M_{dura}^i \in \mathbb{R}^{D_i\times L_i}$. These two maps indicate each anchor's center offset and duration offset respectively.

3.4. Specific action pattern learning

Actions of different scales have different feature representations. Correspondingly, different layers in SFPN have different feature granularity and temporal resolution. Therefore, we advocate applying a targeted training strategy for actions at different scales, assuring that each layer in SFPN is only responsible for actions at a specific scale range. Specifically, define the scale $S \in [0,1]$ as the ratio of action length to video length. Then the first layer (the top layer of the pyramid) with the longest sequence is responsible for small-scale actions whose scale in $[S_{min}^1, S_{max}^1]$, and the second layer (the middle layer of the pyramid) with a moderate sequence length is responsible for medium-scale actions whose scale in $[S_{min}^2, S_{max}^2]$. The remaining third layer with the shortest sequence is responsible for large-scale actions whose scale belongs to $[S_{min}^3, S_{max}^3]$. To implement this idea, a two-stage training strategy is designed.

Two-stage training strategy. The two-stages of training ASP are Generalization and Specialization, respectively. Due to the different lengths of feature sequences, the total number of dense anchors contained in each layer varies. In order to ensure that each layer is trained equally and reduce the computational cost, different anchor sampling ratios r_1, r_2, r_3 are set for the three layers, where $0\% < r_1 \le r_2 \le r_3 \le 100\%$. (1) In the former Generalization stage, given the sampling ratio r_i, randomly sample $N_i = \binom{L_i}{2} \cdot r_i$ anchors from valid region to train the IoU-ASP and Offset-ASP of ith layer, as shown in the left column of Fig. 6. The purpose of this stage is to let ASP learn a general pattern of all actions. (2) In the latter Specialization stage, for the ith layer in SFPN, the training samples come from two parts, as shown in the middle column in Fig. 6. The first part are randomly sampled according to the sampling rate of $min(100\%, 2\cdot r_i)$ from anchors whose scale in $[S_{min}^i, S_{max}^i]$ (yellow area in Fig. 6). The second part are sampled from the remaining anchors whose scale in $(0, S_{min}^i)$ or $(S_{max}^i, 100\%]$, with the sampling rate of r_i (green area in Fig. 6). This Specialization stage highlights the effect of anchors at the certain scale range, making the detection head of each layer more specialized.

3.5. Training & inference

Label Assignment. Assuming the video duration is L_v, each snippet corresponds to a video period. In annotation, a ground-truth action which starts at t_s^* and ends at t_e^*. Expanding boundary from moment to region, the start region is defined as $R_s = [t_s^* - 1.5\frac{T_v}{L_i}, t_s^* + 1.5\frac{T_v}{L_i}]$ and end region is defined as $R_e = [t_e^* - 1.5\frac{T_v}{L_i}, t_e^* + 1.5\frac{T_v}{L_i}]$. For Boundary Predictor, compute the overlap between each snippet period and R_s as the label of start probabilities $G_s^i \in \mathbb{R}^{L_i}$. Similarly, compute the overlap between snippet period and R_e as the label of end probabilities G_e^i. For IoU-ASP, following [12], IoU between each anchor and all actions are calculated and then arranged into a map $G_{IoU}^i \in \mathbb{R}^{D_i\times L_i}$. Anchors with IoU score greater than 0.9 participate in the training of Offset-ASP. For anchor start at t_s and end at t_e, assuming its corresponding ground-truth action is $[t_s^*, t_e^*]$, the center offset Δc and duration offset Δd are calculated as Eqs. (1)–(2), and arranged into maps $G_{cent}^i \in \mathbb{R}^{D_i\times L_i}$ and $G_{dura}^i \in \mathbb{R}^{D_i\times L_i}$.

$$c = \frac{t_s + t_e}{2}, d = t_e - t_s, c^* = \frac{t_s^* + t_e^*}{2}, d^* = t_e^* - t_s^* \tag{1}$$

$$\Delta c = \frac{c^* - c}{d}, \Delta d = \log\frac{d^*}{d} \tag{2}$$

Basic Loss. (1) For the ith layer, the loss of Boundary-Predictor is the re-weighted binary cross-entropy loss L_B Eq. (3), where $P = \{p_t\}_{t=1}^{L_i}$ and $G = \{g_t\}_{t=1}^{L_i}$ represent the predicted boundary probabilities and its label of all snippets, respectively. Snippets with $g_t > 0.5$ serve as positive samples (*i.e.*, $\delta\{g_t > 0.5\} = 1$) and others are negative samples. T_i^+ and T_i^- are the number of positive and negative samples in this layer respectively. (2) The loss of IoU-ASP is the re-weighted binary cross-entropy loss L_{asp} Eq. (4) and L2 loss L_2, where $M = \{m_j\}_{j=1}^{N_i}$ and $G = \{g_j\}_{j=1}^{N_i}$ represent the predicted value and ground-truth value of each sampled anchor in the ith layer. N_i, N_i^+, N_i^- represent the number of sampled anchors, positive anchors whose IoU greater than 0.9 and negative anchors of this layer, respectively. (3) The loss of

