IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 26, 2021, accepted May 16, 2021, date of publication May 24, 2021, date of current version June 1, 2021.

*Digital Object Identifier 10.1109/ACCESS.2021.3083273*

# Efficient Spatio-Temporal Modeling Methods for Real-Time Violence Recognition

**MIN-SEOK KANG[1], RAE-HONG PARK[1,2], (Life Senior Member, IEEE), AND HYUNG-MIN PARK[1], (Senior Member, IEEE)**

[1]Department of Electronic Engineering, Sogang University, Seoul 04107, South Korea
[2]ICT Convergence Disaster/Safety Research Institute, Sogang University, Seoul 04107, South Korea

Corresponding author: Hyung-Min Park (hpark@sogang.ac.kr)

This work was supported by the Korea Agency for Infrastructure Technology Advancement (KAIA) Grant through the Ministry of the Interior and Safety under Grant 21PQWO-B153358-03.

**ABSTRACT** Violence recognition is challenging since recognition must be performed on videos acquired by a lot of surveillance cameras at any time or place. It should make reliable detections in real time and inform surveillance personnel promptly when violent crimes take place. Therefore, we focus on efficient violence recognition for real-time and on-device operation, for easy expansion into a surveillance system with numerous cameras. In this paper, we propose a novel violence detection pipeline that can be combined with the conventional 2-dimensional Convolutional Neural Networks (2D CNNs). In particular, frame-grouping is proposed to give the 2D CNNs the ability to learn spatio-temporal representations in videos. It is a simple processing method to average the channels of input frames and group three consecutive channel-averaged frames as an input of the 2D CNNs. Furthermore, we present spatial and temporal attention modules that are lightweight but consistently improve the performance of violence recognition. The spatial attention module named Motion Saliency Map (MSM) can capture salient regions of feature maps derived from the motion boundaries using the difference between consecutive frames. The temporal attention module called Temporal Squeeze-and-Excitation (T-SE) block can inherently highlight the time periods that are correlated with a target event. Our proposed pipeline brings significant performance improvements compared to the 2D CNNs followed by the Long Short-Term Memory (LSTM) and much less computational complexity than existing 3D-CNN-based methods. In particular, MobileNetV3 and EfficientNet-B0 with our proposed modules achieved state-of-the-art performance on six different violence datasets. Our codes are available at https://github.com/ahstarwab/Violence_Detection.

**INDEX TERMS** Real-time violence detection, efficient spatio-temporal attention, efficient convolution method for spatio-temporal modeling.

## I. INTRODUCTION

Recently, reliable automatic surveillance systems attract much interest where occurrences of crime situations take place occasionally at any time. In particular, for violence recognition, it is essential to detect violent action in real time so that the police can be dispatched promptly during crimes. Although video-based action recognition has achieved impressive improvement over the last few years, most of the works have focused on performance, not efficiency.

2D Convolutional Neural Networks (CNNs) have shown remarkable results on image recognition tasks by performing a cross-correlation on a single multi-channel image with convolution kernels. This approach, however, has a limitation in being applied to video understanding since a video is a temporal sequence of frames and 2D CNNs cannot encode dynamic motion information. Simple approaches to produce spatio-temporal video representation have applied Recurrent Neural Networks (RNNs) to the output of the CNN layers [1]–[4]. Nevertheless, those approaches have shown shortcomings in performance since they did not perform convolution across multiple frames in early layers of the networks. Since the conventional 2D CNN models did not encode temporal information, video-based action recognition was a challenging task.

C3D [5] and I3D [6] introduced 3D CNNs by expanding 2D filters to the time axis to encode spatio-temporal information. They were successfully applied to video action

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

recognition tasks by capturing both spatial and temporal information in videos with 3D filters. Although the 3D CNN models showed substantial improvement in performance, they were computationally expensive and needed a large number of parameters.

Instead of using only RGB frames as input, two-stream networks [7] and many other frameworks based on 2D or 3D CNNs [6], [8], [9] tried to combine RGB frames with optical flow features as input. They achieved impressive improvement on video action recognition. Since the optical flow represents a field of dense motion vectors, it has a great impact on performance for video action recognition. However, optical flow algorithms usually require too expensive computational costs to perform real-time action recognition in videos. Some of the works conducted human pose estimation for action recognition [10]–[12] to concentrate on human actions. Skeleton Points Interaction Learning (SPIL) module [12] computed the interaction weights to model interactions between extracted human skeleton points for violence recognition. However, these approaches required an additional cost for extracting skeleton information and filtered out too much information from raw data. Furthermore, wrong estimation of the skeleton features can cause performance degradation.

We focused on violence detection in real-world situations by embedding our algorithm on a camera module, for easy expansion into a surveillance system with a huge number of cameras. Also, our algorithm will notice surveillance personnel at the moment violent crimes take place, so it should be operated in real time. Therefore, we design an efficient violent action recognition system for real-time and on-device surveillance. Three properties required for a real-time and on-device deep neural network system are (1) low computational cost, (2) sufficiently compact model size, and (3) sufficiently high accuracy. Our proposed network is developed by considering these properties.

Model compression is an active area of compressing the model with minimal performance degradation. Related techniques include parameter pruning [13]–[17], low-rank approximation [18]–[21], quantization [22]–[25], knowledge distillation [26]–[29], and use of compact models [30]–[33]. MobileNets [31]–[33] leveraged Depthwise Separable Convolution to reduce the model size by decomposing the typical 2D convolution into depth-wise and point-wise convolution. Compact CNN [30] further compressed the channels of typical CNN layers of 2D CNN backbones to reduce the model size. On the other hand, without changing any existing CNN architectures, we propose a method that can model the spatio-temporal information.

Humans usually focus on actors compared to backgrounds for recognizing violent situations since violent crimes can happen anywhere but are committed by humans. Spatio-temporal saliency detection methods were proposed to amplify relevant regions and reduce irrelevant backgrounds in dynamical scenes. Guo and Zhang generated a saliency map by applying the Fourier transform to a quaternion image that is a weighted sum of color, intensity, and also the motion feature which is the difference between intensities of the corresponding pixels in consecutive frames [34]. Instead of using original RGB channels, they used red/green and blue/yellow channels inspired by a human visual cortex. Kim *et al.* computed the motion features with edge and color orientation histograms to generate a spatio-temporal saliency map [35]. Nasaruddin *et al.* proposed background (BG) subtraction method to blur uninteresting areas in the surveillance video by using the binary bitmaps of each frame [36].

Some works utilized spatial and temporal attention modules in video action recognition to reduce redundant information over space and time [37]–[41]. Since inferring the attention was not a trivial task, attention modules usually required a large memory consumption and computational complexity. Pre-trained object detectors were used to assist their spatial attention module [40]. Moreover, two loss terms for the spatial attention and one loss term for the temporal attention were introduced [41]. In this work, we focus on utilizing lightweight and fast attention modules for real-time applications. First, we generate an *easy-to-train* attention map by combining the knowledge of a motion feature and a morphological dilation for the spatial attention. Then, we introduce a simple architecture to recalibrate time-wise features for the temporal attention.

Automated video surveillance can bring public safety to society while it significantly mitigates an exhausting process for surveillance personnel to detect crime occurrences. Some works integrated facial recognition, object tracking, and crime prediction into their surveillance systems [42]–[50]. Those works, however, could be controversial due to privacy infringements and machine bias (fairness). In this work, we focused on supporting surveillance personnel to cope with violent crimes immediately, with minimizing the risk of invasion of privacy (without any object tracking or facial recognition) and intrusion of bias (crime detection instead of prediction).

Our violence detection pipeline consist of three steps and the description of each step is as follows:

1) Based on the investigation that people in violent situations usually move actively to produce strong pixel differences between consecutive frames than others (e.g. bystanders and backgrounds), we propose an efficient spatial attention module, which is inspired by conventional methods in image processing, such as RGB difference and morphological dilation. The RGB difference between consecutive frames captured by a fixed camera represents the boundaries of moving objects. From the calculated boundaries, we apply a few average pooling layers and CNN layers to generate spatial attention maps.
2) Since violent actions such as punching and kicking usually last for a short time, we propose a method to give 2D CNNs the ability to encode short-term motion information. We replace the channels with time frames, by averaging the RGB channels and

grouping three consecutive channel-averaged frames as an input of the 2D CNN to model short-term dynamics, which is a critical factor to detect violent actions such as punching, pushing, and kicking. We explore some suitable architectures for mobile devices such as SqueezeNet, MobileNets, and EfficientNet with the proposed method and achieve remarkable improvements in terms of the accuracy while significantly decreasing the memory footprint and the computational complexity for video recognition.

3) We propose a temporal attention module that is very similar to Squeeze-and-Excitation (SE) blocks [51]. The SE block recalibrates channel-wise features, while our temporal attention module deals with time-wise features. First, global spatial information is squeezed by using a global average pooling as the SE block does. Then, fully-connected layers and an average pooling across the channel axis are applied to the calculated time-wise tensors to generate adaptive temporal weights. Multiplying the weights with the temporal tensors achieves temporal attention, resulting in performance improvement.

## II. RELATED WORKS

### A. REPLACEMENTS OF 3D CNN FOR SPATIO-TEMPORAL MODELING

Over the past few years, there have been many studies on spatio-temporal modeling for video recognition. 3D CNN was introduced to tackle the problem that 2D CNN cannot encode temporal information. It has successfully applied to the tasks related to video understanding, but it is computationally expensive and requires much more parameters than the 2D CNN. There are many approaches to mitigate the inefficiency issues on the 3D CNN. R(2+1)D [52] and P3D [53] replaced a standard 3D filter by a 2D filter for modeling spatial information and a 1D filter for modeling temporal information. Those approaches have made the 3D CNN lighter and shown comparable performance with the 3D CNN. A temporal shift module (TSM) was introduced [54], shifting a small portion of the channels in feature maps along the time axis to capture temporal information. TSM was applied in the forward path of residual blocks of 2D CNN models. With the same FLOPS and parameters with the 2D CNN, it achieved comparable performance with the 3D CNN. In this work, we introduce a spatio-temporal modeling strategy called frame-grouping, which can further reduce the memory footprint and computational load by averaging the channel of inputs, inspired by the observation that the channel is relatively subordinate among time, channel, height, and width.

### B. SPATIO-TEMPORAL ATTENTION FOR REDUCING REDUNDANT INFORMATION

In a complex video, reducing distracting information over time and space is an essential yet challenging task for video action recognition. Many works have already used spatial attention modules for video analysis, but finding salient regions without any clues is not a trivial task. To mitigate this problem, pre-trained object detectors [40] and some regularizers [41] were used to encourage the spatial attention. Persistence of Appearance (PA) was introduced that was a motion representation calculated by the Euclidean distance between two consecutive CNN features since the boundaries of moving objects are important for distinguishing human actions [55]. On the other hand, our work focuses on computing spatial attention maps derived from the boundaries of moving objects that are further multiplied with original frames.

Attention mechanism has been utilized in a variety of fields including video action recognition [56]–[58]. Video Transformer Network (VTN) was introduced for real-time video action recognition, where Transformer [59] was combined with 2D CNNs for long-temporal modeling. Non-Local (NL) block [56] is a well-known module for video action recognition that can model both spatial and temporal self-attentions jointly. Since the NL block tended to capture appearance similarities, temporal modeling was improved by decoupling temporal self-attention from spatial one and applying Global Temporal Attention (GTA) to learn temporal relationships after spatial self-attention [57]. SE block is another attention module to capture inter-dependencies along the channel dimension. Our temporal attention module is inspired by the fact that attention weights of the SE block can be simply calculated with some fully connected layers. We apply this module to calculate temporal weights by replacing the channel with the time.

