Gated-ViGAT: Efficient Bottom-Up Event Recognition and Explanation Using a New Frame Selection Policy and Gating Mechanism

Nikolaos Gkalelis
CERTH-ITI
Thessaloniki, Greece, 57001
gkalelis@iti.gr

Dimitrios Daskalakis
CERTH-ITI
Thessaloniki, Greece, 57001
dimidask@iti.gr

Vasileios Mezaris
CERTH-ITI
Thessaloniki, Greece, 57001
bmezaris@iti.gr

***Abstract*—In this paper, Gated-ViGAT, an efficient approach for video event recognition, utilizing bottom-up (object) information, a new frame sampling policy and a gating mechanism is proposed[1]. Specifically, the frame sampling policy uses weighted in-degrees (WiDs), derived from the adjacency matrices of graph attention networks (GATs), and a dissimilarity measure to select the most salient and at the same time diverse frames representing the event in the video. Additionally, the proposed gating mechanism fetches the selected frames sequentially, and commits early-exiting when an adequately confident decision is achieved. In this way, only a few frames are processed by the computationally expensive branch of our network that is responsible for the bottom-up information extraction. The experimental evaluation on two large, publicly available video datasets (MiniKinetics, ActivityNet) demonstrates that Gated-ViGAT provides a large computational complexity reduction in comparison to our previous approach (ViGAT), while maintaining the excellent event recognition and explainability performance.**[2]

***Index Terms*—Video event recognition, efficient, attention, bottom-up, gating mechanism, frame selection policy.**

I. Introduction

OVER the last years an increasing number of real-world applications in various sectors, such as multimedia [1] and medicine [2], to name a few, resort to automated event recognition techniques in order to increase the quality of the provided services. Deep learning techniques have achieved major performance leaps and new improvements in this domain continue to push the recognition performance limits every year [1], [3], [4]. These methods usually operate in a top-down fashion, i.e., a neural network is trained using the video class labels and entire frames (or video segments) to implicitly learn to focus on the video regions that are mostly related with the occurring event.

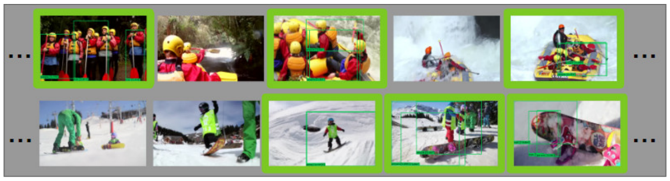


Fig. 1. Gated-ViGAT in action. Frames of two test videos from ActivityNet [10] are shown, belonging to the events “Rafting” (top row) and “Snowboarding” (bottom row). Gated-ViGAT classifies correctly both videos using only three frames (i.e. the ones shown within a green rectangle), derived using our new frame selection policy. The selected frames are relevant to the event and at the same diverse, providing a broad view of the video event, and, thus, a better input for the categorization; e.g., in the bottom row, we can see different skiers and a snowboard. Moreover, bottom-up (object) information provides additional cues for understanding why a video is correctly classified or not.

Studies in cognitive science have suggested that humans interpret complex scenes by selecting a subset of the available sensory information in a bottom-up manner, most probably in order to reduce the complexity of scene analysis [5]–[7]. To this end, bottom-up event recognition approaches, which support the classifier by providing the main objects depicted in the frames, are recently getting increasing attention [8], [9]. These methods not only improve the recognition accuracy but also provide object- and frame-level explanations about the classifier's outcome. However, due to the extraction and processing of the bottom-up (object) information, these methods have a quite high computational cost at inference time, restricting their applicability in applications with strict latency constraints.

Inspired from recent efficient top-down approaches [1], which try to reduce the computational cost of high-capacity 2D or 3D convolutional neural networks (CNNs), here, we extend our previously proposed method, ViGAT [9], using a new frame selection policy and a gating mechanism. Thus, in contrast to ViGAT, the proposed approach, called hereafter Gated-ViGAT, extracts bottom-up information from only a small fraction of the sampled frames as shown in Fig. 1. The proposed frame selection policy utilizes an explanation and a dissimilarity measure (similarly to works in other domains, e.g. [11]) to select the frames that better repre-

This work was supported by the EU Horizon 2020 programme under grant agreement 101021866 (CRiTERIA).

[1] Source code is made publicly available at: https://github.com/bmezaris/Gated-ViGAT

sent the event depicted in the video as well as provide a diverse overview of it. Additionally, we replace the CNN-based gating mechanism of [1] with one that combines both convolution and attention [9], [12] in order to be able to process sequences of frames (not only individual frames as in [1]) and thus capture more effectively both the short- and long-term dependencies of the event occurring in the video. Consequently, the proposed Gated-ViGAT, continues to achieve a high recognition performance, as ViGAT, but with a significant computational complexity reduction. Moreover, contrarily to efficient top-down approaches, e.g. [1], [13], it can provide explanations about the classification outcome, as we demonstrate with a comprehensive qualitative study. We evaluate the proposed method in two large, publicly available video datasets (MiniKinetics [14], ActivityNet [10]), verifying its efficacy. In summary, our major contributions are:

- We present a new frame selection policy and a gating mechanism, and adapt them to our recently proposed bottom-up approach achieving a considerable computational complexity reduction at inference stage.
- The proposed Gated-ViGAT retains the high recognition performance of ViGAT, outperforming the best top-down approaches, and in comparison to them, can provide object- and frame-level explanations about the event recognition outcome.

II. Related Work

We survey video recognition approaches that are mostly related to ours. For a broader literature survey the interested reader is referred to [15], [16].

A. Event and action recognition

A major focus on this area is the design of approaches that capture more effectively the long-term dependencies of events/actions in videos. For instance, in [17], a non-local module modifies a 2D ResNet backbone in order to better capture the action dynamics in videos. Similarly, in [3], an attentive pooling mechanism is applied in various CNN networks to combine frame-level action recognition scores. In [18], a mechanism inserted in 3D-CNNs adaptively adjusts the temporal resolution of feature maps depending on the complexity of the action. In [4], a local- and a global-branch are used to extract semantic and temporal action information, respectively. Video vision transformer (ViViT) [19] factorizes attention to spatial and temporal dimensions in order to effectively process long video sequences. Differently from the above approaches, ViGAT [9] uses an object detector to extract bottom-up (i.e. object) information from video frames, a Vision Transformer (ViT) backbone to derive a feature representation for each object and frame, and an attention-based head network to recognize and explain events in video. The above methods have considerably improved the event/action recognition performance, however, employ networks with quite high computational complexity imposing limitations to their widespread applicability.