Offset-ASP is Smooth L1 loss L_1. Note that only those sampled anchors with IoU greater than 0.9 participate in the training of Offset-ASP. The re-weights in L_B and L_{asp} are used to balance the number between positive and negative samples.

$$\begin{aligned}L_B(P,G) = -\frac{1}{L_i}\sum_{t=1}^{L_i}(\frac{L_i}{T_i^+}\cdot\delta\{g_t > 0.5\}\cdot\log p_t \\ +\frac{L_i}{T_i^-}(1-\delta\{g_t > 0.5\}\cdot\log(1-p_t)))\end{aligned} \tag{3}$$

$$\begin{aligned}L_{asp}(M,G) = -\frac{1}{N_i}\sum_{j=1}^{N_i}(\frac{N_i}{N_i^+}\cdot\delta\{g_j > 0.9\}\cdot\log m_j \\ +\frac{N_i}{N_i^-}(1-\delta\{g_j > 0.9\}\cdot\log(1-m_j)))\end{aligned} \tag{4}$$

Total Loss. The loss of the ith layer is composed of three parts: boundary loss, IoU loss and offset loss, as shown in Eqs. (5) (6) (7) respectively. The total training objective is the sum of all three layers, formulated as Eq. (8). Besides, in order to balance the value between different terms, the coefficient $\lambda 1$ and $\lambda 2$ are set as 5 and 10.

$$L^i_{bound} = L_B(P^i_s, G^i_s) + L_B(P^i_e, G^i_e) \tag{5}$$

$$L^i_{IoU} = L_{asp}(M^i_{cls}, G^i_{IoU}) + \lambda_1\cdot L_2(M^i_{reg}, G^i_{IoU}) \tag{6}$$

$$L^i_{off} = L_1(M^i_{cent}, G^i_{cent}) + L_1(M^i_{dura}, G^i_{dura}) \tag{7}$$

$$L_{total} = \sum_{i=1}^{3}(L^i_B + L^i_{IoU} + \lambda_2\cdot L^i_{off}) \tag{8}$$

Inference. Each layer outputs P^i_s, P^i_e, M^i_{cls}, M^i_{reg}, M^i_{cent} and M^i_{dura} from the detector head in inference. Following [10,12,22], snippet in boundary probability P_s is screened out as candidate start point if it is local peak or its probability is greater than $0.5\cdot\max(P_s)$. And snippets can be selected as candidate end point from P_e in the same way. Then the candidate start and end points are combined into proposals. In order to produce the more reasonable proposals, as shown in Fig. 6, each layer of SFPN only outputs anchors in the corresponding scale range. Specifically, for the ith layer, its responsible scale range is $[S^i_{min}, S^i_{max}]$. Furthermore, in the proposals set, for any start snippet whose index is sdx and centered at time t_1, and the end snippet whose index is edx and centered at time t_2, we can get its start probability $p_s = P^i_s[sdx]$, end probability $p_e = P^i_e[edx]$ and IoU score $p_{IoU} = M^i_{cls}[edx - sdx, sdx]\cdot M^i_{reg}[edx - sdx, sdx]$. Its center is $c = (sdx + edx)/2L_i$ and duration is $d = (edx - sdx)/L_i$. Subsequently, to refine the boundary, we can obtain center offset $\Delta c = M^i_{cent}[edx - sdx, sdx]$ and duration offset $\Delta d = M^i_{dura}[edx - sdx, sdx]$, and adjust the boundary to as Eq. (9):

$$\begin{aligned}c' = d\cdot\Delta c + c, d' = d\cdot e^{\Delta d} \\ t'_1 = c' - \frac{d'}{2}, t'_2 = c' + \frac{d'}{2}\end{aligned} \tag{9}$$

Finally, proposals generated by each layer are merged together according to their confidence. We sort them from high to low and select the top K as final proposals. The action classifier assigns every proposal with a certain label and a classification score P_{label}. The final score for the proposal (t'_1, t'_2) is shown as Eq. (10). Then Soft-NMS [32] is adopted to remove redundant segments.

$$score_{t'_1,t'_2} = p_s\cdot p_e\cdot p_{IoU}\cdot p_{label} \tag{10}$$

4. Experiments

4.1. Dataset and settings

Dataset. In order to verify the effectiveness of our method, we test M^3Net on two challenging datasets. **ActivityNet-1.3** [27] contains 200 categories of daily life, sports, *etc*. We use the training set for model training and report the performance on the validation set. **THUMOS-14** [28] consists of 413 videos with 20 action classes which is almost sports. In THUMOS-14, the validation set contains 200 long videos, including 3007 action segments, and the test set contains 213 videos, including 3358 action segments. Following the standard practice of THUMOS-14, we train M^3Net on the validation set and valid it on the test set.