### C. RECENT VIOLENCE DETECTION METHODS

There have been various violence detection methods with 2D CNN + Long Short-Term Memory (LSTM) and 3D-CNN-based models [60]–[64]. Sudhakaran and Lanz used frame differences instead of RGB frames as an input of the 2D CNN + LSTM to generate a better representation of changes between adjacent frames [65]. Cheng *et al.* introduced a two-stream network to encode raw frames and optical flow features in each stream that consist of 3D CNN [9]. Contrary to most of the works using the LSTM or 3D CNN for temporal modeling, Jain and Vishwakarma aggregated multiple frames called Dynamic Image (DI) to represent motion features in a single frame and fed it into 2D CNN [66]. Background subtraction was also used in violence detection to reduce the influence of disturbing backgrounds [36]. Su *et al.* used graph convolution for violence detection with an assistance of a pose estimation model [12]. Wu *et al.* introduced a model called HL-Net to model relations among frames, and they used audio in addition to visual information as an input to improve the performance [67].

## III. PROPOSED APPROACH

Our proposed violence recognition system consists of three parts: Motion Saliency Map (MSM) module, 2D CNNs with frame-grouping, and Temporal Squeeze-and-Excitation

(T-SE) block. The proposed modules (MSM and T-SE block) are lightweight for on-device real-time violence recognition but consistently improve detection performance. In particular, frame-grouping brings a significant performance improvement while decreasing computational cost by a third. In this section, we explain the intuitions and details of our proposed modules.

### A. INTUITIONS

#### 1) INTUITION OF MSM

In violence recognition, background information is not an important factor. To avoid modeling irrelevant features from complex scenes, we tried to focus on human actions by generating attention maps. In a video captured by a fixed camera like a closed-circuit television (CCTV), motion features represent the boundaries of moving objects. We focus on generating attention maps that are derived from motion features calculated from the Euclidean distance between two consecutive frames. As the calculated features are too sharp to be used as attention maps, we add several operations to dilate the obtained boundaries to achieve our intended goal. The module is effective with a fixed camera, since it has to capture the motion boundaries of moving objects between two consecutive frames. However, the next two modules are irrespective of whether the camera is fixed or not.

#### 2) INTUITION OF FRAME-GROUPING

The short-term dynamics is a key factor to recognize violent situations since violent actions such as punching, pushing, and kicking take a relatively short time compared to other events. We tried to model short-term motions in an efficient way. RGB color images are represented by the intensity of red, green, and blue. Fig. 1 shows every channels for three RGB images, plotted with *viridis* colormap. Although color information is useful for many tasks such as image recognition and action recognition, we squeeze the channel dimension to reduce the computational load, assuming that motion information is more important than the richness of color information to detect violent actions. We calculate the mean value of the channel dimension and replace the channel with the time to employ 2D CNN instead of 3D CNN for modeling short-term dynamics efficiently.

#### 3) INTUITION OF T-SE BLOCK

Since certain actions only happen within some time periods of video clips, we introduce an efficient temporal attention module to recalibrate temporal features adaptively. With a small number of additional parameters, the module calculates temporal attention weights for highlighting the time periods that are more correlated to target events.

We constructed violence detection system with frame-grouping as an essential module, while MSM or T-SE block can be excluded depending on memory capacity and computational speed of a hardware. In the next three subsections, we will describe the details of our three proposed modules. Fig. 2 illustrates the overall procedure of our proposed violence recognition system.

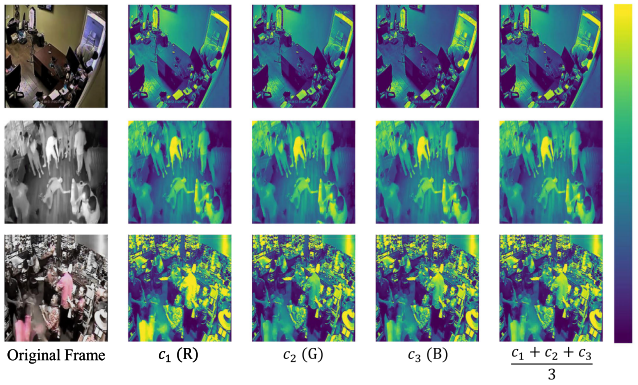



**FIGURE 1. Representations of every channels for three RGB images from the RWF-2000 dataset, plotted with *viridis* colormap. From the left column, RGB color, red, green, blue, and averaged images are displayed.**

### B. MSM

To amplify salient regions related to violence in videos, we propose MSM that can efficiently highlight moving objects. Inspired by the fact that calculated motion features can represent the boundaries of moving objects, we generate attention maps by dilating the motion boundaries. The overview and examples of our MSM module are illustrated in Figs. 3 and 4, respectively. We first calculate the Euclidean distance between two consecutive RGB frames $\mathbf{X}_t, \mathbf{X}_{t+1} \in \mathbb{R}^{3\times H\times W}$ for $1 \leq t \leq T$ and sum up along the channel axis to obtain the boundaries of moving objects $\mathbf{b}_t \in \mathbb{R}^{H\times W}$ as:

$$\mathbf{b}_t = \sqrt{\sum_{i=1}^{3} (\mathbf{X}_{t+1}^i - \mathbf{X}_t^i)^2}, \tag{1}$$

where $i$ denotes the channel index and $T$ is the total number of input frames. The last step to generate a spatial attention map $\mathbf{m}_t \in \mathbb{R}^{H\times W}$ is to apply pooling layers and convolutional layers to dilate the boundaries as:

$$\mathbf{m}_t = \sigma(conv(pool(\mathbf{b}_t))), \tag{2}$$

where $\sigma$ denotes a sigmoid function, and *conv* and *pool* represent convolutional layers with ReLU activations and average pooling layers, respectively. The detailed parameters in *pool* and *conv* are summarized in Table 1. Without any smoothing operations such as pooling or convolution, the attention map has too sharp and complicated shapes. Since our works focus on finding salient areas derived from the boundaries of moving objects, we apply four convolutional layers and two pooling layers for enlarging the boundaries to find salient areas.

The last step of the MSM module is element-wise multiplication of $\mathbf{m}_t$ and the second frame $\mathbf{X}_{t+1}$, expressed as:

$$\mathbf{Y}_t = \mathbf{X}_{t+1} \odot \mathbf{m}_t, \tag{3}$$

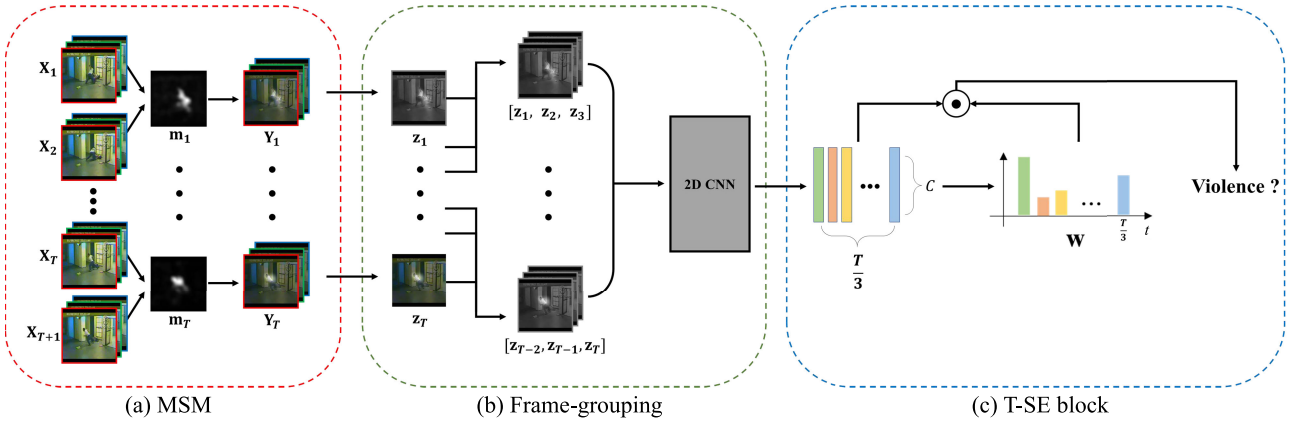



**FIGURE 2.** Overall procedure of our proposed model. (a) $\mathbf{X}_t$, $\mathbf{m}_t$, and $\mathbf{Y}_t$ are an input image, the corresponding spatial attention map, and the attended image feature at frame $t$, respectively. (b) $\mathbf{z}_t$ is a channel-averaged image at frame $t$, and we experimented SqueezeNet, MobileNets, and EfficientNet for 2D CNN. (c) $C$, $T$, and $\mathbf{w}$ are the number of channels, the number of frames, and the calculated temporal weights, respectively.

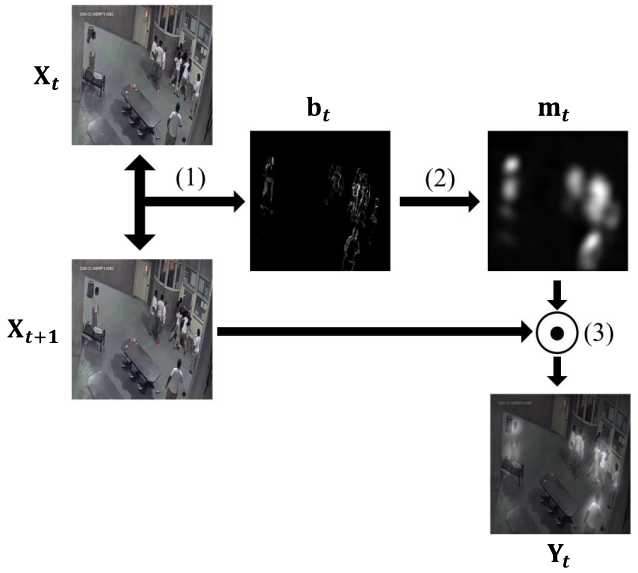



**FIGURE 3.** Illustration of the MSM module. $\mathbf{X}_t$, $\mathbf{b}_t$, $\mathbf{m}_t$, and $\mathbf{Y}_t$ are an input image, the calculated motion boundaries of two consecutive images, the corresponding attention map, and the attended image feature at frame $t$, respectively. $\odot$ denotes the element-wise multiplication. $\mathbf{Y}_t$ is displayed by overlapping $\mathbf{X}_{t+1}$ and $\mathbf{m}_t$ to clearly indicate attended regions.

where $\odot$ denotes the Hadamard Product. As shown in Fig. 4, the attended image feature $\mathbf{Y}_t \in \mathbb{R}^{3\times H\times W}$ successfully highlights attended regions corresponding to the motion of moving objects. $\mathbf{Y}_t$ will be sent to a 2D CNN backbone in the next subsection (frame-grouping).

