IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received August 9, 2021, accepted September 1, 2021, date of publication September 7, 2021, date of current version September 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3110973

ABN: Agent-Aware Boundary Networks for Temporal Action Proposal Generation

KHOA VO[1], (Member, IEEE), KASHU YAMAZAKI[1], SANG TRUONG[1], MINH-TRIET TRAN[2,4], (Member, IEEE), AKIHIRO SUGIMOTO[3], (Member, IEEE), AND NGAN LE[1], (Member, IEEE)

[1]AICV Lab, University of Arkansas, Fayetteville, AR 72703, USA
[2]University of Science, VNU-HCM, Ho Chi Minh City 700000, Vietnam
[3]National Institute of Informatics (NII), Tokyo 101-8430, Japan
[4]Vietnam National University, Ho Chi Minh City 700000, Vietnam

Corresponding author: Khoa Vo (khoavoho@uark.edu)

This work was supported by the National Science Foundation under Award OIA-1946391; funded by Gia Lam Urban Development and Investment Company Limited, Vingroup, and partially supported by Vingroup Innovation Foundation (VINIF) under project code VINIF.2019.DA19.

ABSTRACT Temporal action proposal generation (TAPG) aims to estimate temporal intervals of actions in untrimmed videos, which is a challenging yet plays an important role in many tasks of video analysis and understanding. Despite the great achievement in TAPG, most existing works ignore the human perception of interaction between agents and the surrounding environment by applying a deep learning model as a black-box to the untrimmed videos to extract video visual representation. Therefore, it is beneficial and potentially improves the performance of TAPG if we can capture these interactions between agents and the environment. In this paper, we propose a novel framework named Agent-Aware Boundary Network (ABN), which consists of two sub-networks: (1) an Agent-Aware Representation Network to obtain both agent-agent and agents-environment relationships in the video representation; and (2) a Boundary Generation Network to estimate the confidence score of temporal intervals. In the Agent-Aware Representation Network, the interactions between agents are expressed through local pathway, which operates at a local level to focus on the motions of agents whereas the overall perception of the surroundings are expressed through global pathway, which operates at a global level to perceive the effects of agents-environment. Comprehensive evaluations on 20-action THUMOS-14 and 200-action ActivityNet-1.3 datasets with different backbone networks (i.e C3D, SlowFast and Two-Stream) show that our proposed ABN robustly outperforms state-of-the-art methods regardless of the employed backbone network on TAPG. We further examine the proposal quality by leveraging proposals generated by our method onto temporal action detection (TAD) frameworks and evaluate their detection performances.

INDEX TERMS Temporal action proposal generation, temporal action detection, agent-aware boundary network.

I. INTRODUCTION

Temporal action proposal generation (TAPG) [1]–[14] is one of the most key and fundamental tasks in video understanding i.e. action recognition [15], [16], video summarization [17], [18], video captioning [19], [20], video recommendation [21], video highlight detection [22], and smart surveillance [23], [24]. Given an untrimmed video, TAPG aims to propose temporal intervals with specific starting and ending timestamps for each action. Most of existing TAPG approaches first detect a set of possible starting and ending timestamps of all actions separately, and then a proposal evaluation module is employed to evaluate every possible pair of starting and ending timestamps by predicting its confidence score. The non-maximum suppression (NMS) function is finally used to eliminate redundant candidate proposals based on their confidence scores and overlapping metrics. A robust TAPG method should be able to (i) generate temporal proposals with actual boundaries to cover action instances precisely and exhaustively; (ii) cover multi-duration actions; (iii) generate reliable confidence scores so that proposals can be retrieved properly [6]. Despite recent endeavors

The associate editor coordinating the review of this manuscript and approving it for publication was Sotirios Goudos.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

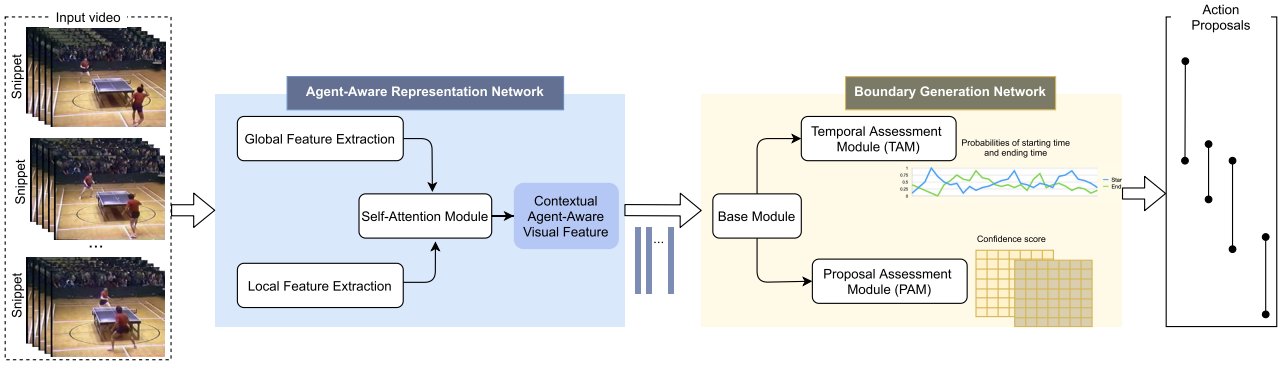


FIGURE 1. An overview architecture for our proposed Contextual Agent-Aware Boundary Network (ABN) for TAPG. ABN consists of two main sub-networks i.e. *Agent-Aware Representation Network* and *Boundary Generation Network*.

[6], [12], [13], TAPG remains as an open problem, especially when facing real-world problems such as action duration variability, activity complexity, camera motion, viewpoint changes, etc. In spite of good achievements on benchmarking datasets, the existing TAPG approaches have some limitations as follows:

- In previous works, video visual representation is extracted by directly applying a backbone model, e.g. C3D/I3D [25], Two-Stream network [15], [26] or SlowFast network [16] into the whole spatial dimensions of the video (or entire snippet). This makes the predictions biased to the background (or environment) instead of agents who commit actions because the agents with their actions usually occupy a small region compared to the entire frame.
- A temporal action proposal is a combination of three entities i.e. agent, action, and environment; however, the existing approaches do not have any mechanism to present such combination as well as express the relationship between these entities.

To address the aforementioned limitations, we leverage human perception process of a temporal action proposal which is a combination of three entities i.e. agent, action, and environment and we propose a novel **Contextual Agent-Aware Boundary Network (ABN)**. Our ABN consists of two main sub-networks i.e. *Agent-Aware Representation Network* and *Boundary Generation Network*. The first sub-network aims to extract video visual representation i.e. contextual agent-aware visual feature, given an untrimmed video whereas the second sub-network aims to estimate the confidence score matrices and the probabilities of starting time and ending time given a video feature. To interpret those entities of agent, action, and environment, our Agent-Aware Representation Network comprises of two semantic pathways corresponding to *local pathway*, which locally extracts information from the agents who commit actions and *global pathway*, which globally extracts information from entire environment. Furthermore, the number of agents in a given video can be arbitrary; however, a few of them are actually committing the action. To extract a semantic local feature, we apply a self-attention module. The final video feature combines both local feature and global feature through a self-attention module. The second sub-network, Boundary Generation Network, takes contextual agent-aware visual feature as an input and consists of three modules corresponding to Base Module to model the temporal relationship as well as provide a shared feature sequence for later modules of Temporal Assessment Module (TAM) and Proposal Assessment Module(PAM). The overall flowchart of our proposed ABN is given in Fig.1.

Our main contributions are summarized as follows:

1) Leveraging the human perception process of understanding an action which combines agents, action and environment, we propose an end-to-end contextual ABN for TAPG. Our ABN contains two sub-networks corresponding to (i) Agent-Aware Representation Network for extracting semantic video feature given untrimmed video and (ii) Boundary Generation Network to evaluate confidence scores of densely distributed proposals.
2) Introducing Agent-Aware Representation Network, a novel video contextual visual representation, for extracting video feature from an untrimmed video. Our semantic Agent-Aware Representation Network involves two parallel pathways: (i) local pathway to tell what agents are doing (ii) global pathway to express the relationship between the agents and the environment.
3) Investigating the impacts of agents and the environment as well as the interaction between agents and their environment in our proposed ABN framework.
4) Examining the action proposal quality and effectiveness of our proposed ABN by putting proposals that it generated to TAD framework and evaluate its detection performance.
5) Benchmarking the proposed ABN on popular datasets in both TAPG and TAD, namely ActivityNet-1.3 with three different backbone networks (i.e. C3D, SlowFast and Two-Stream) and THUMOS-14 with two backbone networks (i.e. C3D and Two-Stream). Our proposed ABN has achieved state-of-the-art performance

on both TAPG and TAD regardless of backbone network. The source code can be found in this URL.[1]

II. RELATED WORK

A. TEMPORAL ACTION PROPOSAL GENERATION (TAPG)

TAPG aims to propose temporal intervals that may contain an action instance with their temporal boundaries and confidence in untrimmed videos. In general, TAPG can be categorized into three groups i.e. anchor-based and boundary-based and hybrid anchor-boundary-based as follows:

- The anchor-based TAPG methods [1], [2], [5], [7]–[10], [27] refer to the temporal boundary refinements of predefined anchors or sliding windows. Those methods are inspired by the achievements of anchor-based object detectors in still images like Faster R-CNN [28], RetinaNet [29], or YOLOv3 [30]. These methods discretize the proposal task into a classification task where multiple predefined anchors with different lengths are used as classes and a class that best fits the ground truth action length is regarded as ground truth true class for training. In such approaches, a large number of proposals are densely generated. Although this approach helps to save computational costs, this approach lacks the flexibility of action duration.
- The boundary-based TAPG methods [11]–[13] resolve the above problem by breaking every action interval into starting and ending points and learn to predict them. In those methods, there are two stages corresponding to generating the boundary probability sequence and applying the Boundary Matching mechanism to generate candidate proposals. In inference time, starting and ending probabilities at every time in the given video are predicted, then, those with local peaks will be chosen as potential boundaries. The potential starting points are paired with potential ending points to become a potential action interval when their interval fits in the predefined upper and lower threshold, along with a confidence score being a multiplication of the starting and ending probabilities. As one of the first boundary-based methods, [11] defined actionness scores by grouping continuous high-score regions as the proposal. Later, boundary-sensitive method [12] proposed a two-stage strategy where boundaries and actionness scores at every temporal point are predicted in the first stage and fused together, filtered by Soft-NMS to get the final proposals at the second stage.
- In order to make use of the advantages of both anchor-based methods and boundary-based methods, [6], [14] proposed hybrid approaches in which the boundary detection and the dense confidence regression are performed simultaneously by using ROI align. Based on the observation that anchor-based methods can uniformly cover all segments in videos but imprecise while boundary-based methods may have more precise boundaries but it may omit some proposals when the quality of actionness score is low, [2] proposed Complementary Temporal Action Proposal (CTAP) generator. BMN [6] is an improvement of BSN [12]. In BMN, a boundary-matching matrix is generated instead of actionness scores to capture an action-duration score for more descriptive final scores, which help to improve the final proposals' prediction. Continuously, drawing the inspiration from BSN [12], [14] proposed Dense Boundary Generator (DBG) and implemented the boundary classification and action completeness regression for densely distributed proposals.

The TAPG approaches can be summarized in Fig. 2.

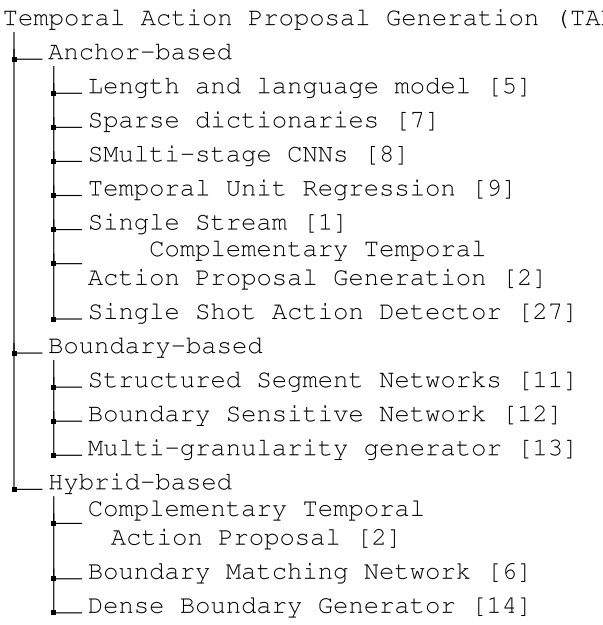


FIGURE 2. Approaches summarization on TAPG.

B. TEMPORAL ACTION DETECTION (TAD)

Depending on spatial or temporal domain, action detection approaches can be categorised into either TAD (TAD) or spatial action detection (SAD) or spatial-temporal action detection. TAD aims to find the temporal intervals of starting action and ending action whereas SAD searches for human region and the corresponding human action in spatial domain. In this work, we focus on TAD which provides the answer of what and when the action happens in a video.

Due to action recognition is a part of TAD, thus, most of the early TAD methods were built based on hand-crafted features, the same as action recognition. Early TAD methods are based on efficient spatio-temporal feature representations and motion propagation across frames in videos such as HOG3D [31], SIFT3D [32], ESURF [33], MBH [34] etc. As the performance of the methods using hand-crafted features became stabilized, TAD has reached a leveling off. With the convolutional neural networks (CNNs) was developed [35], a lot of effective TAD approaches have proposed. In general, TAD can be divided into either one-stage detection or two-stage detection.

[1] https://github.com/vhvkhoa/TAPG-AgentEnvNetwork.git

In one-stage framework, both temporal proposal and action classification are learnt simultaneously. Due to the similarity between TAD and object detection example, SSAD [27], SS-TAD [36] made use of single-shot detector to solve TAD with one-stage detection. While both SSAD and SS-TAD make use of C3D feature [25], [37]–[39], SS-TAD adopts the anchor mechanism and the stacked GRU units.

Unlike one-stage framework, two-stage approach is based on the paradigm of proposal generation-and-then classification i.e. extracts temporal proposals first, and then processes with the classification and regression operations. Similar to proposal generation in object detection, TAPG plays the most important role in TAD in this two-stage approach paradigm. Two-stage framework is the mainstream method, so most papers adopt this. TAD can be implemented by: (i) sliding windows such as S-CNN [8] which fixes some size sliding windows to generate various sizes video segments, and then deal with them by a multi-stage network. S-CNN is build on C3D feature and contains thee sub-networks corresponding to TAPG, classification and localization; However, S-CNN is time consumption when increasing the overlap between the windows to obtains good performance, TURN [9] leverages Faster RCNN to improve S-CNN by integrating boundary regression network. (ii) boundary network such as BSN [12], BMN [6], DBG [14] which aim to deal with video actions of different lengths and with precise temporal boundaries as well as reliable confidence scores. BSN first locates the boundaries of the temporal action segments i.e. starting time and ending time. Both starting time and ending time are then combined into temporal proposal. Based on the sequence of action confidence scores for each candidate proposal, a 32-dimensional proposal-level feature is extracted and benchmarked for evaluating the confidence of the temporal proposals. BMN and DBG are both improvements of BSN with a new confidence evaluation and boundary-matching mechanisms.

C. VIDEO FEATURE REPRESENTATION

Following the success of CNNs on image tasks. In [25], Tran *et al.* proposed a simple linear model named C3D which outperforms all previous best-reported methods. By transferring the 2D pre-trained model to 3D model, [40] proposed I3D. In I3D, the 3D filters are replaced by a set of repeated 2D filters. Inspired by the success of ResNet in image classification, Hara, *et al.* extended ResNet architecture to 3D CNN and proposed 3D ResNet [41]. In their work, they examined various 3D CNN architecture under different backbone such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, ResNet-200, DenseNet-121 and ResNeXt-101. The mainstream networks fall into three categories: Two-Stream networks, Recurrent Neural Network (RNN) with its popular variant named Long Short Term Memory (LSTM), and 3D networks. Two-Stream networks were first introduced by [15] and then they have been improved in [26]. Two-Stream networks explore video appearance and motion clues with two separate networks. One network is to exploit spatial information from individual frames while the other uses temporal information from optical flow. The outputs of the two networks are then combined by late fusion. RNN/LSTM is believed to cope with sequential information better, and thus many proposed methods [42], [43] attempted to incorporate LSTM to deal with action recognition. 3D networks, which were first introduced by [44], extract features from both the spatial and the temporal dimensions by performing 3D convolutions, thereby capturing the motion information encoded in multiple adjacent frames. Later on, C3D features, 3D CNN architectures and their improvements [25], [37]–[41] have been proposed. In [25], Tran *et al.* proposed a simple linear model named C3D which outperforms all previous best-reported methods. By transferring the 2D pre-trained model to 3D model, [40] proposed I3D. In I3D, the 3D filters are replaced by a set of repeated 2D filters. Inspired by the success of ResNet in image classification, Hara, *et al.* extended ResNet architecture to 3D CNN and proposed 3D ResNet [41]. In their work, they examined various 3D CNN architecture under different backbone such as ResNet-18, ResNet-34, ResNet-50, ResNet-101, ResNet-152, ResNet-200, DenseNet-121 and ResNeXt-101. Recently, SlowFast network [16] is a variation of 3D CNN networks category, in which two parallel pathways are utilized to capture appearances of video scene and object motion in each pathway. SlowFast networks have been proposed to tackle the action recognition and action spatial localization tasks and got the highest scores in many benchmark datasets e.g. Kinetics, Charades, AVA, etc.

Our proposed ABN belongs to the category of two-stage TAD and boundary-based TAPG approach where our focus is human perception-based video feature extraction which aims to obtain semantic video representation followed by human perception of action understanding.

III. OUR PROPOSED METHOD: AGENT-AWARE BOUNDARY NETWORK (ABN)

A. PROBLEM FORMULATION

Considering an untrimmed video $\mathcal{V} = \{x_l\}_{l=1}^{L}$ with L frames, our goal is to generate a set of temporal segments which inclusively and tightly contain actions of interest. Given a set of ground truth action segments $\mathcal{A} = \{a_i = (s_i, e_i)\}_{i=1}^{M}$ having M temporal segments with an action segment comprised of a starting timestamp (s_i) and an ending timestamp (e_i), our objective is formalized by minimizing the following objective function:

$$\mathcal{L} = \sum_{i=1}^{M} \log p(a_i|\mathcal{V}) \tag{1}$$

As proposed by [12] and [6], the above objective function may also be achieved indirectly by decomposing the action proposal generation problem into detecting starting and ending timestamps of every actions together with their duration,

as formulated below:

$$\mathcal{L} = \sum_{i=1}^{M} \log p(s_i|\mathcal{V}) + \log p(e_i|\mathcal{V}) + \log p(e_i - s_i|s_i) \quad (2)$$

Instead of representing the video frames separately as individual feature vectors, in our proposed framework, we divide the video $\mathcal{V}$ into $T = \lfloor \frac{L}{\delta} \rfloor$ non-overlapping snippets of δ consecutive frames. Each snippet of δ frames captures the motions taking place alongside the appearances, which are crucial in detecting the starting and ending timestamps for each action.