Table 1
The temporal action detection performance comparison with state-of-the-art methods on ActivityNet-1.3. Bold data indicates the best performance.

Method	Backbone	mAP@tIoU (%)			
		0.5	0.75	0.95	Avg.
BSN [10]	TSN	46.5	29.9	8.0	30.0
BMN [12]	TSN	50.1	34.8	8.3	33.9
G-TAD [2]	TSN	50.4	34.6	9.0	34.1
PCMNet [31]	TSN	51.4	36.1	9.5	35.3
TCA-Net [32]	TSN	52.3	36.7	6.9	35.5
RTD-Net [21]	I3D	47.2	30.7	8.6	30.8
ContextLoc [33]	I3D	56.0	35.2	3.6	34.2
AFDS [7]	I3D	52.4	35.3	6.5	34.4
TAGS [34]	I3D	56.3	36.8	9.6	36.5
GTAN [35]	P3D	52.6	34.1	8.9	34.3
STPT [36]	STPT	51.4	33.7	6.8	33.4
UnLoc-L [37]	CLIP	**58.8**	–	–	–
ActionFormer [38]	R(2 + 1)D	54.7	37.8	8.4	36.6
TriDet [39]	R(2 + 1)D	54.7	38.0	8.4	36.8
M^3Net **(ours)**	**R(2 + 1)D**	55.0	**38.8**	**10.8**	**38.0**

Metric. In order to fully demonstrate the advantages of our proposed method, we test M^3Net on two video action understanding tasks: temporal action detection (TAD) and temporal action proposal (TAP). Compared with TAD, TAP is aimed at locating action segments without labels. For TAD task, mAP under a certain temporal IoU threshold (mAP@IoU) is the main metric. As for TAP task, average recall at a specified average number of proposals (AR@AN) is used to evaluate the proposal performance. Besides, on ActivityNet, the area under the AR-AN curve (AUC) also serves as a TAP metric.

Network Details. The sampling interval σ set as 16 frames and 5 frames for ActivityNet-1.3 and THUMOS-14, respectively. For ActivityNet-1.3, we use linear interpolation to resize the length of video features $T = 200$. For a long video in THUMOS-14, a sliding window with a length of 256 and an overlap of 50% is used to truncate the original video feature. Finally, the results of all windows are concatenated as the total result of the entire long video. The scale ranges of each layer in the pyramid are set as: $[S^1_{min}, S^1_{max}] = [0, 15\%], [S^2_{min}, S^2_{max}] = [10\%, 75\%]$ and $[S^3_{min}, S^3_{max}] = [50\%, 100\%]$. As for the action classifier, following the other two-stage methods, Untrimmed Net [29] serves as the classifier for THUMOS-14, and the recognition model by [30] for ActivityNet-1.3.

Implementation Details. M^3Net is trained using the Adam optimization algorithm with batch size 8 and learning rate 10^{-3}. The sampling ratio of each layer r_1, r_2, r_3 are set to 50%, 40%, 90%, respectively. For ActivityNet-1.3, the training process takes 10 epochs, where the first 7 epochs is the Generalization stage and the rest is the Specialization stage. As for THUMOS-14, the total training epoch is 7, and only the first epoch is the Generalization stage. Besides, the learning rate is decayed to 10^{-4} after 5 epochs on THUMOS-14 and 7 epochs on ActivityNet-1.3.

4.2. Performance evaluation on detection

We compare M^3Net with other state-of-the-art methods on ActivityNet-1.3 and THUMOS-14. Table 1 shows the comparison on the validation set of ActivityNet-1.3. The mAP at IoU thresholds of $\{0.5, 0.75, 0.95\}$ are reported, as well as the Avg.mAP which is calculated at IoU thresholds between 0.5 and 0.95 with the step of

Table 2
The temporal action detection performance comparison with state-of-art methods on THUMOS-14, measured by mAP@tIoU. Bold text indicates the best results. M^3Net performs better than all other existing methods at high tIoU threshold.

Method	Backbone	mAP@tIoU (%)					
		0.3	0.4	0.5	0.6	0.7	Avg.
BSN [10]	TSN	53.5	45.0	36.9	28.4	20.0	36.8
BMN [12]	TSN	56.0	47.4	38.8	29.7	20.5	38.5
G-TAD [12]	TSN	54.5	47.6	40.2	30.8	23.4	39.3
PCMNet [31]	TSN	61.5	55.4	47.2	37.5	27.3	45.8
TCA-Net [32]	TSN	60.6	53.2	44.6	36.8	26.7	44.3
RTD-Net [21]	I3D	68.3	62.3	51.9	38.8	23.7	49.0
ContextLoc [33]	I3D	68.3	63.8	54.3	41.8	26.2	50.9
AFDS [7]	I3D	67.3	62.4	55.5	43.7	31.1	52.0
TAGS [34]	I3D	68.6	63.8	57.0	46.3	31.8	52.8
GTAN [35]	P3D	57.8	47.2	38.8	–	–	–
STPT [36]	STPT	70.6	65.7	56.4	44.6	30.5	53.6
ActionFormer [38]	R(2 + 1)D	**73.4**	**67.4**	59.1	46.7	31.5	55.6
M^3Net **(ours)**	**R(2 + 1)D**	71.9	66.3	**59.5**	**49.9**	**36.8**	**56.9**