### C. FRAME-GROUPING

2D convolution performs cross-correlation on a single multi-channel image by applying 2D kernels to each channel and summing the results across the channel axis. Since it only encodes individual frames, it is incapable of modeling spatio-temporal information from videos. 3D convolution, on the other hand, performs cross-correlation on multiple multi-channel images with 3D-expanded kernels striding along the spatial and temporal axes to encode spatio-temporal information. It requires more parameters and FLOPS compared to 2D convolution since the expanded kernels stride in time as well as space. In this work, we rather use lightweight 2D CNN backbones instead of using heavy 3D CNN models to deal with efficient spatio-temporal modeling. We make each three-channel frame into a single-channel frame in the middle of the forward path and group three consecutive frames to learn spatio-temporal representation in a video with the 2D CNN backbones. We call this method frame-grouping. In this work, we just average the channels instead of grayscale conversion (linear combination of the channels with weights of $[w_R = 0.30, w_G = 0.59, w_B = 0.11]$) since each channel of input frame $\mathbf{X}_t$ is already normalized with specific mean and standard deviation values and the main purpose of frame-grouping is a fast modeling of short-term dynamics rather than colorful representation to capture spatio-temporal information efficiently.

A toy example to process nine RGB frames is illustrated in Fig. 5. The first step of the frame-grouping is to average the channels of input images to obtain single-channel images $\mathbf{z}_t \in \mathbb{R}^{H\times W}$ as:

$$\mathbf{z}_t = \frac{1}{3}\sum_{c=1}^{3}\mathbf{Y}_t^c, \tag{4}$$

**TABLE 1.** Parameters of the MSM module.

| Layer | Kernel width | Stride | #Output Channels |
|---|---|---|---|
| $pool_1$ | 15 | 1 | 1 |
| $pool_2$ | 15 | 1 | 1 |
| $conv_1$ | 7 | 1 | 3 |
| $conv_2$ | 7 | 1 | 8 |
| $conv_3$ | 7 | 1 | 3 |
| $conv_4$ | 7 | 1 | 1 |

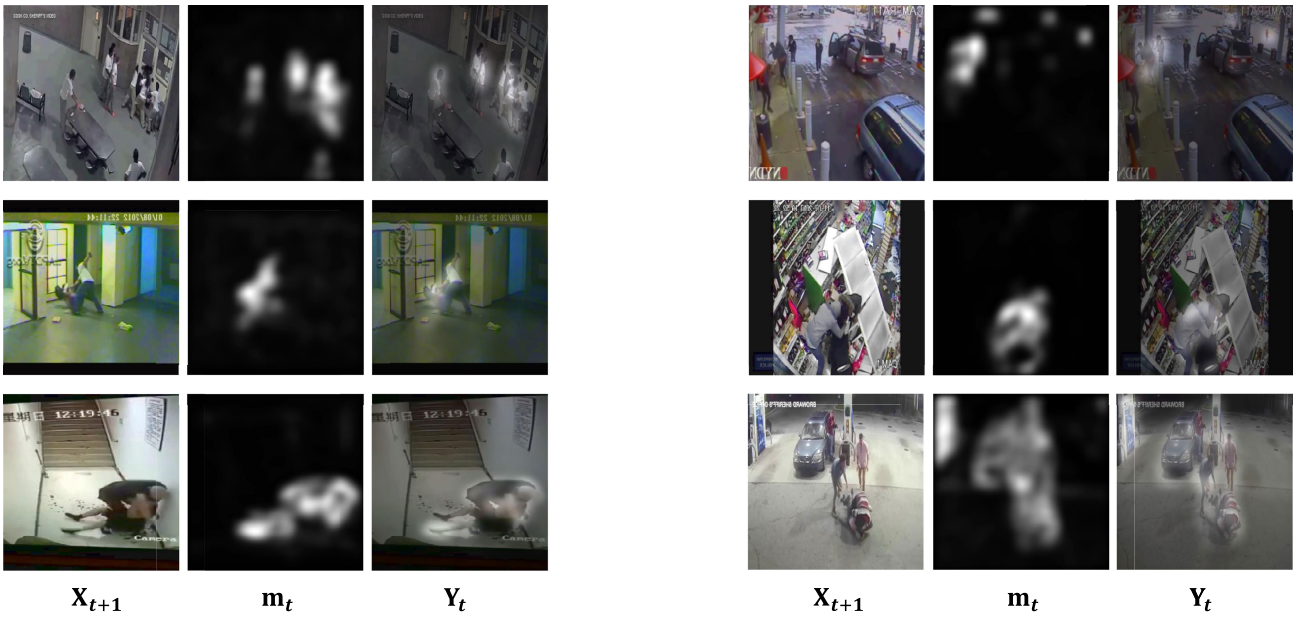



**FIGURE 4.** Results of the MSM module on the RWF-2000 dataset. $\mathbf{X}_{t+1}$, $\mathbf{m}_t$, and $\mathbf{Y}_t$ are an input image, the corresponding attention map, and the attended image feature at frame $t$, respectively. $\mathbf{Y}_t$ is displayed by overlapping $\mathbf{X}_{t+1}$ and $\mathbf{m}_t$ to clearly indicate attended regions.

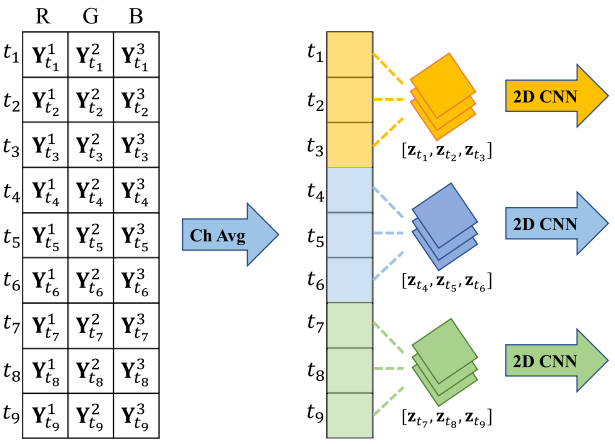



**FIGURE 5.** A toy example of frame-grouping. $t$, $\mathbf{Y}_t$, and $\mathbf{z}_t$ are the time index, the attended image, and the channel-averaged image, respectively.

where the subscript $t$ and the superscript $c$ of $\mathbf{Y}$ represent the time and channel indices, respectively. After averaging the channel, we group three consecutive frames as an input to 2D CNN to map single-channel images $\mathbf{Z} = [\mathbf{z}_1, \mathbf{z}_2, \ldots \mathbf{z}_T]$ to feature maps $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{\frac{T}{3}}]$, which is expressed as:

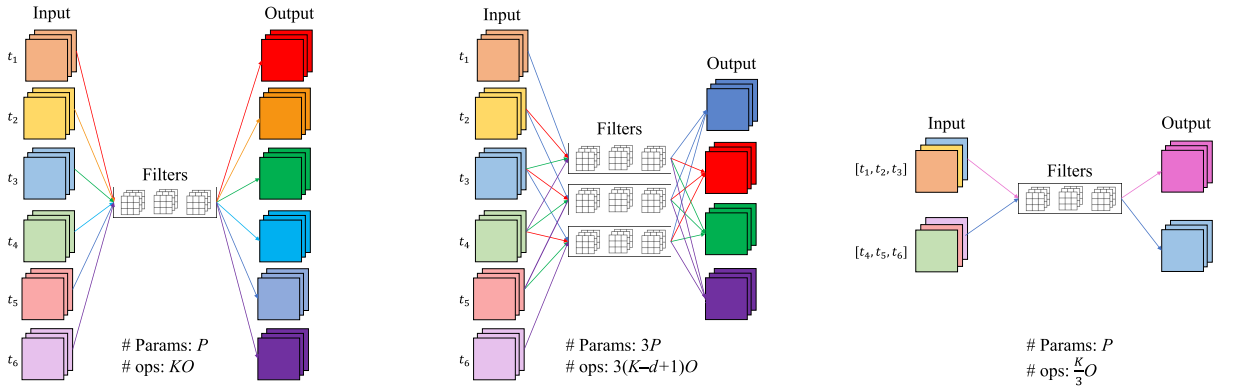
$$\mathbf{u}_n^c = \sum_{t=3n-2}^{3n} \mathbf{v}_{(t \bmod 3)}^c * \mathbf{z}_t, \tag{5}$$

where $\mathbf{V^c} = [\mathbf{v}_0^c, \mathbf{v}_1^c, \mathbf{v}_2^c]$ is a kernel of the first layer of 2D CNN models, $n$ is the temporal index in the feature maps $\mathbf{U}$ with $1 \leq n \leq \frac{T}{3}$, and $T$ is the total number of input frames that should be divisible by 3 since we replace a single three-channel image to three channel-averaged images for an input of 2D CNN. $c$ is the output channel index of the first CNN layer. $*$ denotes the convolution operator. The conventional 2D convolution takes a single three-channel frame as an input, while the 2D convolution with frame-grouping takes three consecutive single-channel frames. Since our proposed module combines three temporally consecutive frames, it inherently encodes temporal information like the conventional 3D convolution.

Fig. 6 briefly shows an example to compare our method (2D convolution with frame-grouping) with the conventional 2D and 3D convolutions. Assume that we have kernels of length $d$ with stride 1 in time for the 3D convolution and $K$ RGB frames to process. Let us denote the numbers of parameters and operations per frame by $P$ and $O$, respectively, for the 2D convolution, where the numbers of channels in inputs and outputs are 3. To process $K$ RGB frames, the 2D convolution needs $P$ parameters and $KO$ operations while the 3D convolution requires $3P$ parameters and $3(K - d + 1)O$ operations. On the other hand, our proposed method needs $P$ parameters and only consumes $\frac{K}{3}O$ operations. Furthermore, as the time steps after the backbone network are reduced by a third compared to the conventional 2D convolution, the size of tensors in the middle of the system is scaled down. Therefore, the memory demand and the number of parameters of the next layers are decreased. We leverage this module to achieve an on-device real-time violence recognition system.

### D. TEMPORAL SQUEEZE-AND-EXCITATION BLOCK

Temporal Squeeze-and-Excitation (T-SE) block is a time-wise gating module that can inherently recalibrate and




(a) 2D convolution (b) 3D convolution (c) 2D convolution with frame-grouping

**FIGURE 6.** Comparison of (a) 2D convolution, (b) 3D convolution, and (c) 2D convolution with frame-grouping for processing six RGB images to generate three-channel CNN feature maps. In the figure, we have 3 × 3 × 3 ($d = 3$) convolutional filters with stride 1 in time for 3D convolution and 3 × 3 convolutional filters for the others. $t$ is the time index, and $P$, $K$, $O$, and $d$ are the number of parameters, the number of frames, the number of operations, and the length of kernels, respectively.

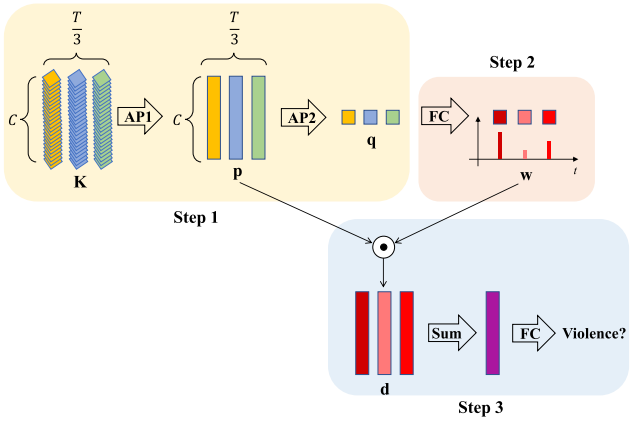



**FIGURE 7.** Illustration of Temporal Squeeze-and-Excitation (T-SE) blocks. $C$ denotes the number of channels of intermediate tensors in a 2D CNN backbone, and ⊙ denotes the element-wise multiplication. **AP1** is a global average pooling over the spatial dimension and **AP2** is an average pooling over the channel dimension.

aggregate the global temporal information to classify a target event with a less computational cost. The difference between SE block [51] and T-SE block is the purpose of two modules. The SE block aims to capture channel-wise dependencies, while our T-SE block recalibrates temporal features for enhancing temporal regions relevant to target events. The T-SE block can be split into three steps, and Fig. 7 illustrates every steps with the same notations as the following equations.