B. Frame selection policies

The utilization of frame selection policies for decreasing the computational complexity of event/action recognition approaches is an area currently receiving increasing attention [1], [13], [20]–[23]. AdaFrame [20] exploits a policy gradient method to select future frames for efficient video recognition. SCSampler [24] utilizes the audio modality and an agent to discard redundant video clips. Similarly, an audio-based previewing tool is used by ListenToLook [22] to select the most salient video frames. In [21], the frame sampling policy is formulated as multiple parallel Markov decision processes and learned using multi-agent reinforcement learning (MARL). SMART [23] selects the most discriminant frames using a multi-frame attention and relation network. FrameExit [1] combines a deterministic frame sampling policy with conditional exiting, i.e., frames are processed until a sufficiently confident decision is reached. The above policies have successfully applied to several top-down video recognition approaches; however, the applicability of such policies to methods that utilize bottom-up video information to the best of our knowledge is an unexplored topic.

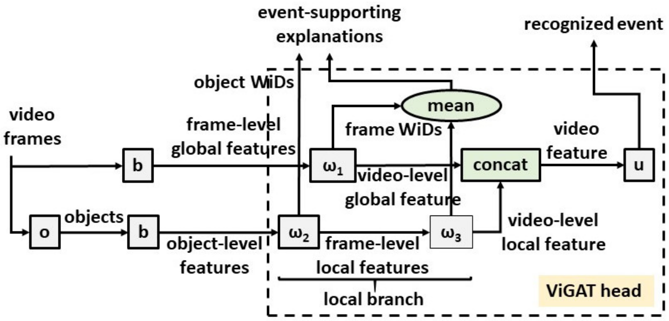


Fig. 2. ViGAT's block diagram [9]. ViGAT consists of an object detector o, a network backbone b and the ViGAT head. The latter consists of three GAT blocks, ω_1, ω_2, ω_3, that process the global (frame-level) and the local (object-level) features, and a dense layer u that uses the video-level feature to recognize the event occurring in the video. Moreover, using the weighted in-degrees (WiDs) of GAT blocks' adjacency matrices, ViGAT can provide event-supporting explanations at object- and frame-level. Gated-ViGAT utilizes the pretrained components of ViGAT and introduces a gating component to build a new architecture for efficient event recognition and explanation.

III. Proposed method

Gated-ViGAT uses the pretrained components of ViGAT [9], whose block diagram is shown in Fig. 2. Specifically, Gated-ViGAT utilises the object detector o, the Vision Transformer (ViT) backbone b [25], and the three GAT blocks [26] and dense layer of the ViGAT head, denoted as ω_1, ω_2, ω_3 and u, respectively. In addition to the above, Gated-ViGAT introduces a gating component [1] consisting of S gates, $g^{(1)}, \dots, g^{(S)}$, whose role is to identify the minimum number of frames to process in the local feature branch (ω_2, ω_3) of the ViGAT, so that the classification performance is retained at the highest degree.

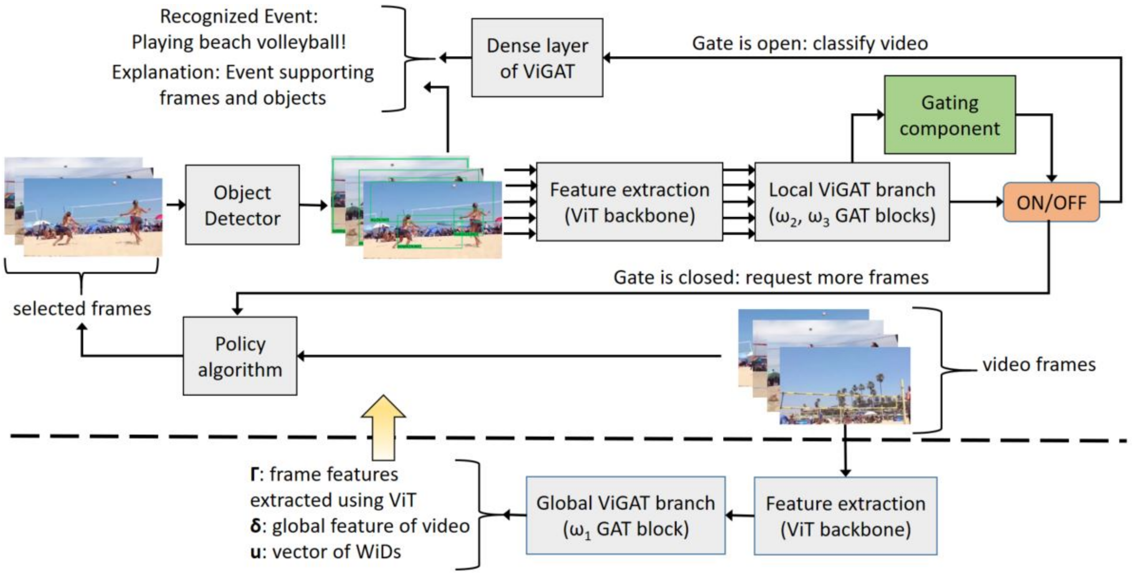

Recognized Event:
Playing beach volleyball!
Explanation: Event supporting frames and objects
Dense layer of ViGAT
Gate is open: classify video
Gating component
Object Detector
Feature extraction (ViT backbone)
Local ViGAT branch (ω_2, ω_3 GAT blocks)
ON/OFF
selected frames
Gate is closed: request more frames
Policy algorithm
video frames
Γ: frame features extracted using ViT
δ: global feature of video
u: vector of WiDs
Global ViGAT branch (ω_1 GAT block)
Feature extraction (ViT backbone)

Fig. 3. Illustration of the proposed Gated-ViGAT. Our architecture utilizes the pretrained components of ViGAT, and the proposed frame selection policy and gating component. The policy algorithm requests sequentially an increasing number of frames until the gating component commits an exiting action, signaling that the provided frames are adequate for recognizing the underlying video event (here "Playing beach voleyball").