Table 3
Temporal action proposal performance comparison with other representative two-stage TAD methods on ActivityNet-1.3 and THUMOS-14. Bold data indicate the best performance. M^3Net outperforms all other methods.

Method	ActivityNet1.3		THUMOS14			
	AR@100	AUC	AR@50	AR@100	AR@200	AR@500
BSN [10]	74.2	66.2	37.5	46.1	53.2	60.6
BMN [12]	75.0	67.1	39.4	47.7	54.7	62.1
RTD-Net [21]	73.2	65.8	41.5	49.3	56.4	62.9
TCA-Net [32]	76.1	68.1	42.1	50.5	57.1	63.6
M^3Net **(ours)**	**77.1**	**70.0**	**47.9**	**56.6**	**63.0**	**69.3**

0.05. Impressively, M^3Net outperforms other representative methods at IoU thresholds 0.75 and 0.95. M^3Net promotes the Avg.mAP from 36.6% to 38.0%, with an increase of more than 1.4%. Moreover, We list the SOTA methods with different backbones. Since UnLoc-L [37] uses a stronger large language model as the backbone, and pretrains on multiple tasks, UnLoc-L obtains the best performance. In spite of the large language model as the backbone, our proposed approach demonstrates superiority over the others. Specifically, when compared to the ActionFormer [38] that shares the same R(2+1)D backbone, our method outperforms ActionFormer [38]. Moreover, in comparison to TAGS [34], which utilizes a more powerful I3D backbone, we also surpass the performance of TAGS [34].

Table 2 presents the detection performance comparison on THUMOS-14. M^3Net achieves comparable performance with the best method [38] at low tIoU thresholds. But, M^3Net outperforms other representative methods at tIoU thresholds 0.5 and 0.7. To locate the boundaries more accurately, a higher threshold is applied to discriminate between positive and negative samples during training. Consequently, M^3Net exhibits heightened performance at high tIoU. Higher tIoU means more precise action boundaries, a more challenging yet critical aspect of action localization. Meanwhile, it is noteworthy that M^3Net shows obvious superiority compared with other FPN-based methods like [7], which substantiates the effectiveness of the SFPN proposed in M^3Net.

4.3. Performance evaluation on proposal

To further verify the advantages of M^3Net, we do not consider action classes of action segments to evaluate its performance on temporal action proposal (TAP) task. Through Table 3, we can further discover the performance improvement of AR@100 and the area under the AR-AN curve (AUC) on ActivityNet-1.3. For instance, M^3Net boosts the AUC from 68.1% to 70.0%. As for THUMOS-14, M^3Net outperforms other methods in all metrics, which certifies the effectiveness of M^3Net once again.

Table 4
Ablation study on the MEM. The performance is reported on ActivityNet-1.3 and THUMOS-14. The best result can be achieved with these two modules are integrated.

Construction	THUMOS14		ActivityNet1.3	
	mAP@0.5	Avg.mAP	mAP@0.5	Avg.mAP
Base	54.5	53.1	54.2	37.0
Base + Movement feature	56.1	54.3	54.5	37.3
Base + MRM	57.7	55.8	54.1	37.3
MEM	**59.5**	**56.9**	**55.0**	**38.0**

Table 5
Ablation study on Multi-Relation Module. The detection performance is reported on ActivityNet-1.3 and THUMOS-14.

Local	Semantic	Segmental	THUMOS14		ActivityNet1.3	
			mAP0.5	Avg.mAP	mAP0.5	Avg.mAP
	✓		56.2	54.3	54.0	36.8
		✓	56.6	54.7	54.6	37.5
	✓	✓	57.3	55.4	54.7	37.7
✓			56.1	54.7	54.5	37.3
✓	✓		57.3	55.4	54.6	37.5
✓		✓	57.9	55.7	54.6	37.7
✓	✓	✓	**59.5**	**56.9**	**55.0**	**38.0**

4.4. Ablation study

Ablation studies are performed thoroughly to verify the role of each module in M^3Net, as well as the impact of different training strategies.

(1) ***Effectiveness of the MEM.*** As discussed before, when constructing the movement enhance feature, MFE and MRM are the crucial designs in MEM. To certify the function of these two modules, we design a Base model for comparison, in which the Movement Feature is removed and MRM is replaced by temporal 1D convolution. As shown in Table 4, we can find that compared with the Base model, Movement Feature and MRM both can promote the performance. Besides, the best results emerge when combining them together as the intact MEM.