**Step 1: Squeezing on the spatial and channel dimensions.** To generate temporal weights, we first squeeze the global spatial information as:

$$\mathbf{p}^c = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{K}^c(i, j), \tag{6}$$

where $\mathbf{K} \in \mathbb{R}^{\frac{T}{3} \times C \times H \times W}$ and $C$ are the output and the number of output channels at an intermediate layer of CNN backbones, respectively, and $\mathbf{p}^c$ is a 1D vector where $\mathbf{p}^c = [p_1^c, p_2^c, \ldots, p_{\frac{T}{3}}^c]$. Then, we squeeze the channel data to generate temporal statistics as:

$$\mathbf{q} = \frac{1}{C} \sum_{c=1}^{C} \mathbf{p}^c, \tag{7}$$

where $\mathbf{q} = [q_1, q_2, \ldots, q_{\frac{T}{3}}]$.

**Step 2: Extraction of temporal weights.** We apply two fully connected layers followed by a sigmoid function to obtain temporal weights:

$$\mathbf{w} = \sigma(g_1(\mathbf{q})), \tag{8}$$

where $\mathbf{w} = [w_1, w_2, \ldots, w_{\frac{T}{3}}]$ is a temporal weight vector and $g_1$ represents two fully connected layers that shrink the number of time steps in half at the first layer and restore the shape at the next.

**Step 3: Adaptive Recalibration on the time dimension (Excitation).** By multiplying $\mathbf{p}^c$ with $\mathbf{w}$, the temporal features are recalibrated so that more relevant time intervals with target events are highlighted than others:

$$\mathbf{d}^c = \mathbf{p}^c \odot \mathbf{w}. \tag{9}$$

Finally, $\mathbf{d}^c$ is further summed up along the time axis and fed into a single fully-connected layer for the final video action classification:

$$score = g_2(\sum_{t=1}^{\frac{T}{3}} \mathbf{d}_t^c), \tag{10}$$

where $g_2$ denotes a single fully connected layer along the channel axis.

## IV. EXPERIMENTS

We have evaluated our methods on six violence datasets: Hockey Fight [68], Movie Fight (Peliculas) [69], Crowd (Violent Flow) [70], Surv (Surveillance Camera Fight) [63],

**TABLE 2. Brief description of violence datasets we evaluated.**

| Dataset name | # of Clips | Durations |
|---|---|---|
| Hockey [68] | 1000 | 1s |
| Movie [69] | 200 | 1s |
| Crowd [70] | 246 | 3∼5s |
| Surv [63] | 300 | 1∼3s |
| RLVS [62] | 2000 | 5s |
| RWF-2000 [9] | 2000 | 5s |

Real Life Violence Situations (RLVS) [62], and RWF-2000 [9]. A brief description of the violence datasets is summarized in Table 2. In this work, we mainly targeted on the RWF-2000 dataset, since it is not only the largest violence dataset but also well splitted to train/test partitions. We assumed the system with MSM module to be applied in a fixed camera, but the datasets we explored contain many videos captured with moving cameras. The investigation regarding effectiveness of MSM with a moving camera is described in Appendix A.

First, we described the implementation details of our models. Second, we experimented our modules with CNN backbones such as SqueezeNet, MobileNets, and EfficientNet that are pre-trained on ImageNet [71]. We also compared our modules with 3D CNNs pretrained on Kinetics [72]. Third, we compared the results of our modules with other violence recognition methods. Fourth, by performing ablation studies, we demonstrated that our proposed modules can efficiently benefit the existing 2D CNN models. Fifth, by calculating FLOPS and the number of parameters of our modules, we showed that the SqueezeNet, MobileNets, and EfficientNet with our proposed modules can achieve real-time and on-device violence recognition. Finally, we demonstrated the results with long-term violence videos called UCF-Crime [73].

### A. IMPLEMENTATION DETAILS

Implementation of our networks was based on PyTorch. We resized input images to have a fixed size of 224 × 224. We trained our models using the Adam optimizer with a learning rate of 0.001 and a batch size of 16 on a Nvidia-RTX Titan throughout the experiments. For violence recognition on each video, we fixed $T = 30$ to set the number of input frames as a multiple of 3, and selected frames at a uniform intervals. Since the length of videos from different datasets are different, time intervals between frames varies. The investigation of an adequate time interval for modeling spatio-temporal information effectively is described in Appendix B. We selected SqueezeNet 1.1, MobileNetV2 with a multiplier of 1.0, MobileNetV3-Large, and EfficientNet-B0 for CNN backbones. We also conducted color jittering and random horizontal flips to prevent over-fitting of models.

### B. EXPLORATION OF CNN BACKBONES

We evaluated several models on the RWF-2000 dataset and summarized the results in Table 3. In this paper, we combined lightweight 2D CNN backbones such as SqueezeNet, MobileNets, and EfficientNet with our proposed modules to construct an efficient violence detector. The brief information of the 2D CNNs we explored is described in Appendix C. We also conducted an experiment with Inflated 3D ConvNet (I3D). The concept of I3D is to inflate 2D filters and pooling kernels into 3D on existing 2D CNN backbones for extracting spatio-temporal features. The I3D was pretrained with the large video dataset named Kinetics while the 2D CNNs were pretrained with the large image dataset named ImageNet. For fair comparison of both the networks, we used only RGB frames as inputs although the I3D could be improved by including an optical flow stream as well as the RGB stream. Even with smaller parameters and faster training and inferencing speeds than the I3D, the 2D CNN backbones with our modules showed comparable performances on the dataset. We also found that the I3D could be improved by MSM. We did not report the I3D with the T-SE block since the number of time steps is shrunk after 3D convolutional layers in the model.

We visualized some output features for the first CNN layer of MobileNetV3 with frame-grouping and I3D in Fig. 8. It demonstrated that our proposed 2D CNN with frame-grouping could process meaningful features such as horizontal edges, vertical edges, background, and foreground from a sequence of frames as 3D CNN like I3D could. Additionally, we found that our violence detection pipeline could inherently extract salient regions while attenuating backgrounds by the first CNN layer. Salient regions with attenuated backgrounds could be emphasized by applying MSM as shown in Fig. 9. We also investigated the interpretability of our model by visualizing Grad-CAM [74] in Appendix D.

### C. COMPARISON WITH OTHER METHODS ON VIOLENCE DATASETS

In Table 4, we show recognition accuracies evaluated on six violence video datasets: Hockey [68], Movie [69], Crowd [70], Surv [63], RLVS [62], and RWF-2000 [9]. We got accuracies from authors' reports for six methods (above the double horizontal line in Table 4) while our method was evaluated by 5-fold cross-validation for all datasets except for the RWF-2000 dataset that is explicitly divided into train/test sets. **Inception-Resnet-V2 + DI** [66] introduced Dynamic Image (DI) that is a weighted aggregate of consecutive frames to summarize objects' movements into a single frame. It used a pre-trained Inception-Resnet-V2 for processing a single DI to classify violent events. **VGG-16 + LSTM** [62] used the VGG-16, pretrained on the ImageNet dataset for spatial modeling followed by a LSTM for temporal modeling. **Xception + Bi-LSTM** [63] used the Xception for spatial modeling and a bidirectional LSTM with attention modules [75] for temporal modeling. **Flow Gated Network** [9] used optical flow features with RGB images as inputs and utilized two-stream network for spatio-temporal modeling to classify violence. **SPIL** [12] computed the

**TABLE 3.** Comparison of four models on the RWF-2000 dataset. The number of parameters (#Params) and FLOPS were reported for the full models. The proposed modules are represented in bold, and the best performance and the smallest #Params and FLOPS are highlighted in bold.

| Model | CNN type | Accuracy (%) | #Params (M) | FLOPS (G) |
|---|---|---|---|---|
| I3D | 3D | 88.8 | 12.3 | 55.7 |
| **MSM + I3D** | | 89.3 | 12.3 | 61.9 |
| **MSM + SqueezeNet 1.1 with frame-grouping + T-SE block** | 2D | 89.0 | **1.2** | 7.5 |
| **MSM + MobileNetV2 with frame-grouping + T-SE block** | | 89.3 | 2.2 | 7.1 |
| **MSM + MobileNetV3 with frame-grouping + T-SE block** | | **90.0** | 2.9 | 6.2 |
| **MSM + EfficientNet-B0 with frame-grouping + T-SE block** | | **92.0** | 1.3 | **4.2** |

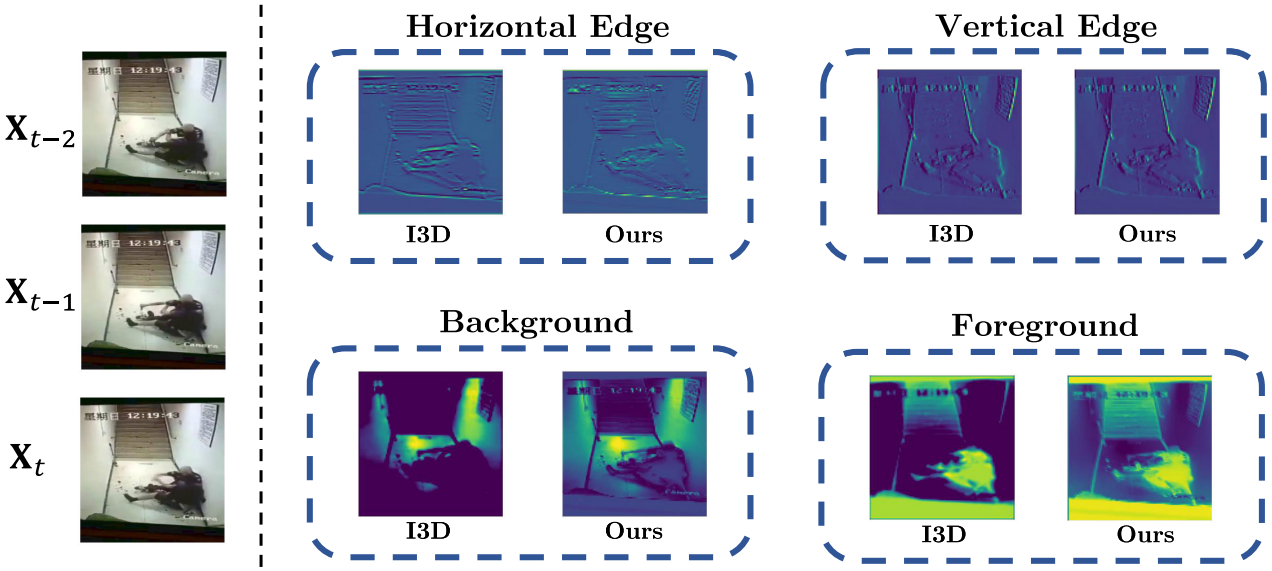



**FIGURE 8.** Examples of output features for the first CNN layer of I3D with 3D CNN and MobileNetV3 with frame-grouping on the RWF-2000 dataset. The leftmost column shows the input frames, and four dashed boxes on the right are four different types of perceptible CNN features of the I3D and MobileNetV3 with our proposed modules.

