Information Elevation Network for Online Action Detection and Anticipation

Sunah Min[1,2] Jinyoung Moon[1,2*]
[1]Electronics and Telecommunications Research Institute (ETRI), South Korea
[2]University of Science and Technology (UST), South Korea
sunahmin1203@gmail.com, jymoon@etri.re.kr

Abstract

Given a partially observed video segment, online action detection and anticipation aim to identify a current action and forecast future actions, respectively. To detect actions in a streaming video for monitoring applications including surveillance, robot assistants, and autonomous driving, online action detection methods have been proposed. Considering the importance of current action in online action detection, we introduce a novel information elevation unit (IEU) that lifts and accumulates the past information relevant to the current action, to compensate for forgotten essential information. Using the IEUs, we propose an information elevation network (IEN) that effectively identifies a current action and anticipates future actions through the dense prediction of past and current action classes within the video segment. For its practical use in online monitoring applications, our IEN takes visual features extracted from a fast action recognition using only RGB frames because extracting optical flows requires heavy computation overhead. On THUMOS-14 and TVSeries, our IEN outperforms state-of-the-art methods using only RGB frames. Furthermore, on the THUMOS-14 dataset, our IEN outperforms the state-of-the-art methods.

1. Introduction

Because of the dramatic increase of streaming videos from numerous cameras, we require the method to detect every action of interest as soon as it takes place and anticipate a future action in advance. Nowadays the video surveillance industry has experienced dramatic growth with the proliferation of CCTV cameras [13]. In addition, vehicles equipped with multiple cameras installed inside or outside for autonomous driving and camera-equipped assistant robots are increasing. Accordingly, online action detection (OAD) methods [4–8, 18], some of which also include online action anticipation (OAA) [7, 18], have been proposed in recent years.

Online action detection (OAD) aims to identify a current action by using past and current visual information within an untrimmed video segment from a streaming video. OAD methods taking a partially observed video segment as input recognize the current action at the latest frame within the input segment. The video segment consist of a fixed number of chunks, each of which consists of a fixed number of consecutive frames. For each chunk, the methods get a visual feature extracted from a pre-trained action recognition network. The first OAD method [4] including a CNN-based model for a single frame and an LSTM-based model taking 16 frames as input does not fully utilize past visual information. Reinforced encoder-decoder network (RED) [7] based on LSTM and temporal recurrent network (TRN) introducing a new recurrent unit, TRN cell, encode past and current visual information within a video segment with equal weights. However, all the chunks are not equally related to the current action. Therefore, information discrimination network (IDN) [5] and temporal filtering network (TFN) [6] detect the current action for a video segment by emphasizing the visual information from chunks related to the current action. Thus, it is important to take into account the relevance of visual information to the current action in designing an OAD method.

For temporal modeling, LSTM [9] takes temporal inputs at every timestep from the previous hidden and cell states (i.e., h_{t-1} and C_{t-1}) and a visual feature at each timestep (i.e., x_t) and does not consider the current information at $t = 0$ (i.e., x_0), which is an important input for OAD. Because of this, the forget gate of LSTM can lose the past information and accumulated information from the previous hidden state (i.e., h_{t-1}) by using only visual information at each timestep (i.e., x_t), as shown in Figure 1. To this end, we introduce an information elevation unit (IEU), which is an extension of LSTM for OAD. To maintain the information related to the current action, the IEU has the additional information elevation gate that lifts the past information relevant to the current action to the cell state. Specifically, the IEU adds the past information from the previous hidden