(2). ***Effectiveness of the Multi-Relation Module.*** Multi-Relation Module (MRM) consists of three paths, aiming to explore video temporal and semantic relations within a unified framework. The local path formed by TPP, it has multi-scale receptive field. From Table 5, exploiting the temporal segmental relation and the global semantic relation are both favorable to the detection performance. In addition, complete MRM at the last row verifies that these three paths do not conflict, and combining them is the best choice.

(3). ***Effectiveness of the U-shape Module.*** As discussed before, when constructing the feature pyramid, U-shape Module is designed to establish interactions between different layers. To reveal the validity of this idea, we use 1D convolution with the step of 2 and AvgPooling to downsample the feature sequence respectively. In addition, we use

Table 6
Ablation study on the U-shape module. The Detection performance is reported on ActivityNet-1.3 and THUMOS-14.

Construction	THUMOS14		ActivityNet1.3	
	mAP@0.5	Avg.mAP	mAP@0.5	Avg.mAP
DownSampling (AvgPooling)	57.5	55.5	54.4	37.5
DownSampling (Convolution)	57.9	55.9	54.5	37.4
ASPP	55.9	54.0	54.1	37.1
PPM	56.4	54.5	54.2	37.3
U-shape module	**59.5**	**56.9**	**55.0**	**38.0**

Table 7
Ablation study on the feature pyramid structure. "length" means the temporal length of single-layer feature sequence. Bold data indicate the best performance.

THUMOS14			ActivityNet1.3		
Length	mAP@0.5	Avg.mAP	Length	mAP@0.5	Avg.mAP
64	53.7	52.6	50	53.0	36.6
128	54.3	53.4	100	54.0	37.1
256	56.3	54.7	200	54.2	37.3
SFPN	**59.5**	**56.9**	**SFPN**	**55.0**	**38.0**

Table 8
Ablation study on the two-stage training strategy. The proposed two-stage training leads to better performance than the single stage method or the Bias method.

Training strategy	THUMOS14		ActivityNet1.3	
	mAP@0.5	Avg.mAP	mAP@0.5	Avg.mAP
Only Bias	55.5	53.5	45.5	29.5
Only generalization	57.5	55.5	54.3	37.4
Only specialization	57.5	55.7	54.6	37.4
Two-stage	**59.5**	**56.9**	**55.0**	**38.0**

Atrous Spatial Pyramid Pooling (ASPP) and Pyramid Pooling Module (PPM) to build feature pyramids respectively. From Table 6, compared with using multi-scale convolution to build FPN, physically detecting actions on different scale features can bring more performance improvement. In addition, compared with only using DownSampling to construct FPN, using U-shape module can get the best performance, which substantiates the importance of this design.

(4). ***Effectiveness of feature pyramid structure.*** For better detection accuracy, M^3Net makes use of the feature pyramid. To verify the impact of this structure, we only use a single layer feature sequence to conduct experiments and compare it with the SFPN, keeping other components unchanged. On ActivityNet-1.3, single-layer feature with lengths of 200, 100 and 50 are used. On THUMOS-14, features with lengths of 256, 128 and 64 are used. Corresponding results are shown in Table 7. The experimental results exhibit that SFPN performs better than any other single-layer structure, thus justifying the feature pyramid is indeed beneficial to the performance.

(5). ***Effectiveness of two-stage training.*** For efficient training of M^3Net and ensuring that the ith layer in SFPN is specialized in actions at a specific scale range $[S^i_{min}, S^i_{max}]$, the two-stage training strategy is applied. When training ASP of the ith layer, in the first Generalization stage, the anchors are randomly selected from all scales according to the sampling ratio r_i. In the second Specialization stage, the sampling ratio inside and outside the range $[S^i_{min}, S^i_{max}]$ are $2 \cdot r_i$ and r_i, respectively. In this ablation study, we train M^3Net using the fully Generalization stage, the fully Specialization stage and the complete two-stage strategy, respectively. In addition, to demonstrate the effectiveness of this training method, we also test a new Bias training strategy that only samples from the range $[S^i_{min}, S^i_{max}]$ with the sampling ratio of r_i.

As shown in Table 8, the performance of the Bias strategy is the worst, indicating that in addition to those samples in the specific scale range, other samples outside this range also play an important role for a better understanding of actions. Further, the training effect is impeded when removing either the Generalization or the Specialization stage, which verifies the effectiveness of the two-stage strategy. We verify the impact of different sampling ratio of the highlighted scale. According to the Table 9, increasing the sampling ratio causes M^3Net to ignore the role of other samples, causing the overall performance to decline.

Table 9
Ablation study on the sampling ratio of two-stage training strategy.