We denote ϕ as a feature extraction function which is applied to every δ-frame snippet, the visual representation feature sequence F of the entire video $\mathcal{V}$ can be defined as:

$$F = \{f_i\}_{i=1}^{T} = \{\phi(x_{\delta\cdot(i-1)+1}, \ldots, x_{\delta\cdot i})\}_{i=1}^{T} \quad (3)$$

In most of the previous works [2]–[4], [6], [12]–[14], [45], [46], function ϕ is simply defined as the extraction of a feature vector from a hidden layer of C3D Network [25], Two-Stream Network [15], or SlowFast Network [16] given a δ-frame snippet. However, as stated in the above sections, this may cause insufficient or noisy information capture because actions and the agents, who create the actions, usually take place in small spatial regions of the video.

Hence, in this work, we propose a novel action proposal generation model in videos, named Agent-Aware Boundary Network, equipped with a new feature extraction mechanism for the function ϕ namely Contextual Agent-Aware Representation Network (described in Sec. III-B1) to be able to incorporate information from both agents and the interaction between them and surrounding environment. Our proposed ABN with the Contextual Agent-Aware Representation Network can be developed on any backbone such as C3D Network [25], Two-Stream Network [15], or the latest model of SlowFast Network [16]. More details are discussed in section III-B.

B. NETWORK DESIGN

The ABN consists of two sub-networks and is demonstrated in Fig. 1. The first sub-network, **Contextual Agent-Aware Representation Network**, extracts contextual Agent-Aware visual representation of a δ-frame snippet at both global and local levels and is detailed in section III-B1. The second sub-network, **Boundary Generation Network**, takes the first component as the input and generates the action proposals and is described in section III-B2.

1) CONTEXTUAL AGENT-AWARE REPRESENTATION NETWORK

The contextual Agent-Aware representation network extracts contextual video visual representation of a δ-frame snippet, which makes a significant contribution in TAPG. Considering our goal is extracting features for a δ-frame snippet from frame t to frame $t + \delta$, our Agent-Aware representation network is illustrated in Fig. 3 and consists of four steps, i.e. (i) backbone feature extraction, (ii) global feature extraction, (iii) local feature extraction, and (iv) feature fusion. Notably, there are two kinds of feature extracted in our proposed network, i.e., environment feature extracted through step 2 plays a role of global semantic level, and agent feature extracted through step 3 plays a role of local semantic level.

Step 1: Backbone Feature Extraction: The backbone network is used to encode global semantic information of the entire δ-frame snippet. In order to prove the robustness of our proposed ABN, which is independent to the backbone network, we adopt Two-Stream, C3D [25] and SlowFast [16] networks in our experiments. These networks process a snippet of frames through multiple blocks of residual convolutional layers, with each block $B_i \in \{B_i\}_i^4$ produces a feature map $S_i \in \{S_i\}_i^4$. Assume N is the last block of the backbone network, the feature map S_N is then used as input for the two parallel pathways as in the next two steps.

Step 2: Global Feature Extraction: To extract global feature, the feature map S_4 keeps going through average pooling and several fully connected layers until the second last layer, forming a vector which captures the overall scene, namely the global feature ϕ_e. Because all spatial dimensions are processed, this pathway captures the abstract information at the global level of the scene, however, it may not able to capture the details like motions of agents inside.

Step 3: Local Features Extraction: The local features extraction consists of two procedures. First, the local semantic feature vectors of each agent appearing in the video snippet are extracted. Then, all local feature vectors extracted from the first step are fused together to form an agent-aware feature vector. In order to fuse an arbitrary number of local semantic features, we employ a self-attention module with an average pooling layer, which is discussed below.

For local semantic feature vectors extraction, we first detect agent appear in the δ-frame snippet by a human detector. The center frame of the snippet is heuristically selected to feed into the detector because it is least diverged compared to frames at both ends of the snippet. We utilize Faster R-CNN [28] model pre-trained on COCO dataset [47] as our human detector after eliminating all object classes except the 'person' class. Detected human bounding boxes with confidence scores above 0.5 are then used to guide the RoIAlign [48] to extract features from S_4, each feature storing local information about appearance and motion of the corresponding agent, called the local semantic feature.

After a set of local semantic features is formed, we employ a self-attention module to fuse them together into a single local agent-aware feature ϕ_a. The self-attention module looks at the local semantic feature of each agent and assigns up-weights to agents who play important roles in the video snippet or are committing observed actions while assigning down-weights to the minor role agents.

In this step, the Faster R-CNN [28] works as a hard attention module which eliminates all the background and only emphasizes humans or agents moving in the scene. On the

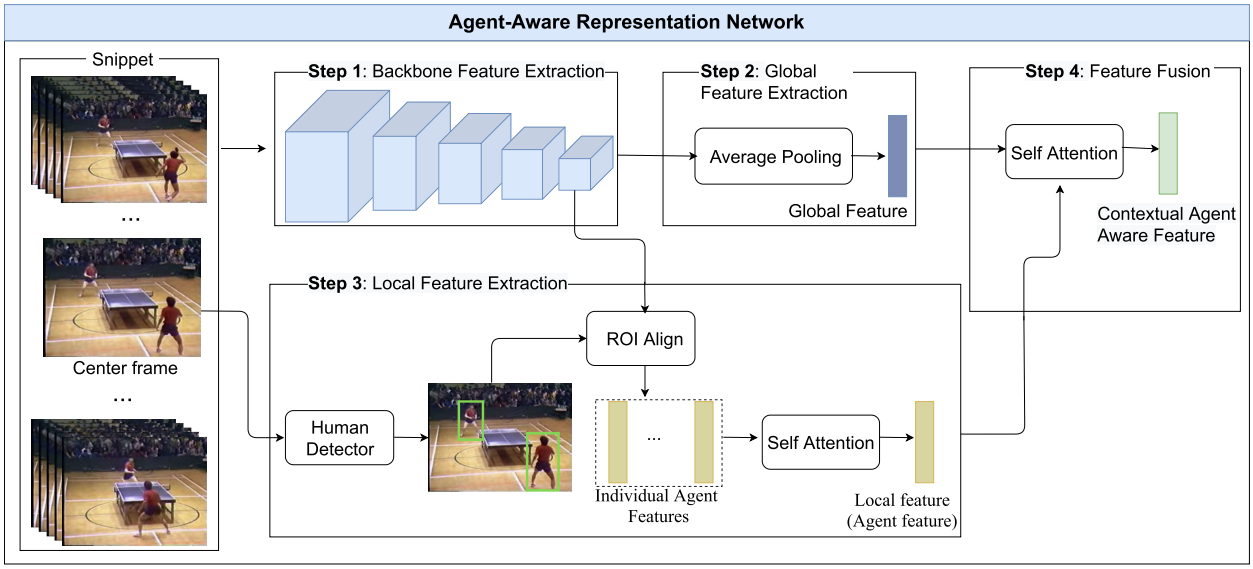


FIGURE 3. An overall architecture of our proposed contextual Agent-Aware representation network which contains four steps. Given a δ-frame snippet, the final video visual feature is conducted by both global feature and local feature.

other hand, the self-attention module works as a soft attention module which helps to concentrate on the right agents but also keeps information of the other agents because the activities we observe may require the interaction between these agents.

Step 4: Feature Fusion: Finally, the environment feature ϕ_e and the agent aware feature ϕ_a are fused by another self-attention module (discussed below). While simultaneously processing these features, the self-attention module would re-weight them by a proper ratio, which helps the overall model to know which type of information to consider while reasoning the action proposals, i.e. deciding whether to emphasize on local information of the agents or global information of the scene.

a: SELF-ATTENTION MODULE

In both TAPG and TAD, an arbitrary number of agents may appear in each snippet, which leads to difficulty to combine them into a single feature vector attentively to represent the snippet. Inspired by that problem, we propose a self-attention module which adopts the Transformer Encoder model [49] to learn to re-weight the importance of the semantic features set based on each of themselves and fuse them together by an average pooling operation.

The self-attention module is employed twice within a snippet, i.e. (i) encode the list of individual agent features to a single multi-agent feature and (ii) fuse both the environment feature ϕ_e and the multi-agent feature ϕ_a to a snippet feature f. Generally, a Transformer Encoder model will encode the set of input features $\Gamma = \{\eta_i\}_{i=1}^{\gamma}$ to three matrices of latent states, namely keys $K = (k_i = \theta_k(\eta_i))$, queries $Q = (q_i = \theta_q(\eta_i))$ and values $V = (v_i = \theta_v(\eta_i))$. Notably, in the agent feature extraction (step 2), γ is equivalent to the number of agents, η is corresponding to individual feature and Γ is a single multi-agent feature whereas in the feature fusion (step 4) γ is set as 2 and η is corresponding to environment feature and multi-agent feature and Γ is snippet feature. θ is defined as a fully connected layer. For each query state q_i of an input feature, an attention function defined in Eq. 4 maps a query q_i and a set of key-value pairs (K, V) to an output. The output is computed as a weighted sum of the values, where the weight assigned to each value is computed by a compatibility function of the query with the corresponding key as follows:

$$\mathcal{A}(q_i, K, V) = \text{softmax}(\frac{q_i \cdot K^T}{\sqrt{d_K}})V \tag{4}$$

where d_K is the number of dimensions in key states. Then, an average pooling layer is applied to fuse the resulting matrix $\Gamma_{\mathcal{A}} = \mathcal{A}(q_i, K, V)_{i=1}^{\gamma}$ and form the overall context feature based on input features set.

The proposed self-attention model is utilized in our proposed ABN in a differentiable fashion and is trained along with the other parts of our network in an end-to-end way, hence, the resulting model may be able to properly generate the contextual Agent-Environment feature, which decreases the impact of background information in every snippet.