*corresponding author

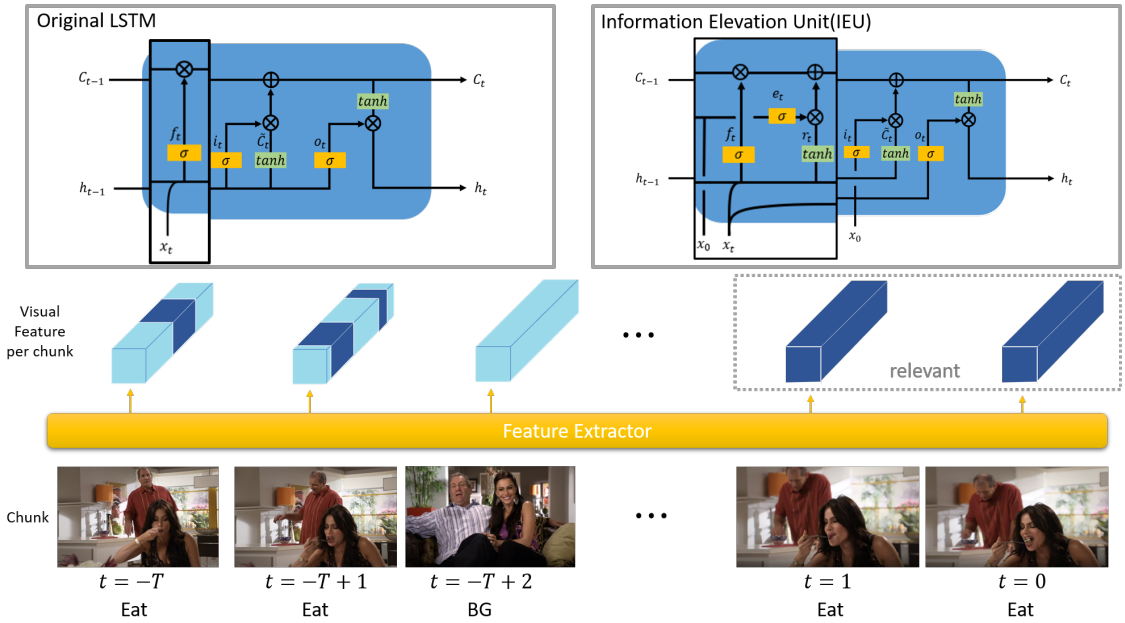

Figure 1. Comparison between the original LSTM and our information elevation unit (IEU). In this video segment, past information at times $-T$ and $-T+1$ is related to the current action. However, when processing the information at time $-T+2$, the LSTM considers only past information from the previous hidden and cell states and at $-T+1$ timestep. In the LSTM, there is a risk of removing accumulated information relevant to current action at the forget gate and accumulating information at the timestep that is irrelevant to current action at the input and output gates. Therefore, the proposed IEU takes the current information together with the past information as inputs and adds an elevation gate to lift and accumulate the past information relevant to the current action.

state (i.e., h_{t-1}) as well as the visual feature at t timestep (i.e., x_t) multiplied by the output from the elevation gate considering the relationship between the past and current information (i.e., h_{t-1} and x_0). Through this, the IEU can reinforce the forgotten past information relevant to the current action.

By using the IEUs, we propose an information elevation network (IEN) that detect a current action and anticipate future actions through the dense prediction of past and current actions. To show the effectiveness and efficiency of our IEN for OAD and OAA, we conduct extensive experiments on two OAD benchmark datasets, THUMOS-14 and TVSeries. Taking the visual features from only RGB frames, the IEN outperforms the state-of-the-art methods using only RGB frames shows comparable performances to the methods using both RGB and flow frames, with a per-frame mAP of 60.4% and mcAP of 81.4% in THUMOS-14 and TVSeries, respectively. In addition, our IEU also outperforms the state-of-the-art methods for OAA.

Our contribution is summarized as follows.

- To maintain the past information relevant to the current action, we introduce IEU, which is an extension of LSTM for OAD, by adding an information elevation gate with an additional input of current information.
- Through the dense prediction of past and current action, our IEN based on IEUs detects a current action and also anticipates future actions with better performance.
- For practical use of OAD for online monitoring applications, we adopt a fast action recognition network using only RGB frames as a feature extractor.
- On the THUMOS-14 and TVSeries datasets, our IEN using RGB frames achieves comparable OAD and outstanding OAA performances to the state-of-the-art methods using RGB and flow frames, respectively.

2. Related Work

2.1. Offline action detection

Offline AD methods detect one or more actions in an untrimmed video. The first OAD method [14] baintroduced a proposal, classification, and localization networks. After generating some video segments of different lengths, the part related to action instances and the part related to the background are identified through the proposal network. The classification network recognizes what the action is in the candidate interval. Finally, the localization network estimates the temporal overlap between GT and the candidate

interval. Lin *et al.* [12] generated proposals by using the estimated start and end scores for action instances rather than generating a proposal of a fixed length.

2.2. Online action detection

OAD was introduced by Geest *et al.* [4]. RED [7] is based on the encoder-decoder structure and predicts the current and future actions by using the past information obtained from the encoder. The cell of a temporal recurrent network, which is proposed by Xu *et al.* [18], generates future information by using the past information and predicts the current action using the past and generated future information. Both the methods consider all chunk with equal priority. To improve this limitation, the information discrimination network (IDN) [5] and temporal filtering network (TFN) [6] processed the information of each timestamp based on its relevance to the current action, which are based on RNN and CNN, respectively. The WOAD [8] provide online action recognition and online action start detection in a weakly-supervised way trained with video-level annotations.

2.3. Action recognition

For a given well-trimmed video containing a single action instance, action recognition (AR) methods predict the probability distribution for N action classes. The AR networks based on 2D and 3D CNNs have shown good performances. Proposed AR networks have been used as feature extractors for various video-related tasks including offline and online temporal action detection. The C3D [16] and I3D [1] features are widely used for video-related task but extracting them requires heavy computation loads. To solve this problem, fast AR methods efficiently extract and model temporal information using shift operation and light-weight motion information, in TSM [11] and PAN [19], respectivly.