The gate networks $g^{(s)}$ are identical, composed of a one-dimensional CNN, a GAT block (as the ones used in ViGAT [9]), and a dense layer that outputs a value in the range [0,1],

$$g^{(s)} : \mathbb{R}^{s+1\times F} \rightarrow [0,1],\ s = 1,\ldots,S, \tag{1}$$

where, F is the feature vector dimensionality produced by the ViT backbone. The sth gate is used to decide whether $Q^{(s)}$ frames are enough to yield a confident decision for the input video. In case that the output of the sth gate is smaller than 0.5 (i.e. the gate is closed), additional frames are selected, and fetched by the local feature branch of ViGAT and the next gate, $g^{(s+1)}$. This process continues until a gate opens (i.e. its output is larger than 0.5) or when the last gate is reached.

The $Q^{(s)}$ frames for the sth gate are selected using a novel frame selection policy (Section III-A2) that utilizes the most salient frame (in terms of explanation) and a dissimilarity criterion in order to achieve high coverage of the event occurring in the video. The gating component is trained end-to-end, while the rest of Gated-ViGAT's components are kept frozen, as explained in the following. This allows us to achieve comparable results while dramatically reducing the computational complexity.

A. Gate training

Suppose an annotated video dataset of N videos and G classes. Each video $\mathcal{V}$ is represented with P frames,

$$\mathcal{V} = \{\mathbf{V}_p\}_{p=1}^{P}, \tag{2}$$

and is associated with a class vector $\mathbf{y} = [y_1, \ldots, y_G]^T \in \{0,1\}^G$, where, $\mathbf{V}_p \in \mathbb{R}^{w\times h\times c}$ is the pth frame, w, h and c are its width, height and number of channels, and y_g is one if the video $\mathcal{V}$ belongs to class g and zero otherwise.

1) Global feature vectors: A ViT backbone b is used to represent each frame with a global feature vector $\gamma_p \in \mathbb{R}^F$,

$$\boldsymbol{\gamma}_p = b(\mathbf{V}_p). \tag{3}$$

The sequence of feature vectors corresponding to the video are then stacked row-wise forming a matrix $\mathbf{\Gamma} = [\boldsymbol{\gamma}_1, \ldots, \boldsymbol{\gamma}_P]^T$ and passed to the block ω_1 of the trained ViGAT head to derive a new feature representation $\boldsymbol{\delta} \in \mathbb{R}^F$ of the video along with a vector of WiDs $\mathbf{u} \in [0,1]^P$,

$$\boldsymbol{\delta}, \mathbf{u} = \omega_1(\mathbf{\Gamma}). \tag{4}$$

Note that the pth element of $\mathbf{u}$ corresponds to the respective video frame and can be used to estimate its importance for explaining model's decision [9].

2) Frame selection policy: The following algorithm is used to select $Q^{(s)}$ frames for the first gate (i.e. s is initially set to 1). Firstly, we compute the index p_1 of the first frame using

$$p_1 = \mathrm{argmax}(\mathbf{u}), \tag{5}$$

i.e., the first frame is the one that corresponds to the highest WiD value, and normalize all the global feature vectors using

$$\tilde{\boldsymbol{\gamma}}_p = \frac{\boldsymbol{\gamma}_p}{\|\boldsymbol{\gamma}_p\|}, \tag{6}$$

where $\|\boldsymbol{\gamma}\|$ is the L_2 norm of any vector $\boldsymbol{\gamma}$. A dissimilarity score α_p of the selected frame with all other frames of the video is then computed using

$$\alpha_p = \frac{1}{2}(1 - \tilde{\boldsymbol{\gamma}}_{p_1}^T \tilde{\boldsymbol{\gamma}}_p). \tag{7}$$

Note that $\alpha_p \in [0,1]$ because $\tilde{\boldsymbol{\gamma}}_{p_1}^T \tilde{\boldsymbol{\gamma}}_p \in [-1,1]$ as it is the cosine distance between the vectors $\tilde{\boldsymbol{\gamma}}_{p_1}$ and $\tilde{\boldsymbol{\gamma}}_p$. Next, both the dissimilarity scores α_p and the WiD values in $\mathbf{u}$ are scaled in the range $[0, 1]$ using min-max normalization,

$$\tilde{\alpha}_p = \frac{\alpha_p - \alpha_{min}}{\alpha_{max} - \alpha_{min}}, \quad (8)$$

$$\tilde{u}_p = \frac{u_p - u_{min}}{u_{max} - u_{min}}, \quad (9)$$

where, $\alpha_{max} = \max(\boldsymbol{\alpha})$, $\alpha_{min} = \min(\boldsymbol{\alpha})$, $u_{max} = \max(\mathbf{u})$, $u_{min} = \min(\mathbf{u})$ and $\boldsymbol{\alpha} = [\alpha_1, \dots, \alpha_P]^T$. The dissimilarity scores are then multiplied with the respective normalized WiDs to update the vector of WiDs,

$$u_p = \tilde{u}_p \odot \tilde{\alpha}_p, \quad (10)$$

where $\odot$ denotes element-wise multiplication. The same procedure (Eqs. (5) to (10)) is followed to select the rest $Q^{(s)} - 1$ frames for the sth gate. Along the different gates, the frames are added incrementally, i.e., for gate $s + 1$, the $Q^{(s)}$ frames selected for gate s are retained and the same procedure (Eqs. (5) to (10)) is followed to select the $Q^{(s+1)} - Q^{(s)}$ additional frames for gate $s + 1$. The number of frames per gate, $Q^{(1)}, \dots, Q^{(S)}$, is a design parameter defined by the user.

3) Local feature vectors: Suppose $\{\mathbf{V}_{p_\iota}\}_{\iota=1}^{Q^{(s)}}$ are the frames selected for gate s. The ViGAT object detector is used to derive K objects from each selected frame, represented with a bounding box, an object class label and the associated degree of confidence (DoC). The ViT backbone is applied to represent each object with a feature vector; these feature representations are sorted in descending order using their DoC values and stacked column-wise to form a matrix $\mathbf{X}_{p_\iota} \in \mathbb{R}^{K \times F}$ for the p_ιth selected frame,

$$\mathbf{X}_{p_\iota} = [\mathbf{x}_{p_\iota,1}, \dots, \mathbf{x}_{p_\iota,K}]^T. \quad (11)$$

The above matrix is fetched by the block ω_2 to derive a "local" feature vector $\boldsymbol{\eta}_{p_\iota} \in \mathbb{R}^F$ and a vector of WiDs $\boldsymbol{\phi}_{p_\iota} \in [0, 1]^K$,

$$\boldsymbol{\eta}_{p_\iota}, \boldsymbol{\phi}_{p_\iota} = \omega_2(\mathbf{X}_{p_\iota}). \quad (12)$$