2) BOUNDARY GENERATION NETWORK

Our ABN belongs to the category of boundary-based approach and the boundary sub-network, i.e. boundary generation network, contains three modules i.e. Base Module, Temporal Assessment Module (TAM) and Proposal Assessment Module (PAM). These modules are described follows:

Base Module first processes the feature sequence F, which is extracted from the video by our contextual Agent-Aware representation network, through several 1D convolutional layers to extract temporal relationships between nearby snippet features. Those 1D convolutional layers are designed with

TABLE 1. The detailed architecture of the boundary generation network which takes the contextual Agent-Aware visual feature F as the input. T and D are the temporal length of the video and maximum duration of proposals in terms of number of snippets. The obtained outputs are O_T and O_P, which are corresponding to boundary-predictions and proposal actionness scores.

ID	Layer		Input	Output
Base Module				
1	1DConv.	$256 \times 3/1$, ReLU	$I : F \times T$	$O_1 : 256 \times T$
2	1DConv.	$128 \times 3/1$, ReLU	$O_1 : 256 \times T$	$O_2 : 128 \times T$
3	1DConv.	$256 \times 3/1$, ReLU	$O_2 : 128 \times T$	$O_3 : 256 \times T$
Temporal Assessment Module (TAM)				
4	1DConv.	$2 \times 3/1$, Sigmoid	$O_3 : 256 \times T$	$O_T : 2 \times T$
Proposal Assessment Module (PAM)				
5	Matching layer		$O_2 : 128 \times T$	$O_5 : 128 \times 32 \times D \times T$
6	3DConv.$512 \times 32 \times 1 \times 1/(32, 0, 0)$, ReLU		$O_5 : 128 \times 32 \times D \times T$	$O_6 : 512 \times 1 \times D \times T$
7	Squeeze		$O_6 : 512 \times 1 \times D \times T$	$O_7 : 512 \times D \times T$
8	2DConv.	$128 \times 1 \times 1/(0, 0)$, ReLU	$O_7 : 512 \times D \times T$	$O_8 : 128 \times D \times T$
9	2DConv.	$128 \times 3 \times 3/(1, 1)$, ReLU	$O_8 : 128 \times D \times T$	$O_9 : 128 \times D \times T$
10	2DConv.	$2 \times 1 \times 1/(0, 0)$, Sigmoid	$O_9 : 128 \times D \times T$	$O_P : 2 \times D \times T$

a stride of 1 and same padding to reserve temporal length of the output feature sequence.

Temporal Assessment Module (TAM) takes the features sequence from base module and estimates probabilities of every temporal location being a starting or ending boundary.

Proposal Assessment Module (PAM) also takes the features sequence from base module and produces two matrices, each of which densely contains the confidence scores of every possible duration at every starting temporal point, but are trained by two different types of loss functions as suggested by [6]. These matrices would have a shape of $D \times T$ with D is the maximum length of the proposals in snippets that we consider and T is the number of snippets. In this work, we set $D = T$ for ActivityNet-1.3 [50] and $D = T/2$ for THUMOS-14 [51] as suggested by [6].

Network architecture of Boundary Generation Network is given in Table 1. In Table 1, base module is represented by layer 1 to layer 3, temporal assessment module is represented by layer 4 and proposal assessment module is represented by layer 5 to layer 10.

C. TRAINING PHASE

1) LABEL GENERATION

We follow [6], [12] to generate the ground truth labels for training process including starting labels, ending labels for TAM training and duration labels for PAM training.

The starting and ending labels are generated for every snippet of the video, which are called $L_S = \{l_n^s\}_{n=1}^T$ and $L_E = \{l_n^e\}_{n=1}^T$, respectively. The boundaries timestamps (starting and ending) of every action instance $a_i = (s_i, e_i)$ are rescaled into T-snippet range by multiplying them with $\frac{T \cdot \text{fps}}{L}$ where fps is the frame rate of the video and the action instance $a_i \in \mathcal{A}$, $\mathcal{A} = \{a_i\}_{i=1}^M$. After rescaling, the action instance a_i becomes a new action instance $a_i^\delta = (s_i^\delta, e_i^\delta)$. For every snippet $t_n \in T$, we denote a temporal region $r_n = [t_n - \frac{1}{2}, t_n + \frac{1}{2}]$. Analogously, for every pair of boundaries (s_i^δ, e_i^δ) of action a_i^δ, we denote regions $r_i^s = [s_i^\delta - \frac{3}{2}, s_i^\delta + \frac{3}{2}]$ and $r_i^e = [e_i^\delta - \frac{3}{2}, e_i^\delta + \frac{3}{2}]$ as their corresponding starting region and ending region. By this formulation, we have two sets of regions $R_S = \{r_i^s\}_{i=1}^M$ and $R_E = \{r_i^e\}_{i=1}^M$ for starting and ending boundaries, respectively. Finally, starting label l_n^s and ending label l_n^e of a snippet t_n are calculated by the following functions:

$$l_n^s = \begin{cases} 1, & \sum_{i=1}^{M} \frac{r_n \cap r_i^s}{r_i^s} \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

$$l_n^e = \begin{cases} 1, & \sum_{i=1}^{M} \frac{r_n \cap r_i^e}{r_i^e} \geq 0.5 \\ 0, & \text{otherwise} \end{cases}$$

The duration labels for a video are gathered into a matrix $L_D \in [0, 1]^{D \times T}$ where D is the maximum length of proposals being considered in number of snippets, as suggested in [6], we set $D = T$ in all of our experiments. With an element at position (t_i, t_j) stands for a proposal action $a_p = (t_s = \frac{t_j \cdot T}{t_v}, t_e = \frac{(t_j + t_i) \cdot T}{t_v})$, it will be assigned by 1 if its Interaction-over-Union with any ground truth action in $\mathcal{A} = \{a_i\}_{i=1}^M$ reach a local maximum, or 0 otherwise.

2) LOSS FUNCTION

As mentioned in section III-B2, TAM will generate probabilities vectors of starting and ending boundaries ($P_S \in \mathbb{R}^T$ and $P_E \in \mathbb{R}^T$), while PAM will generate two actionness scores matrices $P_D^{cc} \in \mathbb{R}^{D \times T}$ and $P_D^{cr} \in \mathbb{R}^{D \times T}$. These four outputs are trained simultaneously by different loss functions as following:

$$\mathcal{L}_{TAM} = \mathcal{L}_{bin}(P_S, L_S) + \mathcal{L}_{bin}(P_E, L_E) \quad (5)$$

$$\mathcal{L}_{PAM} = \mathcal{L}_{bin}(P_D^{cc}, L_D) + \lambda_{reg} \cdot \mathcal{L}_2(P_D^{cr}, L_D) \quad (6)$$

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}_{TAM} + \lambda_2 \cdot \mathcal{L}_{PAM} \quad (7)$$

As proposed by [6], [12], we set $\lambda_{reg} = 10$ and $\lambda_1 = \lambda_2 = 1$, furthermore, $\mathcal{L}_{bin}$ is a weighted binary log-likelihood function to deal with imbalanced number of negative and positive examples in groundtruth labels. Generally, $\mathcal{L}_{bin}(\hat{Y}, Y)$ between prediction $\hat{Y} \in \mathbb{R}^N$ and groundtruth $Y \in \mathbb{R}^N$ is defined as follows:

$$\frac{1}{N}\sum_{i=1}^{N} \alpha^+ \cdot Y_i \cdot \log \hat{Y}_i + \alpha^- \cdot (1 - Y_i) \cdot \log (1 - \hat{Y}_i), \quad (8)$$

where $\cdot$ is multiplication operator. The weighting parameters are automatically set by number of positives and negatives, specifically, $\alpha^+ = \frac{N}{N^+}$ and $\alpha^- = \frac{N}{N^-}$, with N, N^- and N^+ are total number of examples and total number of positive and negative examples, respectively.

D. INFERENCE PHASE

During inference, four outputs are generated by the boundary generation network from the features sequence extracted by our ABN, including P_S, P_E from TAM output (output of layer 4 in Table 1) and P_D^{cc}, P_D^{cr} from PAM output (output of layer 10 in Table 1). Peaking probabilities of starting and ending boundaries from P_S and P_E, which are local maximums, are selected to form initial proposals by pairing every peak starting point with peak ending points behind them and within a pre-defined range. For a proposal formed by t_s and t_e boundaries with duration $d_p = t_e - t_s$, its score s_p, as proposed in [6], are computed as follows:

$$s_p = P_S[t_s] \cdot P_E[t_e] \cdot \sqrt{P_D^{cc}[d_p, t_s] \cdot P_D^{cr}[d_p, t_s]} \quad (9)$$

Then, with a list of proposals and their scores, a Soft-NMS [52] is applied to eliminate highly overlapped proposals before outputting the final list of proposals.

IV. EXPERIMENTS

A. DATASETS & METRICS

1) DATASETS

We evaluate our proposed method on two benchmark datasets, namely ActivityNet-1.3 [50] and THUMOS-14 [51].

ActivityNet-1.3 [50] is a large scale dataset for benchmarking methods in human activity understanding problems, in which, action proposals and action detection are the centers of attention. The dataset contains 200 distinct activity classes and a total of 849 hours of videos collected from YouTube. ActivityNet-1.3 [50] contains roughly 20K untrimmed videos which are divided into training, validation and test sets with the ratio of 0.5, 0.25 and 0.25, respectively. Each video in ActivityNet-1.3 [50] is annotated with one or more temporal intervals accommodating any activity out of 200 activities of interest. Due to the unavailability of annotations on test splits, we compare and report performances of our approach and other state-of-the-art methods on the validation set, unless otherwise stated.