3. Proposed Method

Figure 2 shows the overall architecture of our IEN. For a given video segment, the IEN extracts a visual feature for each chunk, converts the visual feature into a visual embedding, and feeds it into each IEU. Each IEU takes past information from previous hidden and cell states related to a visual feature at each timestep t and the current information from the current chunk at time $t = 0$. Using each hidden state at each timestep t, IEN predicts the probability distribution for each chunk and returns the probability distributions for the current action and future actions. In this section, we explain component modules including early embedding, IEU, and classification modules for OAD and OAA, in detail.

Figure 2. The architecture of the IEN. Taking a video segment consisting of $T + 1$ chunks $V = \{c_t\}_{t=-T}^{0}$ as an input, IEN outputs three probability distributions for a current action and two future actions over K action classes and background by using the concatenated hidden states. Merging lines implies concatenation operations between vectors.

3.1. Problem definition

To formulate the online action detection and anticipation problem, we follow the same setting as in previous methods [7, 18]. A chunk is denoted as $c = \{I_n\}_{n=1}^{N}$ of a set of N consecutive frames, where I_n indicates the n_{th} frame. Given a video segment $V = \{c_t\}_{t=-T}^{0}$ including a current (i.e., at $t = 0$) and T past chunks (from $t = -T$ to $t = -1$) as inputs, an OADA model outputs the probability distribution $p_0 = \{p_{0,k}\}_{k=0}^{K}$ of the current action and the probability distributions of N_a future actions at $\{t_a\}_{a=1}^{N_a}$ for the input segment.

3.2. Early embedding module

For each chunk c_t from a video segment V, the early embedding module generates an embedded visual feature x_t. First, for a chunk c_t, we obtain the extracted feature $c'_t \in \mathbb{R}^{d_v}$, where d_v is the dimension of the extracted video feature. We adopt TSM [11] as a feature extractor, which is a fast AR model that efficiently extracts spatio-temporal visual features with only RGB frames. We feed the extracted feature into a fully connected layer to generate the embedded visual feature $x_t = \text{ELU}\left(W_c \cdot c'_t\right) \in \mathbb{R}^{d_e}$, where $W_c \in \mathbb{R}^{d_v \times d_e}$ is a weight matrix, through the ELU activation function [3], and d_e represents the dimension of the

embedded video feature.

3.3. Information Elevation Unit (IEU)

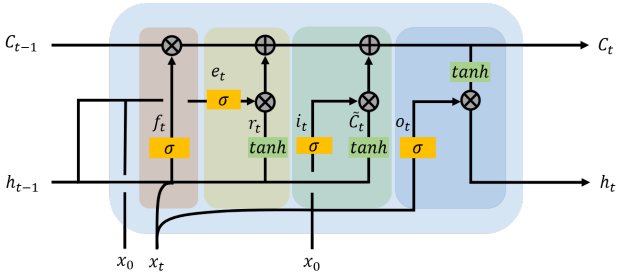

Figure 3. Structure of the information elevation unit (IEU). The IEU is an extended LSTM by adding an elevation gate and taking an additional input, current information x_0. The IEU's forget gate (red box) is the same as the original LSTM and the input and output gate (green box) is similar except the input x_0. The elevation gate (yellow box) is newly added. Merging lines implies concatenation operations between vectors.

We propose a novel RNN unit for OAD, called IEU, which is an extension of the original LSTM unit in two ways. First, the IEU additionally takes the current visual embedding x_t to determine which information is relevant to the current action at all the modified gates except the forget gate. Second, the IEU adds a new gate, called an elevation gate, in order to lift accumulated past information from the previous cell state relevant to the current action, which can be forgotten at the forget gate. The elevation gate takes $x_0 \in \mathbb{R}^{d_e}$ and the previous hidden state $h_{t-1} \in \mathbb{R}^{d_h}$, where d_h represents the dimension of the hidden state, to determine which information from the previous hidden state is relevant to the current action. Figure 3 illustrates the structure of the IEU for OAD. The equations related to all the gates of the IEU are expressed as follows.

$$f_t = \sigma(W_f \cdot [h_{t-1}|x_t]), \tag{1}$$

$$e_t = \sigma(W_e \cdot [h_{t-1}|x_0]), \tag{2}$$

$$r_t = \mathbf{tanh}(W_r \cdot [h_{t-1}|x_t]), \tag{3}$$

$$i_t = \sigma(W_i \cdot [x_t|x_0]), \tag{4}$$

$$\tilde{C}_t = \mathbf{tanh}(W_c \cdot [h_{t-1}|x_t]), \tag{5}$$

$$C_t = (C_{t-1} \cdot f_t) + (r_t \cdot e_t) + (\tilde{C}_t \cdot i_t), \tag{6}$$

$$o_t = \sigma(W_o \cdot [x_t|x_0]), \tag{7}$$

$$h_t = o_t \cdot \mathbf{tanh}(C_t), \tag{8}$$

where $|$ is the concatenation operation, $W_f, W_e, W_r, W_c \in \mathbb{R}^{(d_h+d_e)\times d_h}$, $W_i, W_o \in \mathbb{R}^{(d_e+d_e)\times d_h}$ are learnable parameters, σ is the logistic sigmoid function, and **tanh** is the tangent hyperbolic function.