Sampling ratio	THUMOS14		ActivityNet1.3	
	mAP@0.5	Avg.mAP	mAP@0.5	Avg.mAP
Two-stage (sampling ratio 6)	56.7	54.0	54.2	37.2
Two-stage (sampling ratio 4)	58.2	56.0	54.8	37.7
Two-stage (sampling ratio 2)	**59.5**	**56.9**	**55.0**	**38.0**
Two-stage (sampling ratio 1)	57.5	55.5	54.3	37.4

Table 10
Ablation study on the strategy for duplicate generation. We chose three selection methods, of which Temporally Centered strategy achieved the best performance.

Strategy	mAP@tIoU (%)			
	0.5	0.75	0.95	Avg.
Random strategy	54.5	38.2	11.0	37.5
Average strategy	54.7	38.5	11.4	37.8
Temporally Centered strategy	55.0	38.8	10.8	**38.0**

(6) ***Effectiveness of the strategy for duplicate generation.*** In the MFE, we conduct experiments to investigate the influence of different strategies for selecting static frames. Specifically, we examine three strategies: Temporally Centered, Random, and Average. The Temporally Centered strategy selects the frame located at the temporal center of each clip for duplicate generation. The Random strategy randomly selects a frame within each clip for duplicate generation. Lastly, the Average strategy involves computing the average of all frames within each clip and using the resulting frame for duplicate generation.

As shown in Table 10, the Random strategy exhibits inferior performance due to the potential inclusion of blurry frames or abrupt camera movements. Conversely, Temporally Centered strategy and Average strategy contain centered frame and average frame, respectively, which achieve better performance. Temporally Centered strategy stands out as the most effective, ultimately leading us to adopt this strategy for duplicate selection.

(7). ***Qualitative Comparison.*** To illustrate the boundaries of M^3Net more intuitively, we select four videos from ActivityNet-1.3 and THUMOS-14 and compare our results qualitatively with the Base, in which we remove the MEM and only use the single-layer feature sequence of lengths 100 and 128 for ActivityNet-1.3 and THUMOS-14 respectively. Results with the highest confidence score are visualized in Fig. 7. Firstly, Compared with the base method, M^3Net uses the movement feature, which enlarges the difference between action and background, making the model more accurate in boundary localization. Secondly, it can be seen the High Jump action contains a large-scale action and a small-scale action. Due to the multi-scale module (SFPN) in M^3Net, M^3Net can better capture small-scale action and more accurately locate small-scale action boundaries. Results of qualitative comparison once again prove the superiority of M^3Net.

5. Conclusion

Temporal action detection (TAD) aims to accurately localize action segments and classify their corresponding action classes. In an effort to overcome the movement feature submergence and multi-scale action detection, we propose the M^3Net with two crucial designs. Firstly, we propose the Movement Enhance Module (MEM) to highlight movement feature at both short-term and long-term temporal relations. In short-term temporal, the Movement Feature Extractor (MFE) is used to enhance movement information by exploiting the difference between dynamic clip and static image. In long-term temporal, the

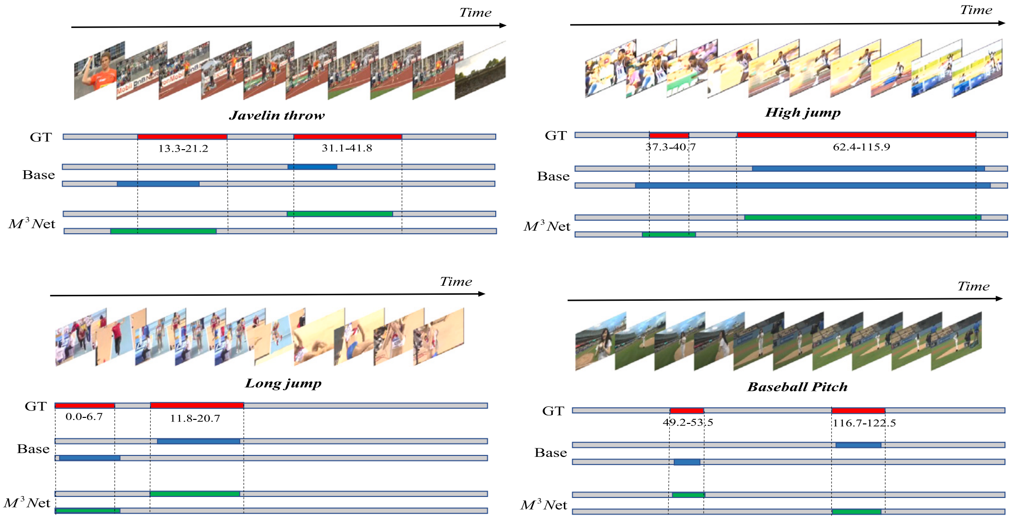


Fig. 7. Qualitative Comparison on ActivityNet-1.3 (first row) and THUMOS-14 (second row). The predictions of M^3Net can cover the ground truth actions with higher overlap. The numbers of the GT bar indicate the start and end times, respectively. M^3Net can better capture small-scale action and more accurately locate small-scale action boundaries.