The kth element of $\boldsymbol{\phi}_{p_\iota}$ can be used as an indicator for the contribution of the kth object in associating the p_ιth frame with the recognized event [8], [9]. The above features are used to form a matrix $\mathbf{H}^{(s)} = [\boldsymbol{\eta}_{p_1}, \dots, \boldsymbol{\eta}_{p_{Q^{(s)}}}]^T \in \mathbb{R}^{Q^{(s)} \times F}$ and fetched by the block ω_3 to derive a "local" feature vector $\boldsymbol{\varrho}^{(s)} \in \mathbb{R}^F$ for the entire video and with respect to gate s,

$$\boldsymbol{\varrho}^{(s)} = \omega_3(\mathbf{H}^{(\mathbf{s})}). \quad (13)$$

4) Gate pseudolabels: The video-level global (4) and local (13) feature vectors are concatenated to form one vector $\boldsymbol{\zeta}^{(s)} \in \mathbb{R}^{2F}$ for the entire video with respect to gate s,

$$\boldsymbol{\zeta}^{(s)} = [\boldsymbol{\delta}; \boldsymbol{\varrho}^{(s)}]. \quad (14)$$

This feature vector is fetched by the dense layer u of the trained ViGAT to derive a vector $\hat{\mathbf{y}} = [\hat{y}_1, \dots, \hat{y}_G]^T$, where $\hat{y}_g \in [0, 1]$ is the degree of confidence that the video belongs to class g. The standard cross-entropy or the categorical cross-entropy loss for single- or multilabeled datasets, respectively, is used to compute the loss l for the specified video. A pseudolabel $o^{(s)} \in \{0, 1\}$ for video with respect to gate s is then derived using [1]

$$o^{(s)} = \begin{cases} 1 & \text{if} \quad l \leq \epsilon^{(s)}, \\ 0 & \text{else} \end{cases} \quad (15)$$

where, $\epsilon^{(s)}$ determines the minimum loss required to exit gate s (and thus the entire gating component), defined as, $\epsilon^{(s)} = \beta \exp \frac{s}{2}$, and a β is scalar parameter.

5) Gate loss: The S gates are trained using the corresponding gate pseudolabels (15) and the binary cross-entropy losses, $l_{bce}()$, summed along all gates [1],

$$\mathcal{L} = \frac{1}{S} \sum_{s=1}^{S} l_{bce}(g^{(s)}(\mathbf{Z}^{(s)}), o^{(s)}) \quad (16)$$

where, $\mathbf{Z}^{(s)} \in \mathbb{R}^{s+1 \times F}$, $\mathbf{Z}^{(1)} = [\boldsymbol{\delta}, \boldsymbol{\varrho}^{(1)}]^T$ and $\mathbf{Z}^{(s)} = [\boldsymbol{\delta}, \boldsymbol{\varrho}^{(1)}, \dots, \boldsymbol{\varrho}^{(s)}]^T$ for $s > 1$.

B. Event recognition and explanation

During the inference stage, the matrix $\acute{\mathbf{Z}}^{(s)}$ for the test video $\acute{\mathcal{V}}$ is generated for $s = 1, \dots, S$, sequentially and each time the output of the respective gate is inquired to decide weather to exit the gating component, or, request the generation of additional local feature vectors. Suppose that s^* is the gate for which the video exits the gating component, i.e., $g^{(s^*)}(\acute{\mathbf{Z}}^{(s^*)}) > 0.5$. At this event, the derived global (4) and local (13) feature vectors are concatenated to form a feature vector $\acute{\boldsymbol{\zeta}}^{(s^*)} = [\acute{\boldsymbol{\delta}}; \acute{\boldsymbol{\varrho}}^{(s^*)}]$ for the entire test video (14), which is fetched by the dense layer u of the trained ViGAT to classify the video to one of the G classes. Moreover, as explained in previous works [8], [9] the derived WiDs can be used to provide explanations about the model's event recognition outcome. To this end, explanations at frame-level are produced using the top frames selected by the proposed frame selection policy (Section III-A2). On the other hand, the vector of WiDs $\acute{\boldsymbol{\phi}}$ (12) is exploited to derive explanations at object-level, e.g., by selecting the objects corresponding to the WiDs with highest values [8].

IV. Experiments

A. Datasets

We run experiments on two large, publicly available video datasets: i) MiniKinetics [14] is a subset of the Kinetics dataset [27], consisting of 200 action classes, 80K training and 5K testing video clips. Each clip is sampled from a different YouTube video, has 10 seconds duration and is annotated with a single event/action class label. ii) ActivityNet v1.3 [10] is a popular multilabel video benchmark consisting of 200 event/action classes (including a large number of high-level events), and approximately 10K, 5K and 5K videos for training, validation and testing, respectively. Most videos are 5 to 10 minutes long. As the testing-set labels are not publicly available, the evaluation is performed on the so called validation set, as typically done in the literature.

B. Setup

Uniform sampling is first applied to represent each video with a sequence of $N = 30$ frames for MiniKinetics (e.g. as in [4], [9], [28]) and $N = 120$ frames for ActivityNet (e.g. as in [9], [21]). As explained in Section III, the following components are utilized from ViGAT [9]: a) the Faster R-CNN [29] object detector o pretrained on ImageNet1K and finetuned on Visual Genome dataset, b) the ViT-B/16 backbone b pretrained on ImageNet11K and fine-tuned on ImageNet1K, c) the three GAT blocks, ω_1, ω_2, ω_3, pretrained on MiniKinetics or ActivityNet, depending on the dataset used in the experimental evaluation. As in ViGAT, the number of objects K (11) to extract with the object detector o is set to 50.

The number of gates S of the gating component and the length of frame sequences $Q^{(s)}$ corresponding to the different gates are set to $S = 5$, $\{Q^{(s)}\}_{s=1}^{5} = \{2, 4, 6, 8, 10\}$ and $S = 6$, $\{Q^{(s)}\}_{s=1}^{6} = \{9, 12, 16, 20, 25, 30\}$ for the MiniKinetics and ActivityNet experiment, respectively. We used one more gate and larger frame sequences on ActivityNet because it contains arbitrarily large videos in contrast to MiniKinetics where all videos are rather short (10 secs duration). In both datasets, the gating component is trained for 40 epochs using an initial learning rate of 10^{-4} multiplied by 0.1 at epochs 16 and 35. Similarly to other works in the literature, the recognition performance is measured using the mean average precision (mAP) and top-1 accuracy on ActivityNet and MiniKinetics, respectively. All experiments were run on PCs with an Intel i5 CPU and a single RTX3090 NVIDIA GPU.

