This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3050069, IEEE Transactions on Multimedia

Infrared Action Detection in the Dark via Cross-Stream Attention Mechanism

Xu Chen, Chenqiang Gao*, Chaoyu Li, Yi Yang and Deyu Meng

***Abstract*—Action detection plays an important role in the field of video understanding and attracts considerable attention in the last decade. However, current action detection methods are mainly based on visible videos, and few of them consider scenes with low-light, where actions are difficult to be detected by existing methods, or even by human eyes. Compared with visible videos, infrared videos are more suitable for the dark environment and resistant to background clutter. In this paper, we investigate the temporal action detection problem in the dark by using infrared videos, which is, to the best of our knowledge, the first attempt in the action detection community. Our model takes the whole video as input, a Flow Estimation Network (FEN) is employed to generate the optical flow for infrared data, and it is optimized with the whole network to obtain action-related motion representations. After feature extraction, the infrared stream and flow stream are fed into a Selective Cross-stream Attention (SCA) module to narrow the performance gap between infrared and visible videos. The SCA emphasizes informative snippets and focuses on the more discriminative stream automatically. Then we adopt a snippet-level classifier to obtain action scores for all snippets and link continuous snippets into final detection results. All these modules are trained in an end-to-end manner. We collect an Infrared action Detection (InfDet) dataset obtained in the dark and conduct extensive experiments to verify the effectiveness of the proposed method. Experimental results show that our proposed method surpasses the state-of-the-art temporal action detection methods designed for visible videos, and it also achieves the best performance compared with other infrared action recognition methods on both InfAR and Infrared-Visible datasets.**

***Index Terms*—Infrared video, temporal action detection, selective cross-stream attention.**

I. INTRODUCTION

TEMPORAL action detection, which aims to detect temporal boundaries for all action instances in untrimmed videos, plays an important role in the field of video understanding and attracts considerable attention due to its broad applications in surveillance video analysis, video retrieval, autonomous driving, human-computer interaction, video recommendation, *etc*. However, current temporal action detection methods are mainly based on visible videos and few of them consider scenes with low-light, especially in the dark. In such scenes, actions are difficult to be detected by existing methods, or even by human eyes. Compared with visible videos, infrared videos are more suitable for the dark environment [1] and resistant to background clutter, as shown in Fig. 1.

Fig. 1. Illustration of the action in different modalities in the dark. We compare the action 'drop' under visible imaging and infrared imaging, respectively. It is hard to recognize in visible imaging but obvious for infrared imaging.

Although there have been many works for infrared video analysis, most of them focus on the action recognition task, which only processes the trimmed video and predicts the corresponding action category. In this paper, we investigate the temporal action detection problem in the dark by using infrared videos, which is, to the best of our knowledge, the first attempt in the action detection community. To address this problem, a straightforward way is directly applying the detection framework designed for visible videos to infrared videos. However, there is still a large gap between infrared and visible videos so that current state-of-the-art methods degrade significantly when applied to infrared videos. To narrow this gap, we design a Selective Cross-stream Attention (SCA) module, which can flexibly prefer the discriminative stream and enhance features using the temporal attention mechanism across streams. Besides, we construct a light Flow Estimation Network (FEN) and integrate it into our model to adaptively generate the optical flow. Then we optimize it with the whole model jointly to obtain the action-related representation. Our designs are based on the following observations.

First, unlike the visible videos, the infrared videos lack the fine texture information, which is caused by the physical properties of the thermal imaging system. Existing infrared action analysis methods routinely adopt the multi-stream

* Corresponding author.

X. Chen, C. Gao are with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China and Chongqing Key Laboratory of Signal and Information Processing, Chongqing 400065, China (e-mail: lanncx@gmail.com; gaocq@cqupt.edu.cn).

C. Li is with the School of Communication and Information Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China (email: lichaoyu1997@gmail.com).

Y. Yang is with the Center for Artificial Intelligence, University of Technology Sydney, Ultimo, NSW 2007, Australia (e-mail: yi.yang@uts.edu.au).

D. Meng is with the Institute for Information and System Sciences and Ministry of Education Key Lab of Intelligent Networks and Network Security, Xian Jiaotong University, Xian 710049, China (e-mail: dymeng@mail.xjtu.edu.cn).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3050069, IEEE Transactions on Multimedia

(a) The infrared image and corresponding optical flow of one 'hug' instance.

(b) The infrared image and corresponding optical flow of another 'hug' instance.