3.3.1 Forget and Elevation Gates

As in Eq. (1), the forget gate of IEU is identical to that of the original LSTM. The forget gate determines which accumulated information from the cell state should be forgotten without using the current information at time t. However, the current information is directly related to the output of an OAD model. As a result, even though the accumulated past information from the previous cell states can be related to the visual embedding at $t = 0$, there is a risk that the past information can be forgotten if it has less of a relationship with the previous hidden state at the forget gate. To overcome this limitation, the IEU locates an elevation gate next to the forget gate. The current information (i.e., x_0) as well as the past information from the previous hidden state (i.e., h_{t-1}) instead of the past information at timestep t (i.e., x_t) are taken as input. First, as shown in Eq. (2), the elevation gate determines which past information is reinforced through the sigmoid function with the previous hidden state and the current information. The IEU obtains the output of r_t by taking the past information from the previous hidden state and the visual embedding at time t in as inputs in Eq. (3). To compensate for the forgotten information relevant to the current action, the IEU adds the result of multiplying e_t and r_t to the cell state in Eq. (6).

3.3.2 Input and Output Gates

As in Eq. (4) and Eq. (7), the input and output gates of IEU are modified by taking different inputs, the past information at timestep t (i.e., x_t) and the current information (i.e., x_0) instead of the previous hidden state (i.e., h_{t-1}) to control the two gates according to the current information. The input and output gates determine which past information at timestep $t-1$ is relevant to the current information. The output of the input gate affects the next cell state (i.e., C_t) and that of the output gate affects the next hidden state.

3.4. Classification Module

In the classification module, our IEN predicts the probability distributions of the current chunk $c+t$ over K+1 classes by feeding concatenated all $T+1$ hidden states from $-T$ to 0 as:

$$p_t = [p_{t,k}]_{k=0}^{K}, \tag{9}$$

$$p_{t_d} = \mathbf{softmax}(W_{ad} \cdot h_{t_d}), \tag{10}$$

where $[t_d]_{d=-T}^{0}$, $W_{ad} \in \mathbb{R}^{d_h \times d_{cls}}$ is a trainable matrix, and d_{cls} is the dimension of action classes.

$$p_{t_a} = \mathbf{softmax}(W_{t_a,aa} \cdot h_0), \tag{11}$$

where $[t_a]_{a=1}^{N_{aa}}$, $W_{t,aa} \in \mathbb{R}^{d_h \times d_{cls}}$ is a trainable matrix, and d_{cls} is the dimension of action classes. To train our IEN, we design a total classification loss K+1 classes consisting of classification loss for action detection and classification loss for action anticipation by employing the cross-entropy loss as:

$$L_{ad} = -\sum_{i=-T}^{0}\sum_{k=0}^{K} y_{i,k}\log(p_{i,k}), \tag{12}$$

$$L_{aa} = -\sum_{i=1}^{N_a a}\sum_{k=0}^{K} y_{i,k}\log(p_{i,k}), \tag{13}$$

$$L = \alpha \cdot L_{ad} + (1-\alpha)\cdot L_{aa} \tag{14}$$

where α is a balancing parameter and $y_{i,k}$ is the ground-truth label for the i_{th} timestep. Finally, our IEN returns p_0 = $[p_{0,k}]_{k=0}^{K}$ as its OAD output and some p_{t_a} = $[p_{t_a,k}]_{k=0}^{K}$ as its OAA outputs.

4. Experiments

We conducted experiments of the proposed IEN on the two OAD benchmark datasets, THUMOS-14 and TVSeries. First, this section gives an overview of these datasets. Second, we explain the evaluation metrics used to evaluate OAD performance and describe the experimental settings for implementing the proposed IEN. Third, we compared the performances of state-of-the-art methods to our IEN on both the OAD datasets. Finally, we evaluate three versions of LSTM variants to show the efficiency and effectiveness of our IEU through an ablation study.

4.1. Datasets

4.1.1 THUMOS-14

THUMOS-14 [10] is a dataset initially publicized for a competition for offline action detection and localization. THUMOS-14 collected videos from YouTube. This dataset was divided into 20 action classes related to sports such as diving and tennis swing. As the training set of THUMOS-14 consists of well-trimmed videos, its validation set was used for training, and its test set was used for evaluation, as in [5–7, 18]. Specifically, 200 validation videos were used for training and 213 test videos were used for testing in the experiment.