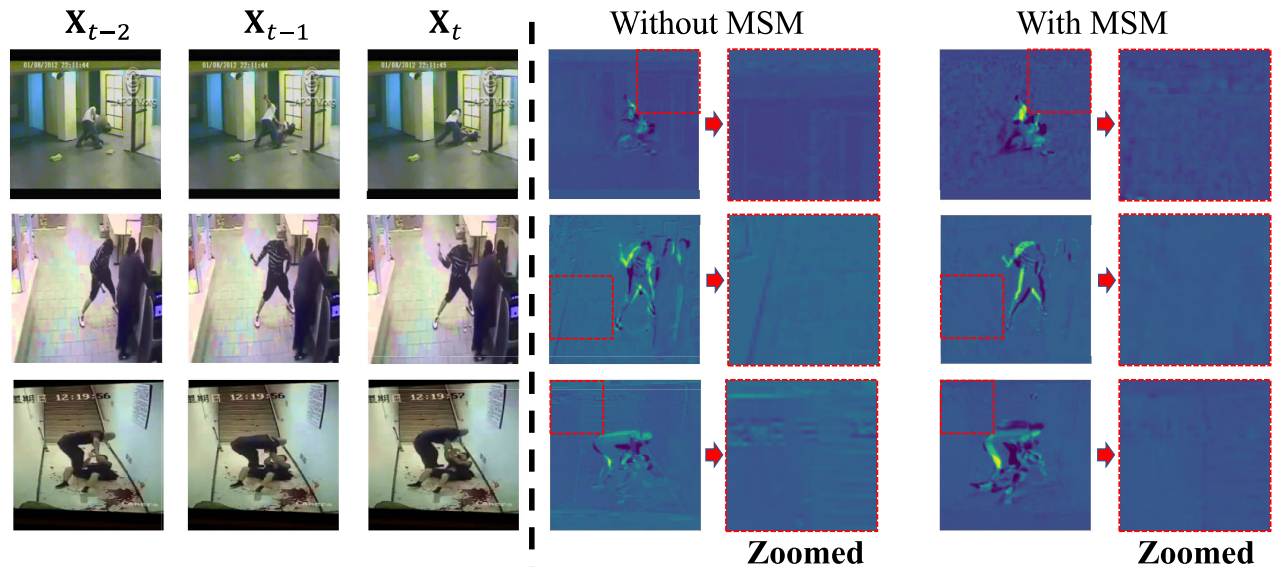



**FIGURE 9.** Output features for the first CNN layer of MobileNetV3 with frame-grouping without and with MSM. The results with MSM attenuate non-salient regions compared to those without MSM. (Best viewed in zoomed images).

interaction weights to model feature and position relations between extracted human skeleton points for violence recognition. astly, **VGG-16 + Bi-GRU** [64] used a pretrained VGG-16 followed by a bidirectional Gated Recurrent Units (GRU). Our approach achieved the best performances for all the six datasets.

**TABLE 4.** Recognition accuracies (%) evaluated on six violence video datasets. The datasets are abbreviated as follows: Hockey (Hockey-Fight), Movie (Movie-Fight or Peliculas), Crowd (Violent Flows), Surv (Surveillance Camera Fight), RLVS (Real Life Violence Situations), and RWF (RWF-2000). The results of our method (below the double horizontal line) are accuracies averaged over 5-fold cross-validation (for all datasets except for the RWF-2000 dataset that is explicitly divided into train/validation sets), whereas the results of other methods (above the double horizontal line) are from authors' reports. The best results for each dataset and our proposed modules are highlighted in bold.

| Model | Hockey | Movie | Crowd | Surv | RLVS | RWF |
|---|---|---|---|---|---|---|
| Inception-Resnet-V2 + DI [66] | 93.3 | **100.0** | - | - | 86.8 | - |
| VGG-16 + LSTM [62] | 95.1 | 99.0 | 90.0 | - | - | - |
| Xception + Bi-LSTM [63] | 98.0 | **100.0** | - | 72.0 | - | - |
| Flow Gated Network [9] | 98.0 | **100.0** | 88.8 | - | - | 87.3 |
| SPIL [12] | 96.8 | 98.5 | 94.5 | - | - | 89.3 |
| VGG-16 + Bi-GRU [64] | 98.0 | - | 95.5 | - | - | - |
| **MobileNetV3 with frame-grouping** | 98.9 | **100.0** | 95.9 | 88.0 | 97.6 | 88.8 |
| **MSM + MobileNetV3 with frame-grouping + T-SE block** | 99.3 | **100.0** | 96.1 | 88.3 | 97.7 | 90.0 |
| **EfficientNet-B0 with frame-grouping** | 99.4 | **100.0** | 97.1 | 91.3 | 97.5 | 89.5 |
| **MSM + EfficientNet-B0 with frame-grouping + T-SE block** | **99.6** | **100.0** | **98.0** | **92.0** | **97.8** | **92.0** |

**TABLE 5.** Ablation studies for all combinations of three proposed modules. Experiments were conducted using MobileNetV3 on the RWF-2000 dataset. $P_b$ represents the number of parameters for the baseline without all the three modules, which is 2,976,635. The best result was highlighted in bold.

| MSM | T-SE block | frame-grouping | Accuracy (%) | #Params |
|---|---|---|---|---|
| | | | 81.0 | $P_b$ |
| ✓ | | | 84.0 | $P_b + 2661$ |
| | ✓ | | 83.8 | $P_b + 900$ |
| ✓ | ✓ | | 84.8 | $P_b + 3561$ |
| | | ✓ | 88.8 | $P_b$ |
| ✓ | | ✓ | 89.3 | $P_b + 2661$ |
| | ✓ | ✓ | 89.3 | $P_b + 100$ |
| ✓ | ✓ | ✓ | **90.0** | $P_b + 2761$ |

**TABLE 6.** Ablation studies for light-weight CNN architectures with and without our proposed modules on the RWF-2000. F and A are the frame-grouping and the proposed attention modules (MSM and T-SE block), respectively.

| Module | Backbone | Accuracy (%) | FLOPS (G) |
|---|---|---|---|
| 2D CNN | SqueezeNet 1.1 | 78.3 | 10.6 |
| **+ F** | | 86.3 | **3.5** |
| **+ F + A** | | **89.0** | 7.5 |
| 2D CNN | MobileNetV2 | 82.8 | 9.4 |
| **+ F** | | 88.5 | **3.1** |
| **+ F + A** | | **89.3** | 7.1 |
| 2D CNN | MobileNetV3 | 81.0 | 6.7 |
| **+ F** | | 88.8 | **2.2** |
| **+ F + A** | | **90.0** | 6.2 |
| 2D CNN | EfficientNet-B0 | 83.0 | 0.4 |
| **+ F** | | 89.5 | **0.1** |
| **+ F + A** | | **92.0** | 4.2 |

**TABLE 7.** Comparison of frame-grouping with TSM on the RWF-2000 dataset. We experimented TSM with the codes and parameters from the author's github. F is the frame-grouping. The decision was made for 30 frames as an input without any spatio-temporal attentions. The best result was highlighted in bold.

| Module | Backbone | Accuracy (%) | FLOPS (G) |
|---|---|---|---|
| 2D CNN | MobileNetV2 | 82.8 | 9.4 |
| +TSM | | 85.0 | 9.4 |
| **+F** | | **88.5** | **3.1** |

### *D. ABLATION STUDY*

#### 1) ABLATION STUDIES ON THREE PROPOSED MODULES

We conducted ablation studies for all combinations of three proposed modules on the RWF-2000 dataset. As shown in Table 5, MSM (for spatial attention) and T-SE block (for temporal attention) consistently improved the performance without requiring many parameters. In particular, frame-grouping significantly increased the performance by enabling 2D CNN models to learn spatio-temporal information with considerable reduction of the computational complexity and memory demand since it reduced time steps by grouping three channel-averaged frames for an input of the 2D CNN. We showed the results of four CNN backbones on the RWF-2000 in Table 6.

#### 2) COMPARISON OF FRAME-GROUPING WITH TSM

We also compared frame-grouping with TSM on the RWF-2000 dataset. We considered the bidirectional TSM (offline TSM) and used the codes and hyperparameters from the author's github and report [54]. The TSM enhanced the performance of the conventional 2D CNN without additional computations by mingling some parts of channels with neighboring frames inside residual blocks of a backbone. The comparison in Table 7 shows that 2D CNN models with frame-grouping achieved better performance than those with the TSM since the frame-grouping explicitly mingled multiple frames to model spatio-temporal information. Furthermore, as the frame-grouping significantly decreases the memory footprint and FLOPS, it may be more suitable for an on-device real-time surveillance system.

### *E. IMPLEMENTATIONS OF AN ONLINE SURVEILLANCE SYSTEM*

The latencies of our models were measured on a single NVIDIA RTX 3090 and Jetson TX2. As shown in Table 8, our methods could achieve real-time or near real-time requirements. We implemented two types of webcam demos to simulate real-time surveillance systems depending on

**TABLE 8. Latencies of 2D CNNs with frame-grouping for processing 30 frames without and with attention modules (MSM and T-SE block). The results that met the real-time requirement (25 fps) were highlighted in bold.**

| Backbone | Attention modules | Jetson TX2 (ms) | RTX 3090 (ms) |
|---|---|---|---|
| SqueezeNet 1.1 | | **9.8** | **2.1** |
| | ✓ | **28.0** | **5.8** |
| MobileNetV2 | | **21.9** | **5.3** |
| | ✓ | **39.2** | **8.0** |
| MobileNetV3 | | **35.3** | **6.8** |
| | ✓ | 51.2 | **10.2** |
| EfficientNet-B0 | | 47.3 | **9.4** |
| | ✓ | 62.2 | **12.4** |

device performance. First, we implemented 2D CNNs with frame-grouping and no attention modules to detect violent actions with the minimum delays. The system started to detect violence after 30 frames were pushed into a queue and inferred the violence whenever an additional frame was captured from a camera. Second, we used our attention modules with the 2D CNNs and the frame-grouping. Since a sufficient time gap between consecutive frames might be helpful for detecting accurate motion boundaries in the MSM, we might skip some frames for inferring violence. For a typical 30-fps video, a single frame was pushed into a queue every five frames to get around 6 fps. The implemented system with our attention modules started to detect the violence every five frames instead of every frame after 30 frames were accumulated in the queue. In this case, MobileNetV3 with frame-grouping and attention modules implemented on a single Jetson TX2 inferred the violence in real time. We will post the real-time demo codes for both the methods and the corresponding demo videos on our github page.[1]

### F. EXPERIMENTS ON LONG-TERM VIOLENCE VIDEOS

We conducted experiments on UCF-Crime [73] that is the dataset containing 1900 untrimmed videos of 13 different anomalous events such as fighting, shooting, arson, road accidents, etc. To demonstrate the stability of our surveillance system, we conducted experiments on the test set of violence-related classes (assault and fighting) with the model trained on the RWF-2000 (without any additional training on the UCF-Crime or other datasets). Contrary to the datasets in Table 2, violent events take place for a short time in the long sequence of video. Therefore, we tested the trained models with the method mentioned in Subsection IV-E and measured area-under-the-curve (AUC) for videos in the UCF-Crime in Table 9 instead of accuracy. There are three assault videos and five fighting videos in the test set. Although the frame-level labels were provided for the test videos, we labeled our own for the violence-related classes to define the violent actions accurately. The label information and the demo videos for the test set will be posted on our github page.[1]

[1]https://github.com/ahstarwab/Violence_Recognition

**TABLE 9. AUC results of each violence-related class on the test set of the UCF-Crime dataset. Experiments were conducted using MobileNetV3 and EfficientNet-B0 with frame-grouping. We averaged the number of frames and the AUC for each class.**

| Backbone | Attention modules | AUC |
|---|---|---|
| MobileNetV3 | | 0.82 |
| | ✓ | 0.86 |
| EfficientNet-B0 | | 0.83 |
| | ✓ | 0.87 |

## V. CONCLUSION

We proposed spatio-temporal attention modules and frame-grouping method to build a practical violence detection system. For spatial attention, MSM was introduced to obtain salient regions derived from motion boundaries. For temporal attention, we introduced T-SE block that could recalibrate temporal features with a small number of additional parameters. In particular, frame-grouping was introduced that was a method averaging the channels and grouping three consecutive channel-averaged images as an input for 2D CNN. It could successfully model short-term dynamics that was a critical feature to classify violent actions such as kicking and punching. We demonstrated the efficiency of our proposed modules with efficient 2D CNN backbones through a variety of experiments and successfully implemented an real-time violence recognition system in a resource-constrained environment. In the future, we will collect more data and explore a variety of data augmentation techniques to train a more robust model. Also, we will extend our work to address various action recognition tasks for a versatile use.