Multi-Relation Module (MRM) is used to enlarge the difference between action and background by capturing multi-temporal relations. Secondly, we propose the Scale FPN (SFPN) to handle the different scale actions. In order to learn scale information in a targeted manner, we design the two-stage training strategy, ensuring that each layer in M^3Net is specialized at a specific scale range. Extensive experiments conducted on the ActivityNet-1.3 and THUMOS-14 validate the superiority of M^3Net. However, our method is still based on dense anchors, resulting in a large number of proposals and low efficiency. Besides, M^3Net does not take advantage of the interaction between the proposals. In future work, we can combine it with the DETR-based method, and build temporal and semantic associations between proposals to detect the actions more accurate and efficient.

CRediT authorship contribution statement

Zixuan Zhao: Writing – original draft, Visualization, Validation, Methodology, Data curation, Conceptualization. **Dongqi Wang:** Writing – original draft, Methodology, Investigation, Conceptualization. **Xu Zhao:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgments

This work has been funded in part by the National Natural Science Foundation of China (NSFC) (No.62176156) and Medical Engineering Cross Research Fund of Shanghai Jiao Tong University (YG2023ZD12).

References

[1] K. Xia, L. Wang, S. Zhou, N. Zheng, W. Tang, Learning to refactor action and co-occurrence features for temporal action localization, in: CVPR, 2022, pp. 13884–13893.

[2] M. Xu, C. Zhao, D.S. Rojas, A. Thabet, B. Ghanem, G-tad: Sub-graph localization for temporal action detection, in: CVPR, 2020, pp. 10156–10165.

[3] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, C. Gan, Graph convolutional networks for temporal action localization, in: ICCV, 2019, pp. 7094–7103.

[4] M. Korban, P. Youngs, S.T. Acton, A multi-modal transformer network for action detection, Pattern Recognit. 142 (2023) 109713.

[5] K. Xia, L. Wang, S. Zhou, G. Hua, W. Tang, Dual relation network for temporal action localization, Pattern Recognit. 129 (2022) 108725.

[6] P. Li, J. Cao, L. Yuan, Q. Ye, X. Xu, Truncated attention-aware proposal networks with multi-scale dilation for temporal action detection, Pattern Recognit. 142 (2023) 109684.

[7] C. Lin, C. Xu, D. Luo, Y. Wang, Y. Tai, C. Wang, J. Li, F. Huang, Y. Fu, Learning salient boundary feature for anchor-free temporal action localization, in: CVPR, 2021, pp. 3320–3329.

[8] X. Dai, B. Singh, G. Zhang, L.S. Davis, Y. Qiu Chen, Temporal context network for activity localization in videos, in: ICCV, 2017, pp. 5793–5802.

[9] T. Lin, X. Zhao, Z. Shou, Single shot temporal action detection, in: ACMMM, 2017, pp. 988–996.

[10] T. Lin, X. Zhao, H. Su, C. Wang, M. Yang, Bsn: Boundary sensitive network for temporal action proposal generation, in: ECCV, 2018, pp. 3–19.

[11] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, D. Lin, Temporal action detection with structured segment networks, in: ICCV, 2017, pp. 2914–2923.

[12] T. Lin, X. Liu, X. Li, E. Ding, S. Wen, Bmn: Boundary-matching network for temporal action proposal generation, in: ICCV, 2019, pp. 3889–3898.

[13] Z. Zhao, D. Wang, X. Zhao, Movement enhancement toward multi-scale video feature representation for temporal action detection, in: ICCV, 2023, pp. 13555–13564.

[14] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal segment networks for action recognition in videos, IEEE Trans. Pattern Anal. Mach. Intell. 41 (11) (2018) 2740–2755.

[15] J. Carreira, A. Zisserman, Quo vadis, action recognition? a new model and the kinetics dataset, in: CVPR, 2017, pp. 6299–6308.

[16] X. Song, C. Lan, W. Zeng, J. Xing, X. Sun, J. Yang, Temporal–spatial mapping for action recognition, IEEE Trans. Circuits Syst. Video Technol. 30 (3) (2019) 748–759.

[17] H. Alwassel, S. Giancola, B. Ghanem, Tsp: Temporally-sensitive pretraining of video encoders for localization tasks, in: ICCV, 2021, pp. 3173–3183.

[18] M. Xu, J.-M. Pérez-Rúa, V. Escorcia, B. Martinez, X. Zhu, L. Zhang, B. Ghanem, T. Xiang, Boundary-sensitive pre-training for temporal localization in videos, in: ICCV, 2021, pp. 7220–7230.

[19] S. Wang, R. Yan, P. Huang, G. Dai, Y. Song, X. Shu, Com-STAL: Compositional spatio-temporal action localization, IEEE Trans. Circuits Syst. Video Technol. (2023).