4.1.2 TVSeries

TVSeries [4] is a realistic dataset consisting of 27 episodes from six famous TV series. Each video contains a single episode whose length is approximately 20 minutes or 40 minutes. The 27 videos are divided into 13, 7, and 7 for training, validation, and test set, respectively. A total of 6,231 action instances over 30 classes appear in this dataset. As its videos are collected from TV series, the dataset includes actions with large variability, with the appearance of several actors and actions occurring simultaneously.

4.2. Evaluation Metrics

Following the evaluation protocol in [5–7, 18], we used the per-frame mean average precision (mAP) and per-frame mean calibrated average precision (mcAP) at the frame-level for THUMOS-14 and TVSeries, respectively.

4.3. Experimental Setting

We set the frame rate of all videos at 24 fps. Each chunk consisted of 32 frames and each video segment consisted of eight chunks. We extracted visual features for all chunks within all training and test videos. As a feature extractor, we used RGB-based TSM [11] pre-trained with Kinetics [2] and extract features from the last global average pooling layer. For training, we generate all positive and negative samples corresponding to video segments from all training videos with current frames of actions and background, respectively. Then, we train the proposed IEN with 64 positive and 64 negative samples, which are selected from the generated training samples, in a batch. For testing, we generate all test samples corresponding to video segments from all test videos. For evaluation, we obtain the predicted results on all the test samples with the best IEN model, which is trained through 50 epochs showing the best performance. We set the dimensions of the extracted features (i.e. d_v), embedded features (i.e. d_e), hidden states (i.e. d_h) as 2,048, 512, and 512, respectively. We used Adam as the optimizer, set the learning rate to 0.0001. For the IEU model for both OAD and OAA, we set α to 0.75.

4.4. Performance comparison

In this section, we compare our IEN to existing OAD state-of-the-art methods on the two benchmark datasets, THUMOS-14 [10] and TVSeries [4]. The OAD models are divided into those using RGB features only (noted as *RGB*) and those using two-stream features using RGB as well as optical flow frames (noted as *RGB+Flow*). As a feature extractor for *RGB+Flow* input, RED [7], TRN [18], IDN [5], and TFN [6] use the two-stream (TS) CNN [17] to extract the same TS features using RGB frames for the appearance and optical flows for the motion, as described in TRN [18] in detail. For *RGB* input, RED [7] and TRN [18] use the

same VGG-16 features [15], as described in TRN [18] in detail, and IDN [5] and TFN [6] use only the appearance part of the TS features. For RGB features, our IEN use TSM [11] with only RGB frames. The IENs trained for only AD and trained for both AD and AA, are denoted as $Ours_{AD}$ and $Ours_{AD+AA}$, respectively.

Table 1. OAD Performance comparison on THUMOS-14 [10].

Input	Method	Feature Extractor	mAP (%)
R	TFN [6]	TS-RGB [17]	45.5
	Ours$_{AD+AA}$	TSM-RGB [11]	59.3
	Ours$_{AD}$		**60.0**
R+F	RED [7]	TS [17]	45.3
	TRN [18]		47.2
	IDN [5]		50.0
	TFN [6]		55.7
	IDN-kin. [5]		**60.3**

Table 2. OAD Performance comparison on TVSeries [4].

Input	Method	Feature Extractor	mcAP (%)
R	RED [7]	VGG [15]	71.2
	TRN [18]		75.4
	IDN [5]	TS-RGB [17]	76.6
	TFN [6]		79.0
	Ours$_{AD+AA}$	TSM-RGB [11]	80.9
	Ours$_{AD}$		**81.3**
R+F	RED [7]	TS [17]	79.2
	TRN [18]		83.7
	IDN [5]		84.7
	TFN [6]		85.0
	IDN-kin. [5]		**86.1**

Table 1 and Table 2 report the OAD performances on THUMOS-14 [10] and TVSeries [4], respectivly. Our IEN achieves comparable OAD performances to the-state-of-the-arts using both RGB and flow frames, on both the THUMOS-14 and TVSeries datasets.

Table 3. OAA Performance comparison.

Method	THUMOS-14		TVSeries	
	in 1s	in 2s	in 1s	in 2s
ED [7]	36.8	31.6	74.6	71.0
RED [7]	37.5	32.1	75.5	71.2
TRN [18]	39.1	34.3	75.9	72.3
Ours$_{AD+AA}$	54.2	44.6	77.3	71.9

Table 3 reports both the OAA performances on THUMOS-14 [10] and TVSeries [4]. Our IEN outperforms to the the-state-of-the-arts, on both the two benchmark datasets.

4.5. Ablation studies

4.5.1 Structure for IEU

To demonstrate the importance of our IEU adding a new elevation gate and using x_0 appropriately, we compare the evaluation results from four models using four recurrent units, i.e., original LSTM, LSTM taking additional current information in a naïve way, LSTM taking additionally current information in a sophisticated way, and our IEU. Figure 4 depicts the three recurrent units and our IEU.