TABLE I
PERFORMANCE COMPARISON ON MINIKINETICS [14] (*NOTE THAT FRAMEEXIT IS EVALUATED ON THE MINIKINETICS VARIANT OF [13])

	top-1(%)
TBN [30]	69.5
BAT [7]	70.6
MARS (3D ResNet backbone) [31]	72.8
Fast-S3D (Inception backbone) [14]	78.0
ATFR (X3D-S backbone) [18]	78.0
ATFR (R(2+1)D backbone) [18]	78.2
RMS (SlowOnly backbone) [28]	78.6
ATFR (I3D backbone) [18]	78.8
Ada3D (I3D backbone on Kinetics) [32]	79.2
ATFR (3D Resnet backbone) [18]	79.3
CGNL (Modified ResNet backbone) [17]	79.5
TCPNet (ResNet backbone on Kinetics) [3]	80.7
LgNet (R3D Backbone) [4]	80.9
FrameExit (EfficientNet backbone) [1]*	75.3
ViGAT [9]	**82.1**
Gated-ViGAT (proposed)	*81.3*

C. Event recognition results

We compare the proposed Gated-ViGAT against the best performing approaches in the literature on the two datasets, namely, TBN [30], BAT [7], MARS [31], Fast-S3D [14], ATFR [18], RMS [28], Ada3D [32], CGNL [17], TCPNet [3], LgNet [4], FrameExit [1], AdaFrame [20], ListenToLook [22], LiteEval [33], SCSampler [24], AR-Net [13], MARL [21] and ViGAT [9]. The recognition performance of the various methods on MiniKinetics and ActivityNet are shown in Tables I and II, respectively. Most of these methods utilize a ResNet type backbone pretrained on ImageNet; when this is not the case we provide the name of the used backbone in brackets. We also denote the best and second best recognition rate with bold and italic fonts, respectively. From these results we see that Gated-ViGAT outperforms all previous methods (which are all top-down) except ViGAT. This confirms that Gated-ViGAT in comparison to the top-down approaches, utilizes effectively the complementary discriminant event information from the bottom-up features, achieving a performance gain; and additionally, can provide comprehensive explanations as we show in the next subsection.

TABLE II
PERFORMANCE COMPARISON ON ACTIVITYNET [10].

	mAP(%)
AdaFrame [20]	71.5
ListenToLook [22]	72.3
LiteEval [33]	72.7
SCSampler [24]	72.9
AR-Net [13]	73.8
FrameExit [1]	77.3
AR-Net (EfficientNet backbone) [13]	79.7
MARL (ResNet backbone on Kinetics) [21]	82.9
FrameExit (X3D-S backbone) [1]	87.4
ViGAT [9]	**88.1**
Gated-ViGAT (proposed)	*87.5*

TABLE III
PERFORMANCE COMPARISON IN TERMS OF COMPUTATIONAL COMPLEXITY (TFLOPS) BETWEEN VIGAT AND GATED-VIGAT ON TWO DATASETS.

	ViGAT	Gated-ViGAT
MiniKinetics	34.4	**8.7**
ActivityNet	137.4	**24.8**

As mentioned above, ViGAT slightly outperforms the proposed Gated-ViGAT, by 0.8% mAP and 0.6% top-1 accuracy in MiniKinetics and ActivityNet, respectively. This is expected, as in contrast to ViGAT that extracts bottom-up information from every sampled frame, Gated-ViGAT extracts bottom-up information from only a small fraction of them. However, due to this fact, Gated-ViGAT is much more efficient than ViGAT at inference time. To quantify this, we used the the Fvcore Flop Counter [34] to measure the computational complexity during inference of the above methods in terms of FLOPs (floating point operations) on both MiniKinetics and ActivityNet. From the obtained results, shown in Table III, we see that Gated-ViGAT provides a dramatic reduction in TFLOPs over ViGAT, specifically, of approximately $\times 4$ and $\times 5.5$ reduction, in MiniKinetics and ActivityNet, respectively. This is because, the most heavy components of both architectures, i.e., the Faster R-CNN object detector and ViT backbone, which are used to extract and process the bottom-up information (i.e. the $K = 50$ objects per frame), are applied considerably fewer times with Gated-ViGAT. Here, we should also note that as expected, during inference, Gated-ViGAT

TABLE IV

NUMBER OF PROCESSED VIDEOS AND RECOGNITION PERFORMANCE (IN TERMS OF TOP-1(%)) WITH RESPECT TO EACH GATE OF GATED-VIGAT ON MINIKINETICS.

	$g^{(1)}$	$g^{(2)}$	$g^{(3)}$	$g^{(4)}$	$g^{(5)}$
# frames	2	4	6	8	10
# videos	179	686	1199	458	2477
top-1(%)	84.9%	83%	81.1%	84.9%	80.7%

TABLE V

NUMBER OF PROCESSED VIDEOS AND RECOGNITION PERFORMANCE (IN TERMS OF MAP(%)) WITH RESPECT TO EACH GATE OF GATED-VIGAT ON ACTIVITYNET.

	$g^{(1)}$	$g^{(2)}$	$g^{(3)}$	$g^{(4)}$	$g^{(5)}$	$g^{(6)}$
# frames	9	12	16	20	25	30
# videos	793	651	722	502	535	1722
mAP(%)	99.8%	94.5%	93.8%	92.7%	86%	71.6%