[20] Z. Zhao, D. Wang, X. Zhao, BACNet: Boundary-anchor complementary network for temporal action detection, in: ICME, IEEE, 2022, pp. 01–06.

[21] J. Tan, J. Tang, L. Wang, G. Wu, Relaxed transformer decoders for direct action proposal generation, in: ICCV, 2021, pp. 13526–13535.

[22] S. Liu, X. Zhao, H. Su, Z. Hu, Tsi: Temporal scale invariant network for action proposal generation, in: ACCV, 2020.
[23] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, M. Paluri, A closer look at spatiotemporal convolutions for action recognition, in: CVPR, 2018, pp. 6450–6459.
[24] K. He, X. Zhang, S. Ren, J. Sun, Spatial pyramid pooling in deep convolutional networks for visual recognition, IEEE Trans. Pattern Anal. Mach. Intell. 37 (9) (2015) 1904–1916.
[25] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A.N. Gomez, L. Kaiser, I. Polosukhin, Attention Is All You Need, Vol. 30, NIPS, 2017.
[26] O. Ronneberger, P. Fischer, T. Brox, U-net: Convolutional networks for biomedical image segmentation, in: MICCAI, Springer, 2015, pp. 234–241.
[27] F.C. Heilbron, V. Escorcia, B. Ghanem, J.C. Niebles, Activitynet: A large-scale video benchmark for human activity understanding, in: CVPR, IEEE, 2015, pp. 961–970.
[28] Y.-G. Jiang, J. Liu, A.R. Zamir, G. Toderici, I. Laptev, M. Shah, R. Sukthankar, THUMOS challenge: Action recognition with a large number of classes, 2014.
[29] L. Wang, Y. Xiong, D. Lin, L. Van Gool, Untrimmednets for weakly supervised action recognition and detection, in: CVPR, 2017, pp. 4325–4334.
[30] Y. Zhao, B. Zhang, Z. Wu, S. Yang, L. Zhou, S. Yan, L. Wang, Y. Xiong, D. Lin, Y. Qiao, et al., Cuhk & ethz & siat submission to activitynet challenge 2017, 8 (8) (2017). arXiv preprint arXiv:1710.08011.
[31] X. Qin, H. Zhao, G. Lin, H. Zeng, S. Xu, X. Li, PcmNet: Position-sensitive context modeling network for temporal action localization, Neurocomputing 510 (2022) 48–58.
[32] Z. Qing, H. Su, W. Gan, D. Wang, W. Wu, X. Wang, Y. Qiao, J. Yan, C. Gao, N. Sang, Temporal context aggregation network for temporal action proposal refinement, in: CVPR, 2021, pp. 485–494.
[33] Z. Zhu, W. Tang, L. Wang, N. Zheng, G. Hua, Enriching local and global contexts for temporal action localization, in: ICCV, 2021, pp. 13516–13525.
[34] S. Nag, X. Zhu, Y.-Z. Song, T. Xiang, Proposal-free temporal action detection via global segmentation mask learning, in: ECCV, Springer, 2022, pp. 645–662.
[35] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, T. Mei, Gaussian temporal awareness networks for action localization, in: CVPR, 2019, pp. 344–353.
[36] Y. Weng, Z. Pan, M. Han, X. Chang, B. Zhuang, An efficient spatio-temporal pyramid transformer for action detection, in: ECCV, Springer, 2022, pp. 358–375.
[37] S. Yan, X. Xiong, A. Nagrani, A. Arnab, Z. Wang, W. Ge, D. Ross, C. Schmid, Unloc: A unified framework for video localization tasks, in: ICCV, 2023, pp. 13623–13633.
[38] C.-L. Zhang, J. Wu, Y. Li, Actionformer: Localizing moments of actions with transformers, in: ECCV, Springer, 2022, pp. 492–510.
[39] D. Shi, Y. Zhong, Q. Cao, L. Ma, J. Li, D. Tao, Tridet: Temporal action detection with relative boundary modeling, in: CVPR, 2023, pp. 18857–18866.

Zixuan Zhao received the BEng and MS. degree in Automation from University of Electronic Science and Technology of China, Chengdu, China. He is currently working toward the Ph.D. degree at the Department of Automation, Shanghai Jiao Tong University. His research interests include video understanding and temporal action detection.

Dongqi Wang received the BEng degree in Automation from Jilin University, Jilin, China. He received MS. degree in the Department of Automation, Shanghai Jiao Tong University. He is now working at Hikvision. His research interests include video understanding and temporal action detection.

Xu Zhao is currently a full professor at the Department of Automation in School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University (SJTU). He got his Ph.D. degree from SJTU in Pattern Recognition and Intelligent System in 2011. He was a visiting scholar in Beckman institute at UIUC from Nov. 2007 to Dec. 2008, and Postdoc research fellow at Northeastern University from Aug. 2012 to Aug. 2013. His research interests include visual analysis of human motion, machine learning and image/video processing.