Table 4. An ablation study when using different types of information sets.

Model	THUMOS-14	TVSeries
	mAP (%)	mcAP (%)
LSTM$_{w/o-x_0}$	58.5	79.9
LSTM$_{w/-x_0-bundle}$	58.9	80.5
LSTM$_{w/-x_0-sophisticated}$	59.4	80.7
Our IEU	60.0	81.3

As presented in Tab. 4, the performance of LSTM$_{w/o-x_0}$ shows the limitation of not using the current information x_0 for OAD. The LSTM$_{w/o-x_0}$ achieves the worst performance compared to the other three units that take the current information x_0 as input. On THUMOS-14 [10], LSTM$_{w/o-x_0}$ achieves at least 0.4% and at most 1.9% lower performances. On TVSeries [4], LSTM$_{w/o-x_0}$ achieves at least 0.6% and at most 1.5% lower performances. This means that taking the current information x_0 as input is required for temporal modeling for OAD. In addition, the second and third units, LSTM$_{w/-x_0-sohpisticated}$ and LSTM$_{w/-x_0-bundle}$, achieve worse performances than our IEU on both THUMOS-14 [10] and TVSeries [4]. This means that the newly-added information elevation gate, which is not included in LSTM$_{w/-x_0}$s, effectively compensates for the limitation of the forget gate. LSTM$_{w/-x_0-sophisticated}$ outperforms LSTM$_{w/-x_0-bundle}$, which means that assigning inputs fed into each gate in an advanced way by considering each gate is a more effective method of temporal modeling for OAD than using inputs in a bundle.

4.5.2 Relationship between OAD and OAA

Table 5 shows the relationship between OAD an OAA performance according to the balancing parameter α between them. The IEN achieves the best OAD performance when the IEN is trained only for OAD. In contrast, IENs achieves better OAA performances when trained for both OAD and OAA. Especially, the IEN trained only for OAA shows the worst OAA performance between all the IENs except the IEN trained only for OAD.

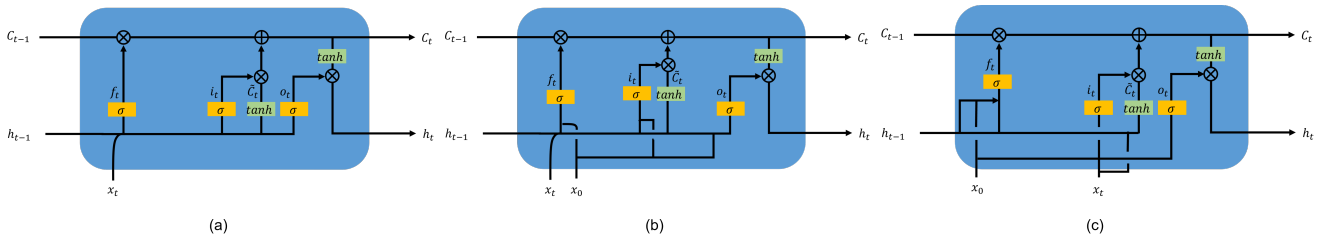

Figure 4. Three compared models for the ablation study. (a) original LSTM w/o-x_0 that does not contain x_0, (a) LSTM w/-x_0 in a naïve way that takes h_{t-1}, x_t, and x_0 in a bundle as input (3) LSTM w/-x_0 in a sophisticated way that uses x_0 instead of x_t or h_{t-1} by considering the role of each gate.

Table 5. Performances according to the balancing parameter α.

$\alpha : (1-\alpha)$	THUMOS-14			TVSeries		
	0s	in 1s	in 2s	0s	in 1s	in 2s
1.00 : 0.00	60.0	2.6	2.1	81.3	50.9	49.9
0.75 : 0.25	59.3	54.2	44.6	80.2	77.9	72.5
0.50 : 0.50	58.4	53.8	46.4	80.2	77.9	72.5
0.25 : 0.75	58.2	53.4	43.9	76.6	77.1	72.0
0.00 : 1.00	2.2	50.5	41.3	51.5	74.7	67.5

4.5.3 Effects of dense prediction

To show the effectiveness of dense prediction of past and current actions, we compare OAD performances between dense prediction of all the chunks and single prediction of the current chunk, as shown in Tab. 6 In all cases, the dense prediction outperforms the single prediction.

Table 6. Performances comparison between dense and single prediction for OAD

α	THUMOS-14			TVSeries		
	dense	single	δ	dense	single	δ
1.00	60.0	59.6	-0.4	81.3	80.5	-0.8
0.75	59.3	59.1	-0.2	80.9	80.4	-0.5
0.50	58.4	57.8	-0.6	80.2	79.3	-0.9
0.25	58.2	57.4	-0.8	76.6	78.8	2.2

4.6. Processing Time for Feature Extraction

Table 7. Processing speed related to feature extraction.