## APPENDIX A
## INVESTIGATION OF THE EFFECTIVENESS OF MSM ON VIDEOS CAPTURED BY MOVING CAMERAS

MSM module is effective with a fixed camera since it calculates motion boundaries between two consecutive frames. Furthermore, the module still showed performance improvement even with datasets containing many videos captured by moving cameras. We showed visualization results of the module for six different datasets other than RWF-2000, including videos captured with moving cameras (on the left) and fixed cameras (on the right) in Fig. 10. Examples on the left were captured with moving cameras and had some distractions such as moving objects, but the MSM module still highlighted salient regions (corresponding to candidates for violence detection, including people who were fighting) successfully with slowly moving cameras. It demonstrates that the MSM module is still effective with slight movements of cameras, not to mention fixed cameras to emphasize salient regions.

## APPENDIX B
## ADEQUATE TIME INTERVAL FOR FRAME-GROUPING

The main purpose of frame-grouping is to extract useful information from multiple frames with the existing 2D CNN

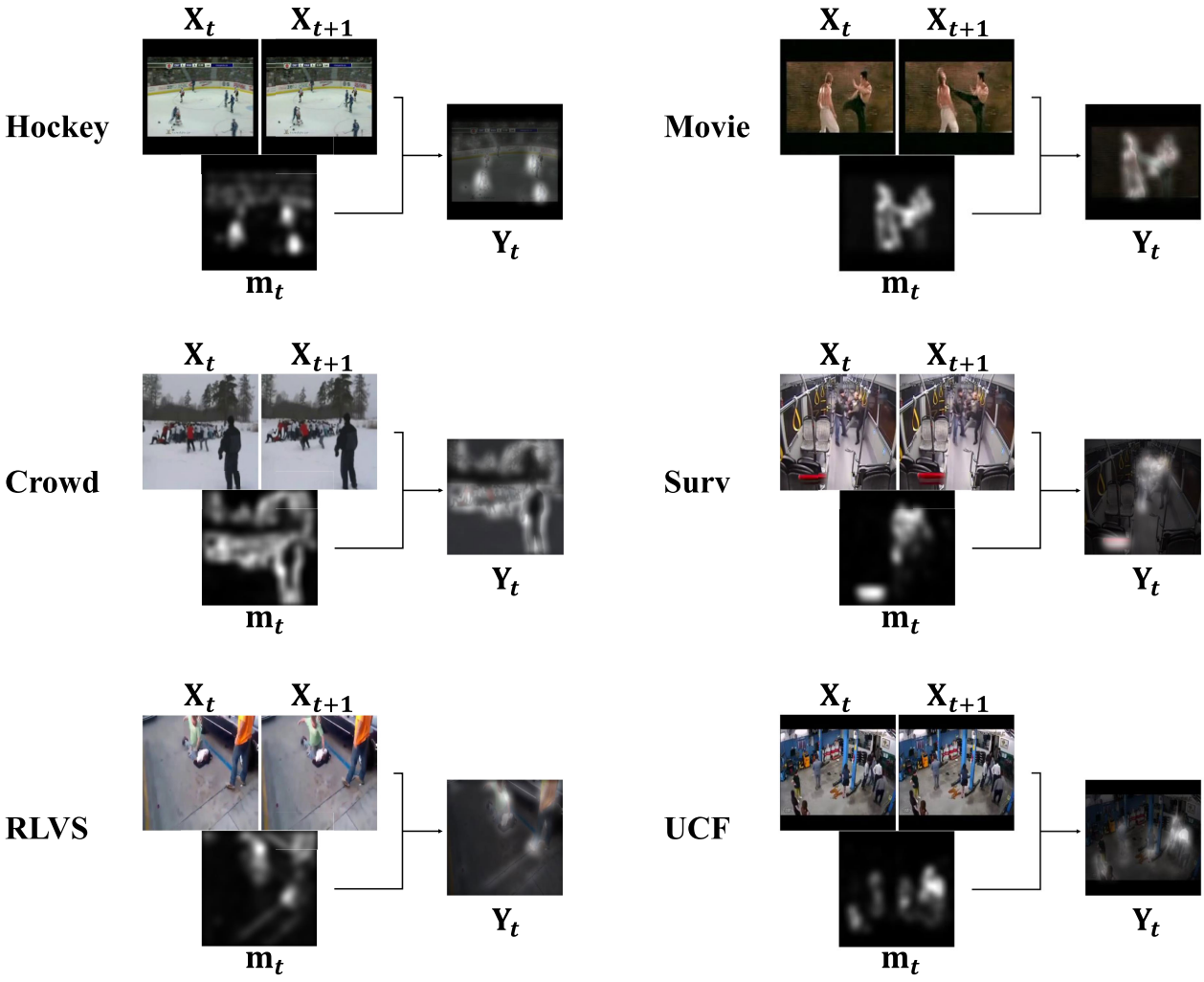



**FIGURE 10.** Results of MSM module on examples selected from six datasets: Hockey (Hockey-Fight), Movie (Movie-Fight or Peliculas), Crowd (Violent Flows), Surv (Surveillance Camera Fight), RLVS (Real Life Violence Situations), and UCF (UCF-Crime). Selected examples were captured with moving cameras (Left: Hockey, Crowd, and RLVS) and fixed cameras (Right: Movie, Surv, and UCF). (Best viewed in zoomed images).

backbones efficiently. We investigated the adequate time interval between consecutive frames to choose a sampling method. Each video varies in frame rate even in the same dataset because most of the datasets we explored are collected from Youtube. Therefore, determining the ideal time interval is impractical. For example, some videos are in slow motion and some videos have too low frame rates. Fig. 11 illustrates three consecutive frames after a uniform sampling (selecting frames at a uniform interval) for a video in each of six violence datasets (excluding UCF-Crime composed of long-term violence videos). Their time durations were different. In the case of RWF-2000, which is 5 seconds long for each video, we selected 30 frames out of about 150 frames (6 fps), and 3 frames were equal to 0.5 seconds long. On the other hand, in the case of Movie-Fight, which has the shortest video among the datasets we experimented (about 1 second), the time duration for three consecutive frames was about 0.1 seconds. Since 0.1∼0.5 seconds is enough to represent short-term dynamics such as punching and kicking, we used the uniform sampling throughout six datasets in Table 2.

## APPENDIX C
## DESCRIPTION OF EFFICIENT 2D CNNs WE USED

We explored four existing CNN architectures that are suitable for embedded systems. We described the used backbone networks in the published order.

### A. SqueezeNet [76]

There are two main strategies for reducing the model size and one strategy for minimizing performance degradation in Squeezenet. To reduce the model size, the authors replaced some of $3 \times 3$ filters with $1 \times 1$ filters. Moreover, they reduced the number of $3 \times 3$ filters. To maximize the accuracy, they placed the downsampling operation later so that early convolutional layers have larger activation maps. The main building block for SqueezeNet is called the ''Fire module'' and there are two stages in the module: ''squeeze

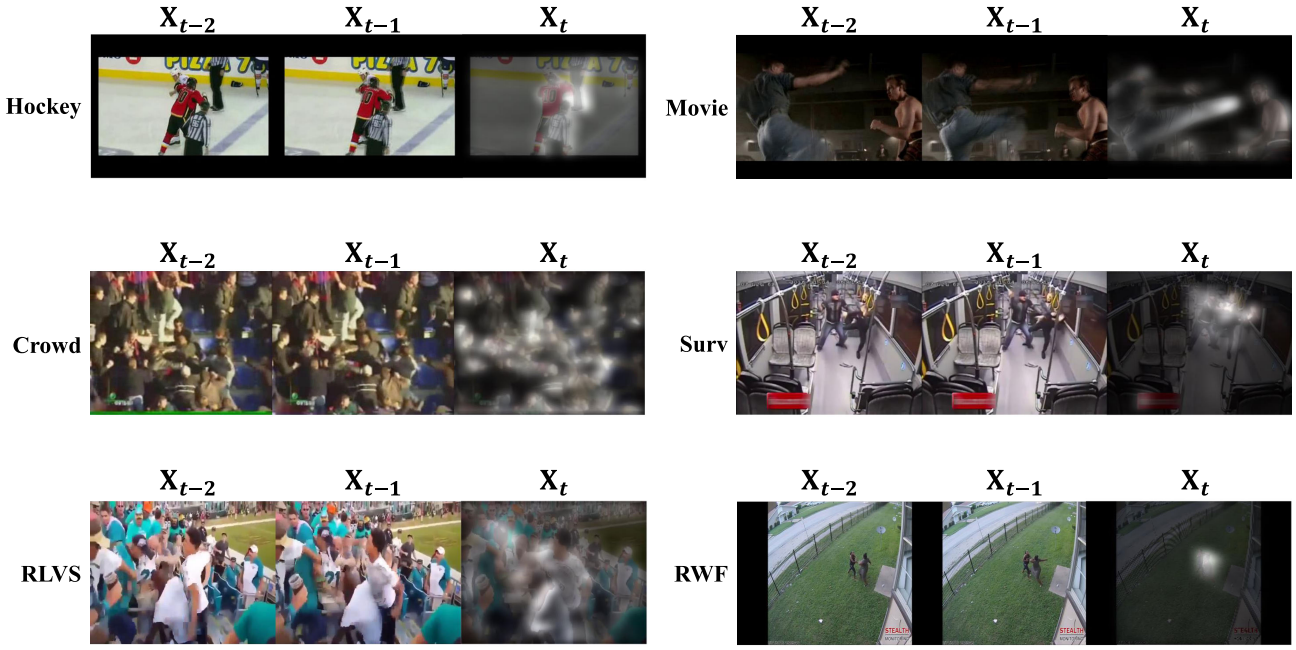



**FIGURE 11.** Three consecutive frames sampled with a uniform sampling for six violence video datasets. The datasets are abbreviated as follows: Hockey (Hockey-Fight), Movie (Movie-Fight or Peliculas), Crowd (Violent Flows), Surv (Surveillance Camera Fight), RLVS (Real Life Violence Situations), and RWF (RWF-2000). Every third frame is displayed by overlapping with MSM to clearly indicate attended regions.(Best viewed in zoomed images).

layer'' and ''expand layer''. The squeeze layer compresses data by performing $1 \times 1$ convolutions to reduce the number of parameters on the following $3 \times 3$ convolutions (in the expand layer). The expand layer consists of $1 \times 1$ and $3 \times 3$ convolution layers, and their outputs are concatenated afterward.