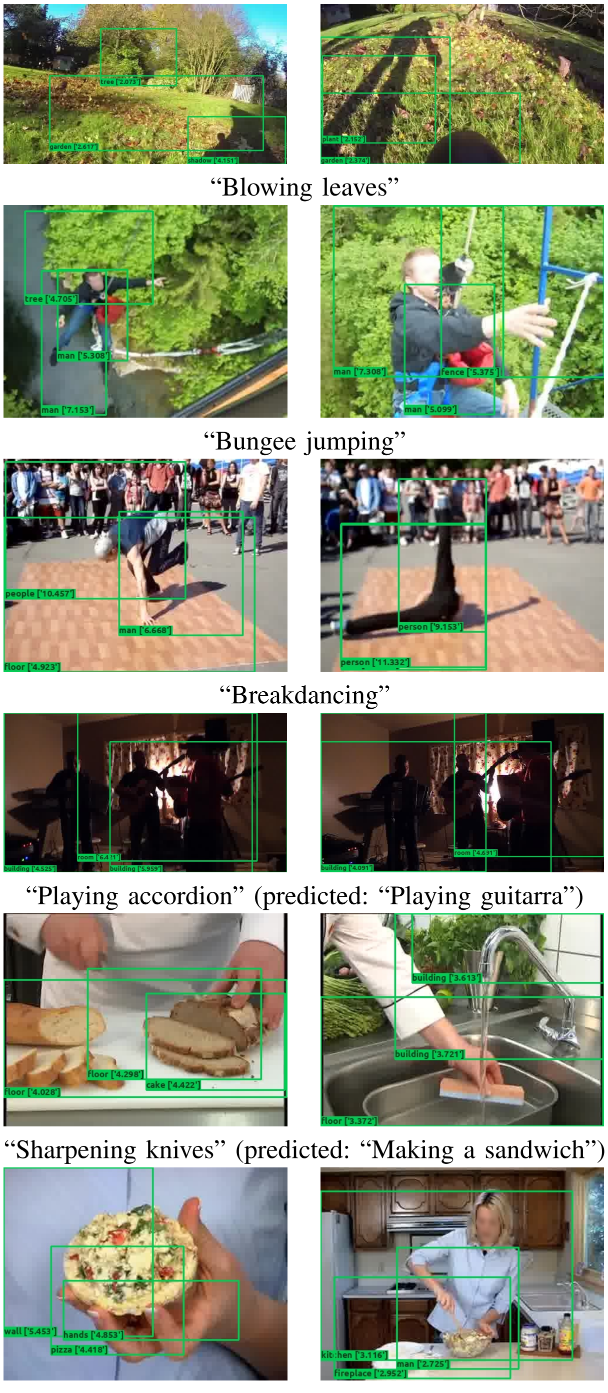
has a higher computational complexity than most top-down approaches (e.g. [3], [4] report requiring in the order of tens or hundreds of GFLOPs at inference). However, as we see in Tables I, II, Gated-ViGAT has outperformed all top-down approaches evaluated here, and, additionally, in comparison to these approaches, can provide explanations concerning the classifier's decision, as shown in Section IV-D.

In order to get additional insight of the proposed method, we provide the number of processed test videos, the number of frames used to extract the bottom-up information and the recognition performance, with respect to the individual gates of Gated-ViGAT, for MiniKinetics and ActivityNet, in Tables IV and V, respectively. Using this information, the average number of frames per video is estimated to 7 and 20 for MiniKinetics and ActivityNet, respectively. This is expected as most videos of ActivityNet are longer than the ones of MiniKinetics (see Section IV-A). We also observe a recognition rate drop as the gate number s increases. This is due to the fact that the most difficult to recognize videos do not exit the gating component early, thus reducing the performance of the gates towards the end of the component; this is in agreement with similar studies in the literature [1].

D. Explanations of event recognition results

In contrast to the majority of the top-down approaches, the proposed approach can provide explanations at object- and frame-level about the event recognition outcome. This is done by exploiting the bottom-up information, i.e., by ranking the extracted objects and sampled frames of the test video using the corresponding WiDs, as explained in Section III-B. For instance, explanations for six test videos of ActivityNet dataset are shown in Fig. 4. Each row of this figure corresponds to a specific test video, where, the first three rows represent correctly classified videos, and the last three misclassified ones. The explanations consist of two frames and three objects per frame, as selected by Gated-ViGAT.

From the obtained results, for instance, in the correctly recognized "Blowing leaves" video (first row), we can see the "person", "garden" and "tree" objects, which are important cues for categorizing this video to the said event; and similarly, the same is true for the objects "man", "tree", "fences", and, "person", "floor", for the correctly classified videos "Bungee jumping" (second row) and "Breakdancing" (third row), re-



"Blowing leaves"

"Bungee jumping"

"Breakdancing"

"Playing accordion" (predicted: "Playing guitarra")

"Sharpening knives" (predicted: "Making a sandwich")

"Making a salad" (predicted: "Having an ice-cream")

Fig. 4. Each row of the figure provides an explanation example produced by Gated-ViGAT for a test video from ActivityNet. An explanation consists of the two most salient frames, and, in each frame the three most salient objects, derived using Gated-ViGAT. The first three examples correspond to correctly classified videos while the last three to miscategorized ones.

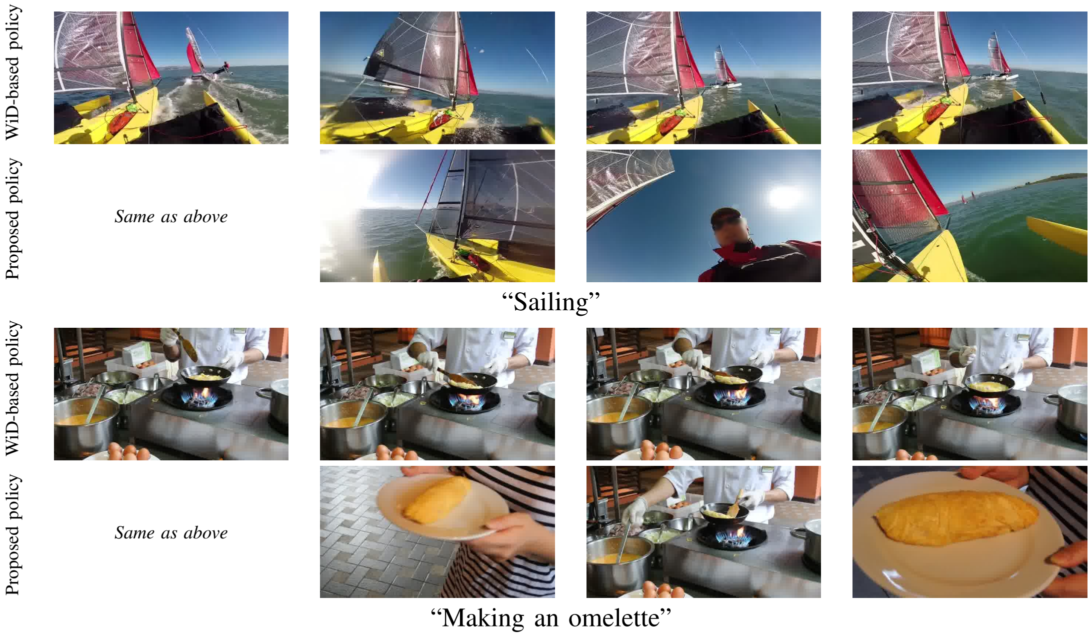


Fig. 5. Proposed vs WiD-based policy. We see that the proposed frame selection policy can provide a broader overview of the event in the video. For instance, in the first video example belonging to a “Sailing” event, the proposed approach provides frames depicting the sailor and the sailing boat from different viewing angles. The same is observed in the second example, labeled “Making an omelette”, where the frames selected with our approach alternate between close and distant views of the event.