	RGB	TSM [11]	Optical Flow
Speed (fps)	266	1,162	12

For fast OAD, we also extracted visual features using only RGB frames. Table 7 presents the three kinds of speed values required for extracting RGB frames, extracting TSM [11] features, and optical flows. We measured the feature extraction speed in terms of frames per second (FPS), using a Titan XP GPU. Among them, the speed of extracting optical flows is slowest at 12 fps for dense optical flows in OpenCV. In addition, when adopting visual features obtained from optical flows, we require additional time to extract visual features based on optical flows through an AR network, which is not included in Tab. 7. Owing to the excessive time cost for optical flows, adopting them makes OAD methods infeasible for practical services such as video monitoring for specific actions.

4.7. Qualitative Evaluation

As shown in Fig. 5, we visualize qualitative results on THUMOS-14 [10] and TVSeries [4]. The top two parts and bottom two parts of Fig. 5 show the results of the qualitative evaluation on THUMOS-14 and TVSeries, respectively. In particular, the IEN shows remarkable qualitative results on THUMOS-14. On THUMOS-14, the IEN determines the current action with high-predictive action probabilities for the action. Moreover, on THUMOS-14, the predicted probabilities between actions and background are distinguishable, as shown in Fig. 5.

5. Conclusion

In this paper, we propose a novel IEU, which extension of LSTM for OAD, by adding a new elevation gate with an additional input of current information to maintain and aggregate necessary past information related to the current action. Through the dense prediction of past and current actions, the IEN based on the IEUs enhances both the OAD and OAA performances. To the best of our knowledge, our IEN is the first attempt that considers the computational overhead for the practical use of OAD. For practical use of OAD, we adopt a fast action recognition network, TSM [11], for extracting RGB-based visual features by excluding flow frames that require heavy computation overhead. On the OAD benchmark datasets, THUMOS-14 [10] and TVSeries [4], we confirm that our IEN using only RGB frames achieves comparable OAD performances and outstanding OAA performances, compared to the state-of-the-arts using both RGB and flow frames.