THUMOS-14 [51], on the other hand, is primarily a dataset for action recognition. Fortunately, a track of action localization and detection are derived from a portion of its videos. Concretely, 200 and 214 untrimmed videos are extracted from the validation and test sets of THUMOS-14 [51], respectively, for training and testing methods in action detection.

2) METRICS

To comprehensively evaluate the performance of the proposed ABN, we not only evaluate it in action proposals generation task, but also in action detection task.

For action proposals generation, on both ActivityNet-1.3 [50] and THUMOS-14 [51], we measure AR with different Average Numbers (ANs) of proposals, denoted as AR@AN. AN is defined as the average number of proposals kept by every video in the dataset. Temporal intersection over union (tIoU) is used as the sole metric to classify a proposal. We follow the traditional practice, tIoU thresholds set from 0.5 to 0.95 with a step size of 0.05 are used on ActivityNet-1.3, while tIoU thresholds set from 0.5 to 1.0 with a step size of 0.05 are used on THUMOS-14. On ActivityNet-1.3 particularly, we report the score of area under the Average Recall (AR) versus Average Number of Proposals per Video curve (AUC), with the average number of proposals ranges from 0 to 100.

For action detection task, following previous works [6], [12], we mainly evaluate our method by Mean Average Precision at tIoU (mAP@tIoU), with the tIoU in ranges [0.5, 0.75, 0.95] and [0.3, 0.4, 0.5, 0.6, 0.7] for ActivityNet-1.3 and THUMOS-14, respectively. At a specified tIoU, Average Precision is calculated for every action class and then averaged up to mAP@tIoU. On ActivityNet-1.3 particularly, we also report the Average mAP which is averaged among all mAP@tIoU scores.

For comparability purposes, we follow the same setting up which was described in [6]. We re-scaled all videos to 1600 frames by linear interpolation and extracted features for every separate snippet with length $\delta = 16$ frames. Therefore, every video sequence will be represented by a feature sequence with the length of exactly 100 features.

B. IMPLEMENTATION DETAILS

On AcitivityNet [50], we benchmark our proposed ABN on C3D [25], SlowFast [16] and Two-Stream [26] backbones, the first two backbones are pre-trained on Kinetics-400 [63] dataset, while the last one is pre-trained on recognition track of ActivityNet-1.3 [50]. The feature map S_N of each backbone is extracted for the local feature extraction step, which has 2048 dimensions, 2304 dimensions and 400 dimensions for C3D, SlowFast and Two-Stream, respectively. We always keep this feature size throughout our proposed network and output the Contextual Agent-Environment Feature of the same size as that of backbone feature.

On THUMOS-14 [51], for fair comparisons with prior works [6], [12], [14], we employ C3D [25] and Two-Stream [26] networks as the backbones of our proposed ABN. The C3D [25] backbone is pre-trained on Kinetics-400 [63],

TABLE 2. Comparison in terms of AR@AN and AUC between our proposed ABN against other state-of-the-art TAPG methods on validation set and test set of ActivityNet-1.3 dataset. The best performance is shown in bold. The second best performance is shown in *italic*.

Methods	Year	Feature	AR@100 (val)	AUC (val)	AUC (test)
TCN [3]	ICCV2017	Two-Stream [15]	-	59.58	61.56
MSRA [4]	CVPRW2017	P3D [39]	-	63.12	64.18
SSTAD [36]	BMVC2017	C3D [25]	73.01	64.40	64.80
CTAP [2]	ECCV2017	Two-Stream [15]	73.17	65.72	-
BSN [12]	ECCV2018	Two-Stream [15]	74.16	66.17	66.26
SRG [46]	TCSVT 2019	Two-Stream [15]	74.65	66.06	-
MGG [13]	CVPR2019	I3D [41]	74.54	66.43	66.47
BMN [6]	ICCV2019	Two-Stream [15]	75.01	67.10	67.19
DBG [14]	AAAI2020	Two-Stream [15]	*76.65*	68.23	68.57
BSN++ [53]	ACCV2020	Two-Stream [15]	76.52	68.26	-
TSI++ [54]	ACCV2020	Two-Stream [15]	76.31	68.35	68.85
MR [55]	ECCV2020	Two-Stream [15]	75.27	66.51	–
SSTAP [56]	CVPR2021	Two-Stream [15]	75.20	67.23	–
Our Proposed ABN	–	Two-Stream	76.39	68.84	**69.40**
	–	Slow-Fast	76.64	*69.08*	*69.30*
	–	C3D	**76.72**	**69.16**	69.26

whereas Two-Stream backbone is pre-trained on the action recognition track of ActivityNet-1.3 [50]. Output feature map S_4 from C3D and Two-Stream backbones are 2048 and 400 dimensions, respectively.

In the local feature extraction step, we adopt a Faster R-CNN [28] model pre-trained on COCO dataset [47] to detect human bounding boxes for generating local features later by RoI alignment with the feature map S_4.

The Transformer Encoders we used in Self-Attention Module for contextually merging local features into the local agent-aware feature or merging the local agent-aware feature with the global feature together share the same architecture of 4 attention heads and 1 transformer layer.

For every experiment, we trained our model for 10 epochs with initial learning rate of 0.0001 and Adam optimizer, the best performed model on validation set is chosen for further comparison.

In addition, we apply an augmentation where any groundtruth video, whose groundtruth actions having average length higher than a factor of τ_{upper} of its length, will be discarded. Contrarily, any groundtruth video, whose groundtruth actions having average length lower than a factor of τ_{lower} of its length, will be duplicated. We empirically observe that with $\tau_{upper} = 0.98$ and $\tau_{lower} = 0.3$, those augmentations during training will help the network achieve better performance and more robust on both datasets of ActivityNet-1.3 [50] and THUMOS-14 [51].

C. PERFORMANCE ON TAPG

1) COMPARE WITH STATE-OF-THE-ART METHODS

Table 2 shows the comparison in terms of AR@AN (AN = 100) and AUC between the ABN against other state-of-the-art methods on both validation set and test set of ActivityNet-1.3 [50] dataset. Our performance is given in the last three rows of Table 2. Compared against other state-of-the-art approaches, our proposed ABN obtains better performance on both AR@AN and AUC metrics regardless the backbone network. Concretely, the ABN outperforms BMN with 2.21%, 2.11% and 2.07% in terms of AUC on test set, when using Two-Stream [26], SlowFast [16] and C3D [25] backbones, respectively. With the most recent state-of-the-art, namely DBG [14], our proposed network makes the gaps of 0.83%, 0.73% and 0.69% on AUC on testing set with Two-Stream, SlowFast and C3D backbones, respectively.

Additionally, Table 3 summarizes the performances of our proposed ABN and other state-of-the-arts on testing set of THUMOS-14 [51] in terms of AR@AN (AN is in a set of [50, 100, 200, 500, 1000]). Our experiments are conducted on C3D [25] and Two-Stream [26] backbone networks following previous works for fair comparisons. Besides, inspired by [14], we also measure the performance of our method with both NMS and Soft-NMS in post-processing phase. Surprisingly, our proposed ABN outperforms all the previous works with very large margins as shown in Table 3. We also noticed that using NMS will help the method to have better performances on top 200 proposals, while Soft-NMS will help the method to have better performances on more proposals e.g. 1000 ones.

Fig. 4 illustrates some qualitative results of the generated proposals by ABN and BMN [6] on ActivityNet-1.3 [50]. The experimental results show that Agent-Aware Boundary Network generates much better proposals, which almost perfectly cover the groundtruth events and tightly fit with their boundaries.

2) GENERALIZABILITY OF PROPOSALS

One of the most important properties of a TAPG method is generating high quality proposals for unseen

TABLE 3. Comparisons with other state-of-the-art TAPG methods on testing set of THUMOS-14 dataset in terms of AR@AN, where SNMS stands for Soft-NMS. The best performance is shown in bold. The second best performance is shown in *italic*.

Feature	Methods	Year	@50	@100	@200	@500	@1000
C3D	SCNN-prop [8]	CVPR 2016	17.22	26.17	37.01	51.57	58.20
	SST [1]	CVPR 2017	19.90	28.36	37.90	51.58	60.27
	TURN-TAP [9]	ICCV 2017	19.63	27.96	38.34	53.52	60.75
	BSN [12]	ECCV 2018	29.58	37.38	45.55	54.67	59.48
	MGG [13]	CVPR 2019	29.11	36.31	44.32	54.95	60.98
	BMN [6]	ICCV 2019	32.73	40.68	47.86	56.42	60.44
	DBG+SNMS [14]	AAAI 2020	30.55	38.82	46.59	56.42	62.17
	DBG [14]	AAAI 2020	32.55	41.07	48.83	57.58	59.55
	Our Proposed ABN + SNMS	–	*34.25*	*44.01*	*52.05*	**60.57**	**65.39**
	Our Proposed ABN + NMS	–	**36.01**	**45.41**	**52.74**	*59.91*	*62.47*
Two-Stream	TURN-TAG [9]	ICCV 2017	18.55	29.00	39.61	-	-
	CTAP [2]	ECCV 2018	32.49	42.61	51.97	-	-
	BSN [12]	ECCV 2018	37.46	46.06	53.21	60.64	64.52
	MGG [13]	CVPR 2019	39.93	47.75	54.65	61.36	64.06
	BMN [6]	ICCV 2019	39.36	47.72	54.70	62.07	65.49
	DBG+SNMS [14]	AAAI 2020	37.32	46.67	54.50	*62.21*	66.40
	DBG+NMS [14]	AAAI 2020	40.89	49.24	55.76	61.43	61.95
	SSTAP [56]	CVPR2021	*41.01*	*50.12*	*56.69*	–	**68.81**
	Our Proposed ABN + SNMS	–	40.87	49.09	56.24	**63.53**	*67.29*
	Our Proposed ABN + NMS	–	**44.89**	**51.86**	**57.36**	61.67	62.59

TABLE 4. Generalizability evaluation on ActivityNet 1.3. The best performance is shown in bold.