spectively. In the video “Playing accordion”, we see that the focus is on the middle person playing guitar, misleading Gated-ViGAT to incorrectly categorize this video as “Playing guitarra”. Finally, regarding the videos “Sharpening knives” and “Making a salad” miscategorized as “Making a sandwich” and “Having an ice-cream”, the most salient frames depict a person cutting bread and holding an object resembling an ice-cream, respectively, explaining why these videos have been misclassified. We should also note that although in some cases the object annotations are inaccurate, which is attributed to object detector imperfections, the focus of Gated-ViGAT (i.e. the regions covered by the bounding boxes of the objects) is on the area where the underlying event is actually occurring, helping the user of the model to understand why the video was categorized in the said event class.

E. Ablation study on the frame selection policy

The following frame selection policies are evaluated. **a)** Random: Θ frames are randomly selected for deriving both the global (4) and local (12) feature vectors. **b)** WiD-based: Θ frames are used, as in (a), but are selected according to their global WiD values (4). **c)** Random local: all P frames are used to derive global feature vectors (4); then, Θ frames are randomly selected for deriving the local feature vectors (12). **d)** WiD-based local: as above, all P frames are used to derive global feature vectors (4); then, Θ frames are selected according to their global WiD values (4), and, these frames are used for computing the local feature vectors (12). **e)** FrameExit policy: Θ frames are selected using the frame selection policy described in [1]. **f)** Proposed policy: all P frames are used to derive global feature vectors (4); then, Θ frames are selected using the proposed frame selection policy (Section III-A2) for deriving the local features. **g)** Gated-ViGAT (proposed): here, in addition to the proposed frame selection policy, as in (f), the gating mechanism is also used for selecting Θ frames per video on average, for deriving local features.

All policies are evaluated using the mAP(%) performance measure for $\Theta = 10, 20, 30$ frames on the ActivityNet. From the obtained results, shown in Table VI, we observe that the Gated-ViGAT policy provides the best performance. We assume that this is because the proposed policy selects diverse frames (due to the exploitation of frame dissimilarity scores (7), (10)) and at the same time with high explanation power (due to the WiD-based values (5), (10)), thus representing well the overall video content. It is also interesting to see that even without utilizing the gating component of Gated-ViGAT our proposed policy (f) achieves the second-best performance, surpassing the FrameExit policy that is the best policy reported in the literature.

TABLE VI
COMPARISON OF DIFFERENT FRAME SELECTION POLICIES IN TERMS OF RECOGNITION PERFORMANCE (MAP(%)) ON ACTIVITYNET.

policy / # frames	10	20	30
Random	83%	85.5%	86.5%
WiD-based	84.9%	86.1%	86.9%
Random on local	85.4%	86.6%	86.9%
WiD-based on local	86.6%	87.1%	87.5%
FrameExit policy [1]	86.2%	*87.3%*	87.5%
Proposed policy	*86.7%*	*87.3%*	*87.6%*
Gated-ViGAT (proposed)	**86.8%**	**87.5%**	**87.7%**

To gain further insight into the proposed frame selection policy (f) (Section III-A2) we provide a visual comparison of it with the WiD-based one (b) in Fig. 5. Specifically, the four most salient frames selected by each policy for two different videos of ActivityNet are shown at each row of the figure. We observe that the frames selected using the WiD-based policy (first and third row) are quite alike, e.g. see the "Sailing" video example where all frames depict the sailing boat from similar viewing angles; and the "Making an omelette" example where all frames show almost the same cooking scene (the cook, frying pan, etc.) in distant view. In contrast, the proposed policy (second and fourth row) selects frames very relevant to the event but also dissimilar to each other, thus obtaining a broader view of the video event. For instance, see the "Sailing" video example, where the selected frames depict the sailor and the sailing boat from quite different viewing angles; and, similarly the "Making an omelette" example, where two frames depict the cook with the frying pan preparing the omelette in distant view, and the other two show in close view the prepared omelette in the plate.

V. Conclusion

We presented Gated-ViGAT, an efficient bottom-up approach for event recognition in video. Specifically, a new policy algorithm for selecting the most salient and diverse frames of the video and a gating component combining convolutional layers and a graph attention network to learn more effectively the long-term dependencies of the video were proposed. The evaluation of Gated-ViGAT on two popular video datasets (ActivityNet, MiniKinetics) showed the efficacy of the method in terms of both recognition performance and computational complexity. Possible future work directions include the investigation of faster object detectors and ViT backbones [35], [36] and the extension of Gated-ViGAT for online event recognition in streaming applications [37].

References

[1] A. Ghodrati, B. E. Bejnordi, and A. Habibian, "FrameExit: Conditional early exiting for efficient video recognition," in *Proc. IEEE/CVF CVPR*, virtual, Jun. 2021, pp. 15 608–15 618.

[2] A. Esteva, K. Chou, S. Yeung *et al.*, "Deep learning-enabled medical computer vision," in *Digit.Med*, vol. 4, Jan. 2021.

[3] Z. Gao, Q. Wang, B. Zhang, Q. Hu, and P. Li, "Temporal-attentive covariance pooling networks for video recognition," in *Proc. NIPS*, vol. 33, Virtual Event, Dec. 2021, pp. 13 587–13 598.

[4] J. Zhou, Z. Fu, Q. Huang, Q. Liu, and Y. Wang, "LgNet: A local-global network for action recognition and beyond," *IEEE Trans. Multimedia*, pp. 1–14, Feb. 2022.

[5] J. K. Tsotsos, S. M. Culhane, W. Y. Kei Wai, Y. Lai, N. Davis, and F. Nuflo, "Modeling visual attention via selective tuning," *Artificial Intelligence*, vol. 78, no. 1, pp. 507–545, Oct. 1995.

[6] L. Itti, C. Koch, and E. Niebur, "A model of saliency-based visual attention for rapid scene analysis," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 11, pp. 1254–1259, 1998.

[7] L. Chi, Z. Yuan, Y. Mu, and C. Wang, "Non-local neural networks with grouped bilinear attentional transforms," in *Proc. IEEE/CVF CVPR*, Seattle, WA, USA, Jun. 2020, pp. 11 801–11 810.