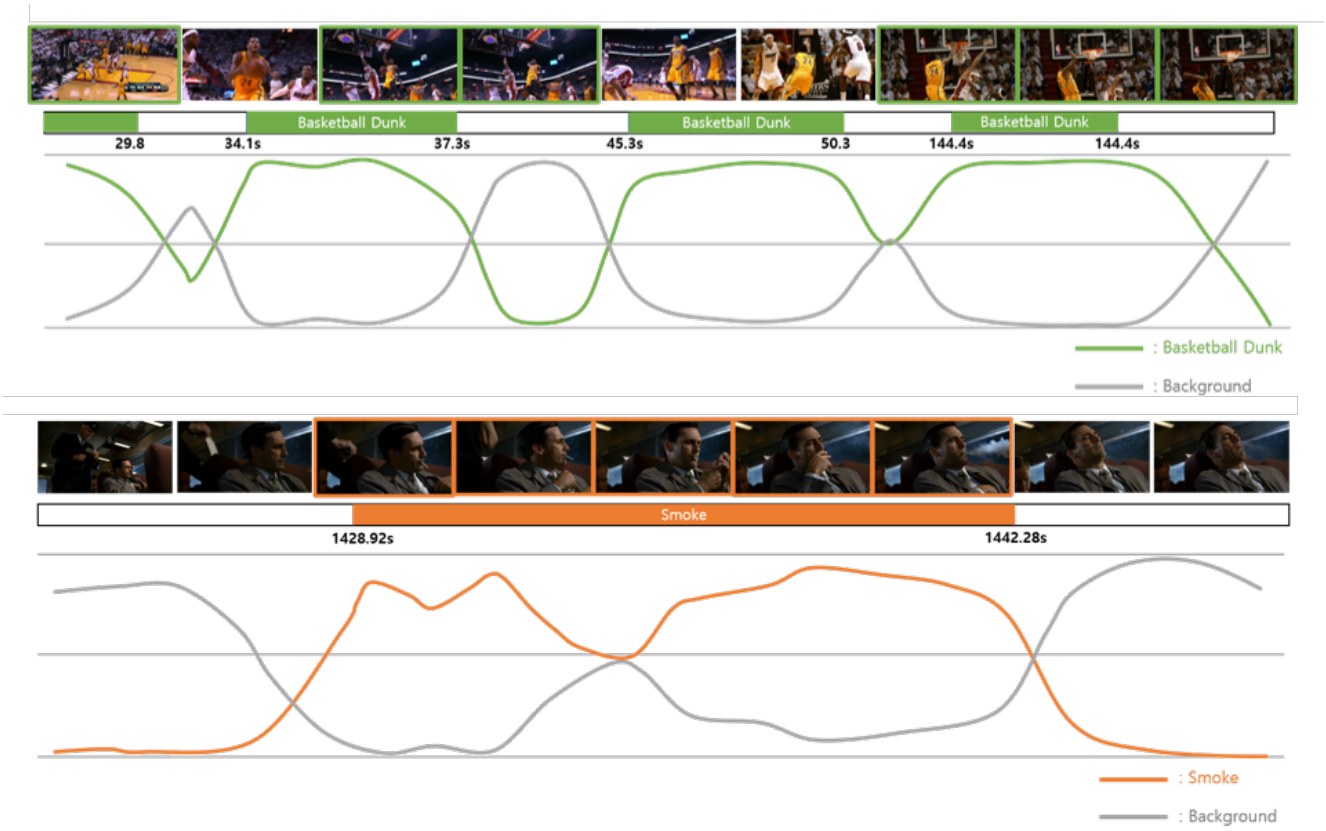

Figure 5. Qualitative evaluation of IEN on THUMOS-14 [10] and TVSeries [4]. The frames painted in color represent detected actions and the below graph represents the predicted action probabilities on their action class.

6. Acknowledgement

This work was supported by Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2020-0-00004, Development of Previsional Intelligence based on Long-term Visual Memory Network.

References

[1] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017. 3

[2] Joao Carreira and Andrew Zisserman. Quo vaids, action recognition? a new model and the kinectics dataset. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6299–6308, 2017. 5

[3] Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. Fast and accurate deep network learning by exponential linear units (elus). In *Proc. International Conference Learning Representations (ICLR)*, 2015. 3

[4] Roeland De Geest, Efstratios Gavves, Amir Ghodrati, Zhenyang Li, Cees Snoek, and Tinne Tuytelaars. Online action detection. In *Proc. European Conference on Computer Vision (ECCV)*, pages 269–284, 2016. 1, 3, 5, 6, 7, 8

[5] Hyunjun Eun, Jinyoung Moon, Jongyoul Park, Chanho Jung, and Changick Kim. Learning to discriminate information for online action detection. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 809–818, 2020. 1, 3, 5, 6

[6] Hyunjun Eun, Jinyoung Moon, Jongyoul Park, Chanho Jung, and Changick Kim. Temporal filtering networks for online action detection. *Pattern Recognition*, 111:107695, 2021. 1, 3, 5, 6

[7] Jiyang Gao, Zhenheng Yang, and Ram Nevatia. Red: Reinforced encoder-decoder networks for action anticipation. In *Proc. British Machine Vision Conference (BMVC)*, 2017. 1, 3, 5, 6

[8] Mingfe Gao, Yingbo Zhou, Ran Xu, Richard Socher, and Caiming Xiong. Woad: Weakly supervised online action detection in untrimmed videos. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1915–1923, 2021. 1, 3

[9] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 1

[10] Yu-Gang Jiang, Jingen Liu, A Roshan Zamir, George Toderici, Ivan Laptev, Mubarak Shah, and Rahul Sukthankar. Thumos challenge: Action recognition with a large num-

ber of classes. http://crcv.ucf.edu/THUMOS14/, 2014. 5, 6, 7, 8

[11] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 7083–7093, 2019. 3, 5, 6, 7

[12] Tianwei Lin, Xu Zhao, Haisheng Su, Chongjing Wang, and Ming Yang. Bsn: Boundary sensitive network for temporal action proposal generation. In *Proc. European Conference on Computer Vision (ECCV)*, pages 3–19, 2018. 3

[13] marketsandmarkets. "video surveillance market by system, offering (hardware (camera, storage device, monitor), software (video analytics, video management system) & service (vsaas)), vertical (commercial, infrastructure, residential), and geography – global forecast to 2025". https://www.marketsandmarkets.com/MarketReports/video-surveillance-market-645.html, 2020. (accessed June 8, 2021). 1

[14] Zheng Shou, Dongang Wang, and Shih-Fu Chang. Temporal action localization in untrimmed videos via multi-stage cnns. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1049–1058, 2016. 2

[15] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 6

[16] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proc. IEEE International Conference on Computer Vision (CVPR)*, pages 4489–4497, 2015. 3

[17] Yuanjun Xiong, Limin Wang, Zhe Wang, Bowen Zhang, Hang Song, Wei Li, Dahua Lin, Yu Qiao, Luc Van Gool, and Xiaoou Tang. Cuhk & ethz & siat submission to activitynet challenge 2016. *arXiv preprint arXiv:1608.00797*, 2016. 5, 6

[18] Mingze Xu, Mingfei Gao, Yi-Ting Chen, Larry S Davis, and David J Crandall. Temporal recurrent networks for online action detection. In *Proc. IEEE International Conference on Computer Vision (ICCV)*, pages 5532–5541, 2019. 1, 3, 5, 6

[19] Can Zhang, Yuexian Zou, Guang Chen, and Lei Gan. Pan: Towards fast action recognition via learning persistence of appearance. *arXiv preprint arXiv:2008.03462*, 2020. 3