Methods	Training Data	Seen		Unseen	
		AR@100	AUC	AR@100	AUC
BSN [12]	Seen+Unseen	72.40	63.80	71.84	63.99
	Seen	72.42	64.02	71.32	63.38
BMN [6]	Seen+Unseen	72.96	65.02	72.68	65.05
	Seen	72.47	64.37	72.46	64.47
TSI++ [54]	Seen+Unseen	**74.69**	66.54	74.31	66.14
	Seen	73.59	65.60	73.07	65.05
DBG [14]	Un+/Seen	73.30	66.57	67.23	64.59
	Seen	72.95	66.23	64.77	62.18
Our proposed ABN	Seen+Unseen	74.58	**66.96**	**75.25**	**67.49**
	Seen	**74.40**	**66.69**	**73.66**	**65.49**

action categories. We follow the protocol defined in BSN [12] and BMN [6] to evaluate the generalizability of our proposed ABN. There are two un-overlapped action subsets: "Sports, Exercise, and Recreation" and "Socializing, Relaxing, and Leisure" of ActivityNet-1.3 are chosen as seen and unseen subsets separately. With such selection, there are 87 and 38 action categories, 4455 and 1903 training videos, 2198 and 896 validation videos on seen and unseen subsets separately. We first train our ABN on both seen training set and seen+unseen training set and then evaluate it on seen validation set and unseen validation set separately. Table 4 shows the performance of ABN along with the comparison with BSN [12] and BMN [6]. Compare with BSN and BMN, the ABN achieves better generalizability on both seen and unseen validation sets. This proves that our method can be used to generate proposals for activities and actions that it never met during the training phase.

D. PERFORMANCE ON TAD

Another important aspect worth considering is the utilization of proposals in action detection. Following BSN [12] and BMN [6] for a fair comparison, we adopt top-1 video-level classification results generated by method in [57] on ActivityNet-1.3 to label the proposals generated by our method. Meanwhile, we use top-2 video-level classification results generated by UntrimmedNet [64] to label proposals generated by our method on THUMOS-14. The labeled proposals are then evaluated on mAP@tIoU metric as described in Sec. IV-A2.

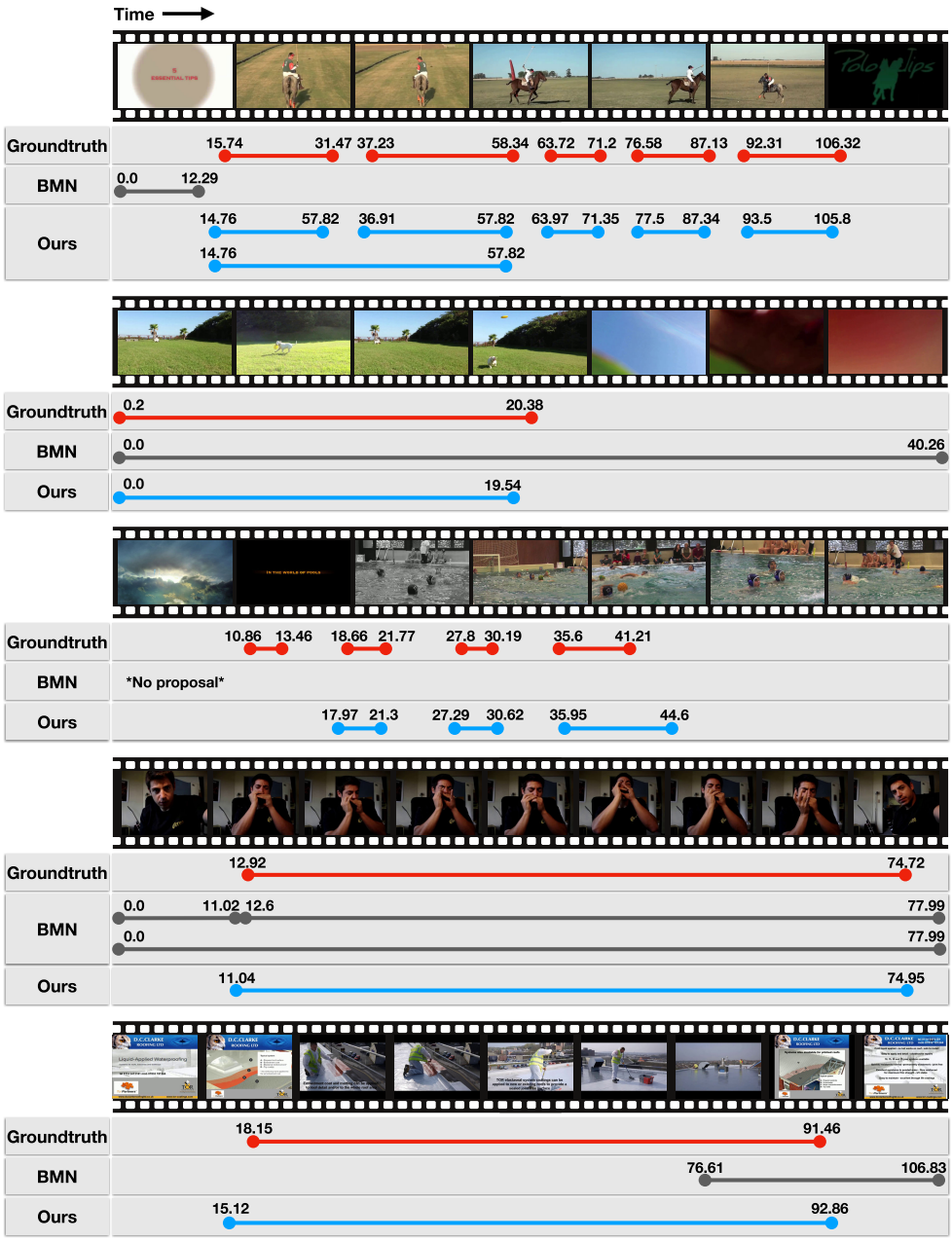


FIGURE 4. Qualitative results of proposals by BMN [6] and our proposed ABN on ActivityNet-1.3 [50], we use our best performed configuration which includes C3D [25] as backbone feature extractor.

Table 5 illustrates the performance of ABN and comparison with other state-of-the-art methods on ActivityNet-1.3 validation set. As we can see, our method outperforms BSN [12], BMN [6] on all settings with a large margin and keeps a good distance with the most recent state-of-the-art mehtod in action detection named GTAD [59] on all settings except mAP@0.75.

The experiment results on THUMOS-14 shown in Table 6 again emphasizes the superior performance of our ABN when compare with other methods including the state-of-the-art method in action detection, namely GTAD [59].

E. ABLATION STUDY

We conduct several ablation studies on the validation set of ActivityNet-1.3 dataset to analyze the contribution of individual feature in the ABN. In addition to different backbone networks, i.e. C3D [25], SlowFast [16] and Two-Stream [26], we have investigated the following ablation configurations for each backbone network.

- Environment Feature Only (Env.): the network relies solely on global features to generate proposals.
- Agent Feature Only (Agent): the network relies completely on the contextual agent-aware features and does not use global feature.

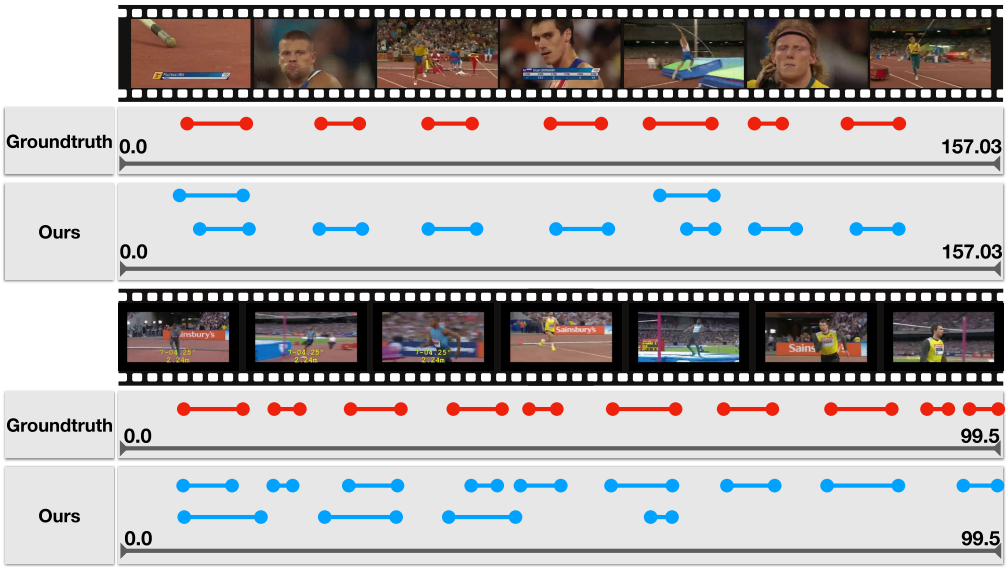


FIGURE 5. Qualitative results of proposals generated by our proposed ABN on THUMOS-14 [51], we use our best performed configuration which includes Two-Stream [15] as backbone feature extractor.

TABLE 5. TAD results on ActivityNet-1.3 in terms of mAP@tIoU and average mAP, where our proposals are combined with video-level classification results generated by [57].