### B. MobileNets

MobileNets focus on reducing latency and the number of parameters in limited resources for mobile and embedded applications. Some core ideas for three versions of MobileNets are as follows.

#### 1) MobileNetV1 [31]

Depthwise separable convolution was introduced to reduce the number of operations and parameters in MobileNetV1. Depthwise separable convolution consists of depthwise convolution and pointwise convolution. Depthwise convolution applies a convolution filter to each input channel for spatial filtering, while pointwise convolution applies $1 \times 1$ convolution to the output of the depthwise convolution for mingling the channels.

#### 2) MobileNetV2 [32]

The main strategies introduced in MobileNetV2 were linear bottleneck and inverted residual blocks. In the linear bottleneck layer, the channel dimension of input is expanded to reduce the risk of information loss by nonlinear functions such as ReLU. It stems from the fact that information lost in some channels might be preserved in other channels. The inverted residual block has a (''narrow'' $\rightarrow$ ''wide'' $\rightarrow$ ''narrow'') structure in the channel dimension whereas a conventional residual block has a (''wide'' $\rightarrow$ ''narrow'' $\rightarrow$ ''wide'') one. Since skip connections are between narrow layers instead of wider ones, the memory footprint can be reduced.

#### 3) MobileNetV3 [33]

The architecture of MobileNetV3 was found through neural architecture search (NAS) for searching a global network structure (block-wise search), and the NetAdapt algorithm for searching the best number of filters (layer-wise search), starting from MnasNet [77]. The authors also introduced some hand-crafted contributions to improve the previous models (MobileNetV1 and V2). First, they reduced the number of filters in the first $3 \times 3$ convolution layer and removed some layers in the last stage to reduce the latency with little performance degradation. Second, they introduced h-swish (or hard swish) which is more quantization-friendly and faster in computation than the original swish.

### C. EfficientNet [78]

Contrary to MobileNets, Efficientnet focused on optimizing FLOPS rather than latency since it was not targeted at any specific hardware device. Before EfficientNet, most of the existing networks commonly adjust only one or two scaling factors among the number of layers (depth), the number of filters (width), and the resolution of input images (resolution). For instance, in MobileNets, the authors adjusted width and resolution to search the optimized models. On the other hand,

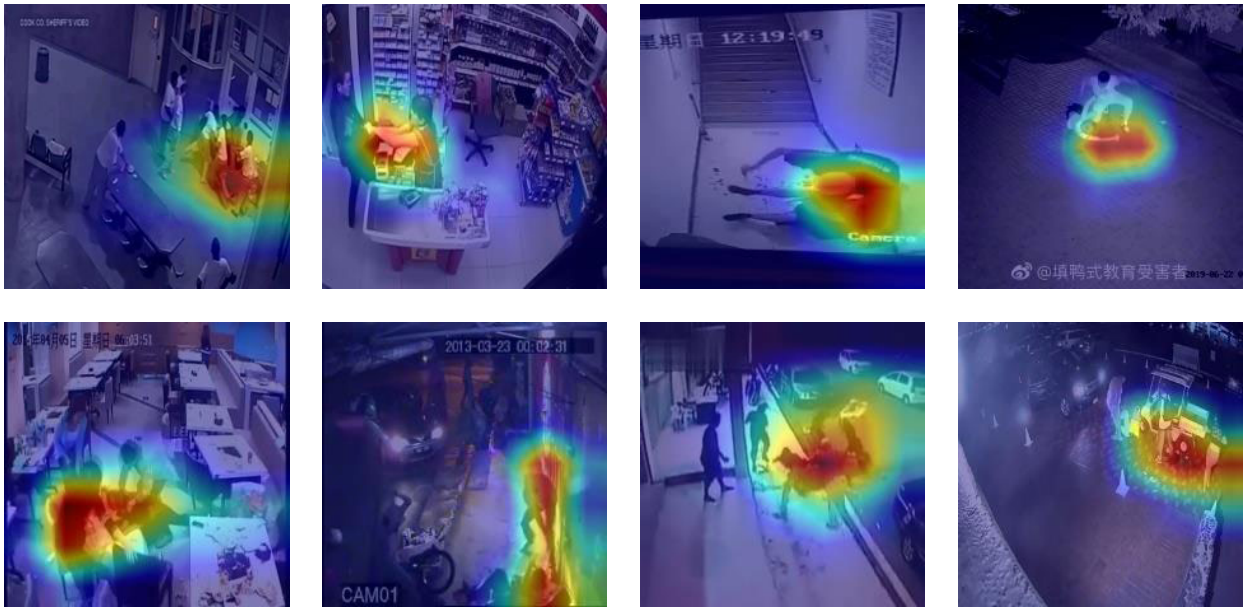



**FIGURE 12.** Grad-CAM results of EfficientNet with frame-grouping on the RWF-2000 dataset. Grad-CAMs of highest scored feature map in each video are illustrated on the corresponding single frame. (Best viewed in zoomed images).

the compound scaling method was proposed in EfficientNet to scale the depth, width, and resolution together. First, the authors defined $\alpha^{\phi}$, $\beta^{\phi}$, and $\gamma^{\phi}$ that are the amounts of scaling for depth, width, and resolution, respectively, where $\phi$ denotes a user-specified coefficient to control the model size. Second, they used NAS to find a baseline network named EfficientNet-B0. Third, they fixed $\phi = 1$ to determine constant values $\alpha$, $\beta$ and $\gamma$, and scaled up the model by adjusting $\phi$ to obtain EfficientNet-B1 to B7, depending on targeting FLOPS and model sizes.

## APPENDIX D
## GRAD-CAM RESULTS ON VIOLENCE DETECTION

We illustrate Grad-CAM [74] results of EfficientNet-B0 with frame-grouping in Fig. 12 to visualize the importance of the spatial locations to judge violence. Grad-CAM is a well-known technique for creating class activation maps to demonstrate the interpretability and transparency of deep-learning models. It visualizes a linear combination of the final convolutional layer's feature maps and the gradient of each class with respect to each feature map. Since our proposed system operates on multiple frames, we draw Grad-CAM of the highest scored feature map (made up of three frames) on the corresponding single frame (last of three frames).

## REFERENCES

[1] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, T. Darrell, and K. Saenko, "Long-term recurrent convolutional networks for visual recognition and description," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2625–2634.

[2] D. Li, Z. Qiu, Q. Dai, T. Yao, and T. Mei, "Recurrent tubelet proposal and recognition networks for action detection," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 303–318.

[3] J. Yue-Hei 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.

[4] L. Sun, K. Jia, K. Chen, D. Y. Yeung, B. E. Shi, and S. Savarese, "Lattice long short-term memory for human action recognition," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2166–2175.

[5] 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, pp. 4489–4497.

[6] 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. 4724–4733.

[7] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. 28th Annu. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, Dec. 2014, pp. 568–576.

[8] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 305–321.

[9] M. Cheng, K. Cai, and M. Li, "RWF-2000: An open large scale video database for violence detection," 2019, *arXiv:1911.05913*. [Online]. Available: http://arxiv.org/abs/1911.05913

[10] C. Wu, X.-J. Wu, and J. Kittler, "Spatial residual layer and dense connection block enhanced spatial temporal graph convolutional network for skeleton-based action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1740–1748.

[11] M. Li, S. Chen, X. Chen, Y. Zhang, Y. Wang, and Q. Tian, "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 3590–3598.

[12] Y. Su, G. Lin, J. Zhu, and Q. Wu, "Human interaction learning on 3D skeleton point clouds for video violence recognition," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K., Aug. 2020, pp. 74–90.

[13] S. Srinivas and R. V. Babu, "Data-free parameter pruning for deep neural networks," in *Proc. Brit. Mach. Vis. Conf.*, 2015, p. 31.

[14] S. Han, J. Pool, J. Tran, and W. Dally, "Learning both weights and connections for efficient neural network," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, Dec. 2015, pp. 1135–1143.

[15] H. Li, A. Kadav, I. Durdanovic, H. Samet, and H. Graf, "Pruning filters for efficient convnets," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–9.