[8] N. Gkalelis, A. Goulas, D. Galanopoulos, and V. Mezaris, "ObjectGraphs: Using objects and a graph convolutional network for the bottom-up recognition and explanation of events in video," in *Proc. IEEE/CVF CVPRW*, Jun. 2021, pp. 3375–3383.

[9] N. Gkalelis, D. Daskalakis, and V. Mezaris, "ViGAT: Bottom-up event recognition and explanation in video using factorized graph attention network," *IEEE Access*, vol. 10, pp. 108 797–108 816, 2022.

[10] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles, "ActivityNet: A large-scale video benchmark for human activity understanding," in *Proc. IEEE/CVF CVPR*, 2015, pp. 961–970.

[11] E. Apostolidis, G. Balaouras, V. Mezaris, and I. Patras, "Summarizing videos using concentrated attention and considering the uniqueness and diversity of the video frames," in *Proc. ICMR*, New York, NY, USA, Jun. 2022, pp. 407–415.

[12] H. Wu, B. Xiao, N. Codella, M. Liu, X. Dai, L. Yuan, and L. Zhang, "CvT: Introducing convolutions to vision transformers," in *Proc. IEEE/CVF ICCV*, Montreal, QC, Canada, Oct. 2021, pp. 22–31.

[13] Y. Meng, C. Lin, R. Panda, P. Sattigeri, L. Karlinsky *et al.*, "AR-Net: Adaptive frame resolution for efficient action recognition," in *Proc. ECCV*, Glasgow, UK, Aug. 2020, pp. 86–104.

[14] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. ECCV*, vol. 11219, Munich, Germany, Sep. 2018, pp. 318–335.

[15] Y. Zhu, X. Li, C. Liu, M. Zolfaghari, Y. Xiong, C. Wu, Z. Zhang, J. Tighe, R. Manmatha, and M. Li, "A comprehensive study of deep video action recognition," *CoRR*, vol. abs/2012.06567, 2020.

[16] P. Pareek and A. Thakkar, "A survey on video-based human action recognition: recent updates, datasets, challenges, and applications," *Artificial Intelligence Review*, vol. 54, pp. 2259–2322, 2021.

[17] K. Yue, M. Sun, Y. Yuan, F. Zhou, E. Ding, and F. Xu, "Compact generalized non-local network," in *Proc. NIPS*, Montréal, Canada, Dec. 2018, pp. 6511–6520.

[18] M. Fayyaz, E. B. Rad, A. Diba, M. Noroozi, E. Adeli, L. V. Gool, and J. Gall, "3D CNNs with adaptive temporal feature resolutions," in *Proc. IEEE/CVF CVPR*, Virtual Event, Jun. 2021, pp. 4731–4740.

[19] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF ICCV*, Montreal, QC, Canada, Oct. 2021, pp. 6836–6846.

[20] Z. Wu, C. Xiong, C. Ma, R. Socher, and L. S. Davis, "AdaFrame: Adaptive frame selection for fast video recognition," in *Proc. IEEE/CVF CVPR*, Long Beach, CA, USA, Jun. 2020, pp. 1278–1287.

[21] W. Wu, D. He, X. Tan, S. Chen, and S. Wen, "Multi-agent reinforcement learning based frame sampling for effective untrimmed video recognition," in *Proc. IEEE/CVF ICCV*, Jul. 2019, pp. 6222–6231.

[22] R. Gao, T. Oh, K. Grauman, and L. Torresani, "Listen to look: Action recognition by previewing audio," in *Proc. IEEE/CVF CVPR*, Seattle, WA, USA, Jun. 2020, pp. 10 454–10 464.

[23] S. N. Gowda, M. Rohrbach, and L. Sevilla-Lara, "SMART frame selection for action recognition," in *Proc. AAAI*, vol. 35(2), May 2021, pp. 1451–1459.

[24] B. Korbar, D. Tran, and L. Torresani, "SCSampler: Sampling salient clips from video for efficient action recognition," in *Proc. IEEE/CVF ICCV*, Seoul, Korea (South), Oct./Nov. 2019, pp. 6231–6241.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," in *Proc. ICLR*, Virtual Event, Austria, May 2021.

[26] P. Velickovic, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, "Graph attention networks," in *Proc. ICLR*, Vancouver, BC, Canada, Apr./May 2018.

[27] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *Proc. IEEE CVPR*, Virtual Event, Jul. 2017.

[28] J. Kim, S. Cha, D. Wee, S. Bae, and J. Kim, "Regularization on spatio-temporally smoothed feature for action recognition," in *Proc. IEEE/CVF CVPR*, Seattle, WA, USA, Jun. 2020, pp. 12 100–12 109.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. NIPS*, vol. 28, 2015.

[30] Y. Li, S. Song, Y. Li, and J. Liu, "Temporal bilinear networks for video action recognition," in *Proc. AAAI*, vol. 33, Honolulu, Hawaii, USA, Jul. 2019, pp. 8674–8681.

[31] N. Crasto, P. Weinzaepfel, K. Alahari, and C. Schmid, "MARS: Motion-augmented RGB stream for action recognition," in *Proc. IEEE/CVF CVPR*, Long Beach, CA, USA, Jun. 2019, pp. 7874–7883.

[32] H. Li, Z. Wu, A. Shrivastava, and L. S. Davis, "2D or not 2D? adaptive 3D convolution selection for efficient video recognition," in *Proc. IEEE/CVF CVPR*, Virtual Event, Jun. 2021, pp. 6155–6164.

[33] Z. Wu, C. Xiong, Y. Jiang, and L. S. Davis, "LiteEval: A coarse-to-fine framework for resource efficient video recognition," in *Proc. NIPS*, Vancouver, Canada, 2019, pp. 7778–7787.

[34] "Flop counter for pytorch models," https://github.com/facebookresearch/fvcore/blob/main/docs/flop_count.md, [Accessed: 2022-10-10].

[35] Y. Li, A. Dua, and F. Ren, "Light-weight RetinaNet for object detection on edge devices," in *Proc. IEEE WF-IoT*, Virtual Event, 2020.

[36] Q. Zhang and Y. Yang, "ResT: An efficient transformer for visual recognition," in *Proc. NIPS*, Virtual Event, Dec. 2021, pp. 15 475–15 485.

[37] C.-Y. Wu and P. Krähenbühl, "Towards long-form video understanding," in *Proc. IEEE/CVF CVPR*, 2021, pp. 1884–1894.