Method		0.5	0.75	0.95	Average
CDC [58]	CVPR2017	43.83	25.88	0.21	22.77
BSN [12]	ECCV2018	46.45	29.96	8.02	30.03
BMN [6]	ICCV2019	50.07	34.78	8.29	33.85
GTAD [59]	CVPR2020	50.36	34.6	9.02	34.09
P-GCN [60]	ICCV2019	42.9	28.1	2.5	27.0
MR [55]	ECCV2020	43.5	33.9	9.2	30.1
BC-GNN [61]	ECCV2020	50.56	34.75	9.37	34.26
TSI++ [54]	ACCV2020	51.2	35.0	6.6	34.2
SSTAP [56]	CVPR2021	50.72	**35.28**	7.87	**34.48**
Our Proposed ABN	–	**51.78**	34.18	**10.29**	34.22

TABLE 6. Performance comparisons between our proposed ABN and the other proposal generation methods in terms of TAD on the testing set of THUMOS-14, where mAP is reported with tIoU set from 0.3 to 0.7 and Unet classifier is used.

Method		0.7	0.6	0.5	0.4	0.3
SST [1]	CVPR2017	4.7	10.9	20.0	31.5	41.2
TURN-TAP [9]	CVPR2017	6.3	14.1	24.5	35.3	46.3
BSN [12]	ECCV2018	20.0	28.4	36.9	45.0	53.5
BMN [6]	ICCV2019	20.5	29.7	38.8	47.4	56.0
MGG [13]	CVPR2019	21.3	29.5	37.4	46.8	53.9
DBG [14]	AAAI2019	21.7	30.2	39.8	49.4	57.8
GTAN [62]	CVPR2019	–	–	38.8	47.2	57.8
GTAD [59]	CVPR2020	23.4	30.8	40.2	47.6	54.5
BC-GNN [61]	ECCV2020	23.1	31.2	40.4	49.1	57.1
SSTAP [56]	CVPR2021	22.8	32.8	42.3	51.5	58.4
Our proposed ABN	–	**25.56**	**37.04**	**46.12**	**53.95**	**59.87**

- Both Environment and Agent Feature: both global feature and local feature are used and fused by Self-Attention Module. This network configuration is actually our proposed ABN.

TABLE 7. Ablation studies on the effectiveness of each component in the proposed ABN on ActivityNet-1.3 dataset with environment feature (Env.) and agent feature (Agent) in terms of AR@AN (*AN* = 100) and AUC. The ablation study is conducted on various features i.e C3D, SlowFast and Two-Stream.

Feature		AR@1	AR@10	AR@100	AUC
C3D [25]	Env.	33.58	57.50	75.07	67.55
	Agent	30.14	53.80	72.76	64.54
	Agent-Env. (Ours)	**33.87**	**59.21**	**76.72**	**69.16**
SlowFast [16]	Env	33.74	57.11	75.59	67.85
	Agent	32.24	52.77	72.67	64.27
	Agent-Env. (Ours)	**34.09**	**58.95**	**76.64**	**69.08**
Two-Stream [15]	Env.	32.59	56.72	74.94	67.14
	Agent	32.27	52.61	72.72	64.16
	Agent-Env. (Ours)	**33.61**	**58.86**	**76.39**	**68.84**

Table 7 provides the results in terms of AR@AN (AN = 1, 10, 100) and AUC metrics of the ABN under different feature configurations on ActivityNet-1.3. In Environment Feature Only (Env.) configuration, there is only a global information about the environment of entire video frames, whereas in Agent Feature Only (Agent) configuration, there is only a local information about the agents. Table 7 has shown that each feature has its own contribution and the combination of both features exhibits the best performance. This result proves that our aforementioned observations are valid and reliable.

V. CONCLUSION

In this paper, we proposed a novel contextual Agent-Aware Boundary Network (ABN) for the TAPG. Our ABN contains two components corresponding to Agent-Environment representation network and boundary generation network. The first component extracts the contextual visual representation of the video whereas the second component with boundary-based mechanism aims at evaluating confidence scores of densely distributed proposals. Different from the previous works, which apply backbone network into the entire video frame, the video visual representation in the proposed ABN involves two parallel pathways: (i) the local pathway, which plays at the agents level and tells about where the agents are and what the agents are doing; (ii) the global pathway, which plays at an environment level and tells about how the environment affects after receiving the actions from the agents as well as the relationship between the agents, actions, and the environment. The experiments have demonstrated that our proposed ABN outperforms state-of-the-art proposal generation methods with C3D, SlowFast and Two-Stream backbone networks on both ActivityNet-1.3 and THUMOS-14 datasets. Our superior performance relies on both global feature and local feature, and demonstrates the robustness of the proposed ABN regardless of the backbone network as well as the effectiveness of our two-pathway contextual Agent-Environment visual representation. Additionally, our proposed method can also be generalized well to generate proposals for activities and actions that it never sees in training phase. Therefore, our method also shows the superior results in further applications like action detection task.

REFERENCES

[1] S. Buch, V. Escorcia, C. Shen, B. Ghanem, and J. C. Niebles, "SST: Single-stream temporal action proposals," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2911–2920.

[2] J. Gao, K. Chen, and R. Nevatia, "CTAP: Complementary temporal action proposal generation," in *Proc. ECCV*, Sep. 2018, pp. 68–83.

[3] X. Dai, B. Singh, G. Zhang, L. S. Davis, and Y. Q. Chen, "Temporal context network for activity localization in videos," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5727–5736.

[4] T. Yao, Y. Li, Z. Qiu, F. Long, Y. Pan, D. Li, and T. Mei, "MSR Asia MSM at activitynet challenge 2017: Trimmed action recognition, temporal action proposals and densecaptioning events in videos," in *Proc. CVPR Workshops*, 2017, pp. 1–6.

[5] A. Richard and J. Gall, "Temporal action detection using a statistical language model," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 3131–3140.

[6] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "BMN: Boundary-matching network for temporal action proposal generation," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 3889–3898.

[7] F. C. Heilbron, J. C. Niebles, and B. Ghanem, "Fast temporal activity proposals for efficient detection of human actions in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1914–1923.

[8] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1049–1058.

[9] J. Gao, Z. Yang, C. Sun, K. Chen, and R. Nevatia, "TURN TAP: Temporal unit regression network for temporal action proposals," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 3648–3656.

[10] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster R-CNN architecture for temporal action localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1130–1139.

[11] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2933–2942.

[12] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "BSN: Boundary sensitive network for temporal action proposal generation," *Proc. ECCV*, Sep. 2018, pp. 3–21.

[13] Y. Liu, L. Ma, Y. Zhang, W. Liu, and S.-F. Chang, "Multi-granularity generator for temporal action proposal," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3604–3613.

[14] C. Lin, J. Li, Y. Wang, Y. Tai, D. Luo, Z. Cui, C. Wang, J. Li, F. Huang, and R. Ji, "Fast learning of temporal action proposal via dense boundary generator," in *Proc. AAAI*, 2020, pp. 11499–11506.

[15] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. NeurIPS*, 2014, pp. 568–576.

[16] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6201–6210.

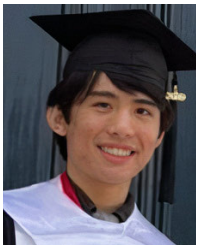
[17] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing videos by exploiting temporal structure," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4507–4515.