[16] P. Molchanov, S. Tyree, T. Karras, T. Aila, and J. Kautz, "Pruning convolutional neural networks for resource efficient inference," in *Proc. 5th Int. Conf. Learn. Represent. (ICLR)*, Toulon, France, Apr. 2017, pp. 1–17.
[17] R. Yu, A. Li, C.-F. Chen, J.-H. Lai, V. I. Morariu, X. Han, M. Gao, C.-Y. Lin, and L. S. Davis, "NISP: Pruning networks using neuron importance score propagation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 9194–9203.
[18] J. Ye, "Generalized low rank approximations of matrices," *Mach. Learn.*, vol. 61, nos. 1–3, pp. 167–191, Nov. 2005.
[19] X. Zhang, J. Zou, X. Ming, K. He, and J. Sun, "Efficient and accurate approximations of nonlinear convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1984–1992.
[20] C. Tai, T. Xiao, Y. Zhang, X. Wang, and E. Weinan, "Convolutional neural networks with low-rank regularization," in *Proc. 4th Int. Conf. Learn. Represent. (ICLR)*, San Juan, Puerto Rico, May 2016.
[21] E. L. Denton, W. Zaremba, J. Bruna, Y. LeCun, and R. Fergus, "Exploiting linear structure within convolutional networks for efficient evaluation," in *Proc. Adv. Neural Inf. Process. Syst.*, Dec. 2014, pp. 1269–1277.
[22] Y. Gong, L. Liu, M. Yang, and L. Bourdev, "Compressing deep convolutional networks using vector quantization," 2014, *arXiv:1412.6115*. [Online]. Available: http://arxiv.org/abs/1412.6115
[23] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4820–4828.
[24] S. Shin, K. Hwang, and W. Sung, "Fixed-point performance analysis of recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 976–980.
[25] H. Zhang, J. Li, K. Kara, D. Alistarh, J. Liu, and C. Zhang, "The ZipML framework for training models with end-to-end low precision: The cans, the cannots, and a little bit of deep learning," 2016, *arXiv:1611.05402*. [Online]. Available: http://arxiv.org/abs/1611.05402
[26] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, *arXiv:1503.02531*. [Online]. Available: http://arxiv.org/abs/1503.02531
[27] G. Chen, W. Choi, X. Yu, T. Han, and M. Chandraker, "Learning efficient object detection models with knowledge distillation," in *Proc. Neural Inf. Process. Syst.*, Dec. 2017, pp. 742–751.
[28] R. G. Lopes, S. Fenu, and T. Starner, "Data-free knowledge distillation for deep neural networks," 2017, *arXiv:1710.07535*. [Online]. Available: http://arxiv.org/abs/1710.07535
[29] A. Polino, R. Pascanu, and D. Alistarh, "Model compression via distillation and quantization," in *Proc. 6th Int. Conf. Learn. Represent. (ICLR)*, Vancouver, BC, Canada, Apr. 2018, pp. 1–21.
[30] J. Liang, T. Zhang, and G. Feng, "Channel compression: Rethinking information redundancy among channels in CNN architecture," *IEEE Access*, vol. 8, pp. 147265–147274, 2020.
[31] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*. [Online]. Available: http://arxiv.org/abs/1704.04861
[32] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "MobileNetV2: Inverted residuals and linear bottlenecks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 4510–4520.
[33] A. Howard, M. Sandler, B. Chen, W. Wang, L.-C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, H. Adam, and Q. Le, "Searching for MobileNetV3," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 1314–1324.
[34] C. Guo, Q. Ma, and L. Zhang, "Spatio-temporal saliency detection using phase spectrum of quaternion Fourier transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[35] W. Kim, C. Jung, and C. Kim, "Spatiotemporal saliency detection and its applications in static and dynamic scenes," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 446–456, Apr. 2011.
[36] N. Nasaruddin, K. Muchtar, A. Afdhal, and A. P. J. Dwiyantoro, "Deep anomaly detection through visual attention in surveillance videos," *J. Big Data*, vol. 7, no. 1, pp. 1–17, Dec. 2020.
[37] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. M. Snoek, "VideoLSTM convolves, attends and flows for action recognition," *Comput. Vis. Image Understand.*, vol. 166, pp. 41–50, Jan. 2018.
[38] F. Xue, H. Ji, W. Zhang, and Y. Cao, "Attention-based spatial–temporal hierarchical ConvLSTM network for action recognition in videos," *IET Comput. Vis.*, vol. 13, no. 8, pp. 708–718, Dec. 2019.
[39] J. Li, X. Liu, W. Zhang, M. Zhang, J. Song, and N. Sebe, "Spatio-temporal attention networks for action recognition and detection," *IEEE Trans. Multimedia*, vol. 22, no. 11, pp. 2990–3001, Nov. 2020.
[40] O. Ulutan, S. Rallapalli, M. Srivatsa, C. Torres, and B. S. Manjunath, "Actor conditioned attention maps for video action detection," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2020, pp. 516–525.
[41] L. Meng, B. Zhao, B. Chang, G. Huang, W. Sun, F. Tung, and L. Sigal, "Interpretable spatio-temporal attention for video action recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1513–1522.
[42] Y. Wang, T. Bao, C. Ding, and M. Zhu, "Face recognition in real-world surveillance videos with deep learning method," in *Proc. 2nd Int. Conf. Image, Vis. Comput. (ICIVC)*, Jun. 2017, pp. 239–243.
[43] E. Jose, G. M., M. T. P. Haridas, and M. H. Supriya, "Face recognition based surveillance system using FaceNet and MTCNN on jetson TX2," in *Proc. 5th Int. Conf. Adv. Comput. Commun. Syst. (ICACCS)*, Mar. 2019, pp. 608–613.
[44] K. Assaleh, T. Shanableh, and K. Abuqaaud, "Face recognition using different surveillance cameras," in *Proc. 1st Int. Conf. Commun., Signal Process., Their Appl. (ICCSPA)*, Feb. 2013, pp. 1–5.
[45] K. Susheel Kumar, S. Prasad, P. K. Saroj, and R. C. Tripathi, "Multiple cameras using real time object tracking for surveillance and security system," in *Proc. 3rd Int. Conf. Emerg. Trends Eng. Technol.*, Nov. 2010, pp. 213–218.
[46] M. Chandrajit, R. Girisha, and T. Vasudev, "Multiple objects tracking in surveillance video using color and Hu moments," 2016, *arXiv:1608.06148*. [Online]. Available: http://arxiv.org/abs/1608.06148
[47] G. Mathur, D. Somwanshi, and M. M. Bundele, "Intelligent video surveillance based on object tracking," in *Proc. 3rd Int. Conf. Workshops Recent Adv. Innov. Eng. (ICRAIE)*, Nov. 2018, pp. 1–6.
[48] A. Gahalot, S. Dhiman, and L. Chouhan, "Crime prediction and analysis," in *Proc. 2nd Int. Conf. Data Engr. Appl. (IDEA)*, Aug. 2020, pp. 1–6.
[49] N. H. M. Shamsuddin, N. A. Ali, and R. Alwee, "An overview on crime prediction methods," in *Proc. 6th ICT Int. Student Project Conf. (ICT-ISPC)*, May 2017, pp. 1–5.
[50] G. A. Martínez-Mascorro, J. R. Abreu-Pederzini, J. C. Ortiz-Bayliss, A. Garcia-Collantes, and H. Terashima-Marín, "Criminal intention detection at early stages of shoplifting cases by using 3D convolutional neural networks," *Computation*, vol. 9, no. 2, p. 24, Feb. 2021.
[51] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, "Squeeze-and-excitation networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, Aug. 2020.
[52] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6450–6459.
[53] 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.
[54] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 7082–7092.
[55] C. Zhang, Y. Zou, G. Chen, and L. Gan, "PAN: Towards fast action recognition via learning persistence of appearance," 2020, *arXiv:2008.03462*. [Online]. Available: http://arxiv.org/abs/2008.03462
[56] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7794–7803.
[57] B. He, X. Yang, Z. Wu, H. Chen, S.-N. Lim, and A. Shrivastava, "GTA: Global temporal attention for video action understanding," 2020, *arXiv:2012.08510*. [Online]. Available: http://arxiv.org/abs/2012.08510
[58] A. Kozlov, V. Andronov, and Y. Gritsenko, "Lightweight network architecture for real-time action recognition," in *Proc. 35th Annu. ACM Symp. Appl. Comput.*, Mar. 2020, pp. 2074–2080.
[59] 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. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
[60] A. Hanson, K. PNVR, S. Krishnagopal, and L. Davis, "Bidirectional convolutional LSTM for the detection of violence in videos," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 280–295.

[61] J. Li, X. Jiang, T. Sun, and K. Xu, "Efficient violence detection using 3D convolutional neural networks," in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 2019, pp. 1–8.
[62] M. M. Soliman, M. H. Kamal, M. A. El-Massih Nashed, Y. M. Mostafa, B. S. Chawky, and D. Khattab, "Violence recognition from videos using deep learning techniques," in *Proc. 9th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Dec. 2019, pp. 80–85.
[63] S. Akti, G. A. Tataroglu, and H. K. Ekenel, "Vision-based fight detection from surveillance cameras," in *Proc. 9th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Nov. 2019, pp. 1–6.
[64] A. Traoré and M. A. Akhloufi, "2D bidirectional gated recurrent unit convolutional neural networks for end-to-end violence detection in videos," in *Proc. Int. Conf. Images Anal. Recognit. (ICIAR)*, Póvoa de Varzim, Portugal, Jun. 2020, pp. 152–160.
[65] S. Sudhakaran and O. Lanz, "Learning to detect violent videos using convolutional long short-term memory," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.
[66] A. Jain and D. K. Vishwakarma, "Deep NeuralNet for violence detection using motion features from dynamic images," in *Proc. 3rd Int. Conf. Smart Syst. Inventive Technol. (ICSSIT)*, Aug. 2020, pp. 826–831.
[67] P. Wu, J. Liu, Y. Shi, Y. Sun, F. Shao, Z. Wu, and Z. Yang, "Not only look, but also listen: Learning multimodal violence detection under weak supervision," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Glasgow, U.K., Aug. 2020, pp. 322–339.
[68] E. B. Nievas, O. D. Suarez, G. B. Garcia, and R. Sukthankar, "Hockey fight detection dataset," in *Computer Analysis of Images and Patterns*. Seville, Spain: Springer, Aug. 2011, pp. 332–339. [Online]. Available: http://visilab.etsii.uclm.es/personas/oscar/FightDetection/
[69] E. B. Nievas, O. D. Suarez, G. B. Garcia, and R. Sukthankar, "Movies fight detection dataset," in *Computer Analysis of Images and Patterns*. Seville, Spain: Springer, Aug. 2011, pp. 332–339. [Online]. Available: http://visilab.etsii.uclm.es/personas/oscar/FightDetection/
[70] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 1–6.
[71] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "ImageNet: A large-scale hierarchical image database," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 248–255.
[72] 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
[73] 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.
[74] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.
[75] D. Bahdanau, K. Cho, Y. Bengio, and R. Aharoni, "Neural machine translation by jointly learning to align and translate," in *Proc. 3rd Int. Conf. Learn. Represent. (ICLR)*, San Diego, CA, USA, May 2015, pp. 1–15.
[76] F. N. Iandola, S. Han, M. W. Moskewicz, K. Ashraf, W. J. Dally, and K. Keutzer, "SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size," 2016, *arXiv:1602.07360*. [Online]. Available: http://arxiv.org/abs/1602.07360
[77] M. Tan, B. Chen, R. Pang, V. Vasudevan, M. Sandler, A. Howard, and Q. V. Le, "MnasNet: Platform-aware neural architecture search for mobile," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 2820–2828.
[78] M. Tan and Q. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in *Proc. Mach. Learn. Res.*, vol. 97, K. Chaudhuri and R. Salakhutdinov, Eds. Long Beach, CA, USA: PMLR, Jun. 2019, pp. 6105–6114.

**MIN-SEOK KANG** received the B.S. degree in computer science and engineering from Sogang University, Seoul, South Korea, in 2019, where he is currently pursuing the master's degree in electronic engineering. In 2019, he joined the intensive program in artificial intelligence with Carnegie Mellon University, Pittsburgh, PA, USA, sponsored by the Ministry of Science and ICT, South Korea, for six months. His current research interests include speech processing and computer vision problems related to recognition.

**RAE-HONG PARK** (Life Senior Member, IEEE) received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1976 and 1979, respectively, and the M.S. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 1981 and 1984, respectively.

In 1984, he joined the faculty of the Department of Electronic Engineering, Sogang University, Seoul. In 1990, he spent his sabbatical year as a Visiting Associate Professor with the Computer Vision Laboratory, Center for Automation Research, University of Maryland, College Park. In 2001 and 2004, he spent sabbatical semesters with the Digital Media Research and Development Center (DTV image/video enhancement), Samsung Electronics Company Ltd., Suwon, South Korea. In 2012, he spent his sabbatical year with the Digital Imaging Business (Research and Development Team) and Visual Display Business (Research and Development Office), Samsung Electronics Company Ltd. He is currently an Emeritus Professor with the Department of Electronic Engineering and a Distinguished Professor with the ICT Convergence Disaster/Safety Research Institute, Sogang University. His current research interests include computer vision, pattern recognition, and video communication. He was a recipient of the 1987 Academic Award presented by the KITE, the 1990 Postdoctoral Fellowship presented by the Korea Science and Engineering Foundation (KOSEF), the 1997 First Sogang Academic Award, the 1999 Professor Achievement Excellence Award presented by Sogang University, and the 2000 Haedong Paper Award presented by the Institute of Electronics Engineers of Korea (IEEK). He is also a co-recipient of the Best Student Paper Award of the IEEE International Symposium Multimedia (ISM 2006) and IEEE International Symposium Consumer Electronics (ISCE 2011). From 1995 to 1996, he served as an Editor for *International Journal of Electronics Engineering* with the Korea Institute of Telematics and Electronics (KITE).

**HYUNG-MIN PARK** (Senior Member, IEEE) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer science from the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, in 1997, 1999, and 2003, respectively. From 2003 to 2005, he held a postdoctoral position at the Department of Bio Systems, KAIST. From 2005 to 2007, he was with the Language Technologies Institute, Carnegie Mellon University. In 2007, he joined the Department of Electronic Engineering, Sogang University, Seoul, South Korea, where he is currently a Professor. His main research interests include robust speech recognition and computer vision.