[18] T. Yao, T. Mei, and Y. Rui, "Highlight detection with pairwise deep ranking for first-person video summarization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 982–990.
[19] Z. Zhang, Y. Shi, C. Yuan, B. Li, P. Wang, W. Hu, and Z.-J. Zha, "Object relational graph with teacher-recommended learning for video captioning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13278–13288.
[20] S. Liu, Z. Ren, and J. Yuan, "SibNet: Sibling convolutional encoder for video captioning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 9, pp. 3259–3272, Sep. 2021.
[21] J. Lee and S. Abu-El-Haija, "Large-scale content-only video recommendation," in *Proc. IEEE Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2017, pp. 987–995.
[22] B. Xiong, Y. Kalantidis, D. Ghadiyaram, and K. Grauman, "Less is more: Learning highlight detection from video duration," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1258–1267.
[23] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6479–6488.
[24] J.-X. Zhong, N. Li, W. Kong, S. Liu, T. H. Li, and G. Li, "Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 1237–1246.
[25] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015.
[26] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 1933–1941.
[27] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proc. 25th ACM Int. Conf. Multimedia*, Oct. 2017, pp. 988–996.
[28] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NeurIPS*, 2015, pp. 91–99.
[29] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2999–3007.
[30] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement," 2018, *arXiv:1804.02767*. [Online]. Available: http://arxiv.org/abs/1804.02767
[31] A. Klaeser, M. Marszalek, and C. Schmid, "A spatio-temporal descriptor based on 3D-gradients," in *Proc. Brit. Mach. Vis. Conf.*, 2008, p. 275.
[32] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proc. 15th Int. Conf. Multimedia (MULTIMEDIA)*, Sep. 2007, pp. 357–360.
[33] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. ECCV*. Berlin, Germany: Springer, 2008, pp. 650–663.
[34] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. ECCV*. Berlin, Germany: Springer, 2006, pp. 428–441.
[35] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, vol. 25. Stateline, NV, USA, Dec. 2012, pp. 1097–1105.
[36] S. Buch, V. Escorcia, B. Ghanem, and J. C. Niebles, "End-to-end, single-stream temporal action detection in untrimmed videos," in *Proc. Brit. Mach. Vis. Conf.*, 2017, pp. 93.1–93-12.
[37] A. Diba, V. Sharma, and L. Van Gool, "Deep temporal linear encoding networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2329–2338.
[38] G. Varol, I. Laptev, and C. Schmid, "Long-term temporal convolutions for action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 6, pp. 1510–1517, Jun. 2018.
[39] Z. Qiu, T. Yao, and T. Mei, "Learning spatio-temporal representation with pseudo-3D residual networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5534–5542.
[40] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6299–6308.
[41] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6546–6555.
[42] J. Donahue, L. A. Hendricks, M. Rohrbach, S. Venugopalan, S. Guadarrama, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 4, pp. 677–691, Apr. 2017.
[43] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici, "Beyond short snippets: Deep networks for video classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4694–4702.
[44] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
[45] T. Lin, X. Zhao, and Z. Shou, "Temporal convolution based action proposal: Submission to ActivityNet 2017," 2017, *arXiv:1707.06750*. [Online]. Available: http://arxiv.org/abs/1707.06750
[46] H. Eun, S. Lee, J. Moon, J. Park, C. Jung, and C. Kim, "SRG: Snippet relatedness-based temporal action proposal generator," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4232–4244, Nov. 2020.
[47] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollar, and L. Zitnick, "Microsoft coco: Common objects in context," in *Proc. ECCV*, Sep. 2014, pp. 740–755.
[48] K. He, G. Gkioxari, P. Dollar, and R. Girshick, "Mask R-CNN," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2961–2969.
[49] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, Red Hook, NY, USA: Curran Associates, 2017, pp. 6000–6010.
[50] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 961–970.
[51] Y.-G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar. (2014). *THUMOS Challenge: Action Recognition With a Large Number of Classes*. [Online]. Available: http://crcv.ucf.edu/THUMOS14/
[52] N. Bodla, B. Singh, R. Chellappa, and L. S. Davis, "Soft-NMS—Improving object detection with one line of code," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 5561–5569.
[53] H. Su, W. Gan, W. Wu, J. Yan, and Y. Qiao, "BSN++: Complementary boundary regressor with scale-balanced relation modeling for temporal action proposal generation," in *Proc. ACCV*, 2020, pp. 2602–2610.
[54] S. Liu, X. Zhao, H. Su, and Z. Hu, "TSI: Temporal scale invariant network for action proposal generation," in *Proc. ACCV*, Nov. 2020.
[55] P. Zhao, L. Xie, C. Ju, Y. Zhang, Y. Wang, and Q. Tian, "Bottom-up temporal action localization with mutual regularization," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 539–555.
[56] X. Wang, S. Zhang, Z. Qing, Y. Shao, C. Gao, and N. Sang, "Self-supervised learning for semi-supervised temporal action proposal," in *Proc. CVPR*, Jun. 2021, pp. 1905–1914.
[57] Y. Xiong, L. Wang, Z. Wang, B. Zhang, H. Song, W. Li, D. Lin, Y. Qiao, L. V. Gool, and X. Tang, "CUHK & ETHZ & SIAT submission to activitynet challenge 2016," 2016, *arXiv:1608.00797*. [Online]. Available: https://arxiv.org/abs/1608.00797
[58] Z. Shou, J. Chan, A. Zareian, K. Miyazawa, and S.-F. Chang, "CDC: Convolutional-De-Convolutional networks for precise temporal action localization in untrimmed videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1417–1426.
[59] M. Xu, C. Zhao, D. S. Rojas, A. Thabet, and B. Ghanem, "G-TAD: Sub-graph localization for temporal action detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10156–10165.
[60] R. Zeng, W. Huang, C. Gan, M. Tan, Y. Rong, P. Zhao, and J. Huang, "Graph convolutional networks for temporal action localization," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7094–7103.
[61] Y. Bai, Y. Wang, Y. Tong, Y. Yang, Q. Liu, and J. Liu, "Boundary content graph neural network for temporal action proposal generation," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2020, pp. 121–137.
[62] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei, "Gaussian temporal awareness networks for action localization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 344–353.

[63] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017, *arXiv:1705.06950*. [Online]. Available: http://arxiv.org/abs/1705.06950

[64] L. Wang, Y. Xiong, D. Lin, and L. V. Gool, "Untrimmednets for weakly supervised action recognition and detection," 2017, *arXiv:1703.03329*. [Online]. Available: https://arxiv.org/abs/1703.03329

KHOA VO (Member, IEEE) received the B.S. degree (Hons.) in computer science from VNUHCM-University of Science. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville. From October 2019 to March 2020, he was an Internship Student with the National Institute of Informatics, Japan. He has coauthored in publications appearing in conferences and journals, including ICASSP, CVPR Workshops, and *Applied Sciences*. His research interests include image captioning, object detection, and temporal action detection.



KASHU YAMAZAKI received the B.S. degree *(summa cum laude)* in mechanical engineering from the University of Arkansas, Fayetteville, in 2020, where he is currently pursuing the master's degree with the Department of Computer Science and Computer Engineering. His research interests include objects detection, segmentation, and video captioning. He has coauthored in publications appearing in conferences and journals, including ICASSP, ICPR, and *Artificial Intelligence Review*. He is a member of Tau Beta Pi.

SANG TRUONG received the B.S. degree in automation and control engineering from International University, VNUHCM-University of Science, in 2019. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Computer Engineering, University of Arkansas, Fayetteville. From September 2019 to May 2021, he worked in finance and machine learning. His research interests include electrocardiography, objects detection, and actions detection.

MINH-TRIET TRAN (Member, IEEE) received the B.Sc., M.Sc., and Ph.D. degrees in computer science from the University of Science, VNU-HCM, in 2001, 2005, and 2009, respectively.

In 2001, he joined the University of Science. He was a Visiting Scholar with the National Institutes of Informatics (NII), Japan, from 2008 to 2010, and the University of Illinois at Urbana-Champaign (UIUC), from 2015 to 2016. He is currently the Vice President of the University of Science, VNU-HCM, and the Director of John von Neumann Institute, VNU-HCM. He is also a Membership Development and Student Activities Coordinator of IEEE Vietnam. He is also a member of the Advisory Council for Artificial Intelligence Development of Ho Chi Minh City and the Vice President of Vietnam Information Security Association (VNISA), South Branch. His research interests include cryptography and security, computer vision and human–computer interaction, and software engineering.

AKIHIRO SUGIMOTO (Member, IEEE) received the B.S., M.S., and Dr. Eng. degrees in mathematical engineering from The University of Tokyo, in 1987, 1989, and 1996, respectively. He joined the Hitachi Advanced Research Laboratory, in 1989, and then temporally moved to the Advanced Telecommunications Research Institute International (ATR), Japan, in 1991. In 1995, he returned to the Hitachi Advanced Research Laboratory, where he lead a project on content-based image retrieval supported by the Ministry of International Trade and Industry, Japan. In 1999, he moved to the Graduate School of Informatics, Kyoto University, as a Lecturer. Since 2002, he has been with the National Institute of Informatics (NII), where he is currently a Full Professor. From 2006 to 2007, he was also a Visiting Professor with the University of Paris-Est Marne-la-Vallée, France. He has published more than 150 peer-reviewed journal/international conference papers.

He is a member of ACM. He received the Best Paper Award from the Information Processing Society of Japan, in 2001, and the Institute of Electronics, Information and Communication Engineers (IEICE), in 2011. He has been an Associate Editor of *International Journal of Computer Vision*, since 2014. He served several top-tier conferences, including IEEE International Conference on Computer Vision (ICCV), European Conference on Computer Vision (ECCV), Asian Conference on Computer Vision (ACCV), International Conference on Pattern Recognition (ICPR), and International Conference of 3D Vision (3DV) as an Area Chair, a Program Chair, and a General Chair. He is a Regular Reviewer of international conferences/journals in *Computer Vision*, *Artificial Intelligence*, and *Pattern Recognition*.

NGAN LE (Member, IEEE) received the bachelor's and master's degrees in computer science from the VNUHCM-University of Science, Vietnam, in 2005 and 2009, respectively, and the master's and Ph.D. degrees in electrical and computer engineering from Carnegie Mellon University (CMU), in 2015 and 2018, respectively. She was a Research Associate with the Department of Electrical and Computer Engineering (ECE), CMU, from 2018 to 2019. She is currently the Director of the Artificial Intelligence and Computer Vision Laboratory and an Assistant Professor with the Department of Computer Science and Computer Engineering, University of Arkansas. Her publications appear in top conferences, including CVPR, MICCAI, ICCV, SPIE, IJCV, and ICIP, and premier journals, including *International Journal of Computer Vision* (IJCV), *Journal of Electrical Systems and Automation* (JESA), IEEE Transactions on Image Processing (TIP), *Pattern Recognition* (PR), *Journal of Dance and Somatic Practices* (JDSP), and IEEE Transactions on Information Forensics and Security (TIFS). She has coauthored more than 60 journals, conference papers, and book chapters, more than nine patents, and inventions. Her current research interests include image understanding, video understanding, computer vision, robotics, machine learning, deep learning, reinforcement learning, biomedical imaging, and singlecell-RNA.

She is currently a Guest Editor of *Scene Understanding in Autonomous Driving* (Frontier) and *Artificial intelligence in Biomedicine and Healthcare* (MDPI). She Co-Organized the Deep Reinforcement Learning Tutorial for Medical Imaging at MICCAI 2018, and Medical Image Learning with Less Labels and Imperfect Data workshop at MICCAI 2019 and MICCAI 2020. She has served as a Reviewer for more than ten top-tier conferences and journals, including IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), AAAI, CVPR, NIPS, ICCV, ECCV, MICCAI, IEEE Transactions on Image Processing (TIP), *Pattern Recognition* (PR), *Telematics and Informatics* (TAI), and *Image and Vision Computing* (IVC).