Fig. 2. The infrared images and optical flows of two 'hug' action instances. In (a), the appearance information in the infrared image is more discriminative than optical flow, while (b) is the opposite.

framework to learn video representations [2]–[5], like using original images, optical flow, OF-MHI (the Motion-History Images [6] of the optical flow), *etc*. For these methods, a fixed operation is usually adopted to combine the outputs of different streams, *e.g*. concatenation, summation, pooling, *etc*. These hard operations are not adaptive for variations of infrared imaging characteristics and scenes. For example, as shown in Fig. 2, when performing the action of 'hug', the infrared video could have salient action characteristics, or sometimes the opposite, due to the difference in action amplitude and the influence of the movement from surrounding objects. Thus, the model should be adaptive for different features in the multi-stream framework. In this paper, we propose the SCA module for flexible feature combination and enhancement. It has a selective head for preferring the infrared stream or flow stream and an attention mechanism to enhance the long-term temporal information across two streams. The attention mechanism has been broadly used and verified in the video analysis community [7]–[9]. The difference between our work and these methods is that they only apply it on the identical feature and do not consider attention across different streams.

Second, existing temporal action detection methods in the visible domain routinely use optical flow to improve detection performance, but they compute optical flow independently apart from the whole model, which does not consider the relationship between the optical flow and the target task. Flow estimation networks [10], [11] can be integrated into the whole framework and jointly optimized to improve performance, but due to the introduction of a large number of parameters and increasing the difficulty of optimization, they are not adopted by current temporal action detection frameworks. Based on this observation, we design FEN for optical flow estimation. More specifically, we adopt an encoder-decoder architecture like FlowNet [10], but it is much lighter, which can be integrated into our model and optimized with other components.

Our contributions are summarized as follows:

- We propose an end-to-end infrared temporal action detection framework for low-light scenes. An SCA module and a jointly trained FEN are designed for narrowing the performance gap between visible videos and infrared videos.
- To well support researches on the infrared action detection problem in dark scenes, we collect an **Inf**rared action **Det**ection (**InfDet**) dataset. We select three typical actions, including 'fight', 'drop', and 'hug', to make our research closer to the real application.
- We perform extensive experiments on the InfDet dataset and other infrared action recognition datasets, and the experimental results verify the effectiveness of our method.

The remainder of this paper is organized as follows. Section II briefly reviews related works on action recognition and action detection. Section III introduces our proposed method in detail. Section IV introduces our dataset and exhibits extensive experiments to verify the effectiveness of our method. Section V concludes this paper.

II. RELATED WORK

A. Action Recognition

Action recognition is the foundation of action detection, which aims to classify a given video clip. In the visible domain, action recognition has been rapidly developed in recent years benefited from deep learning technologies like Convolutional Neural Networks (CNNs). Simonyan and Zisserman [12] designed a two-stream network to utilize both the RGB and optical flow stream independently, and fused their scores for final prediction. Tran *et al*. [2] proposed to model spatio-temporal information using 3D CNNs. To decrease the computation, R(2+1)D [13] decomposed the 3D filters into a spatial filter and a temporal filter. Carreira and Zisserman [14] inflated the weights from pre-trained 2D CNNs, which can leverage both the successful design and solid parameters of deep image classification architecture. Sun *et al*. [15] proposed the optical flow guided feature to generate compact motion representations. More recently, a novel 4D architecture [16] was proposed to model the long-range spatio-temporal representation. Besides, RNN based approaches [8], [17] were also popular to model the spatio-temporal representation in videos.

For the infrared video, it is crucial to learn informative and efficient feature representations. Since infrared frames are insufficient for texture detail, it is natural to adopt the multi-stream architecture. For the early investigation, Gao *et al*. [5]

employed low-level features such as HOG3D [18], 3DSIFT [19], Dense-Traj [20] *etc*. for infrared action recognition. They further proposed a two-stream framework based on 2D CNNs to extract extra features like optical flow and OF-MHI [21]. Some later methods adopted deeper networks than the previous works. Jiang *et al*. [22] designed a two-stream network using C3D [2]. They added a discriminative code layer on the top of the 3D CNNs to generate class-specific representations. Liu *et al*. [23] proposed a cross-dataset registration and generation framework. They extracted iDT [24] features for infrared data and visible data, respectively. Then they aligned and mapped features to a latent space for feature expression through an encoder. Finally, the SVM model was used for classification. This kind of registration and feature expression structure is well designed, but it can not be optimized together with the classifier using gradient descent manner. Recent methods mainly pursued richer representation by increasing the number of input streams. In the fatigue driving detection task, Ma *et al*. [4] tackled infrared images, optical flow, and OF-MHI via a three-stream architecture composed of three 2D networks. Then they concatenated these features and followed a 3D network. Liu *et al*. [3] proposed a global temporal representation and applied a three-stream framework to consider local, global, and spatio-temporal information together. Imran *et al*. [25] generated the stacked dense flow difference image and stacked saliency difference image from the infrared video, and proposed a four-stream framework consisted of CNN and RNN modules. In this paper, we explore an effective way to generate optical flow and obtain discriminative features automatically, with only input the single infrared stream.

B. Action Detection

Generally, the action detection task involves temporal action detection and spatio-temporal action detection. The former aims to detect the temporal boundaries and action categories for instances of specified actions, while the latter needs to further localize the spatial region where the action occurs.

1) temporal action detection: Localizing actions in the temporal dimension is similar to object detection in the spatial space. Therefore, temporal action detection methods usually follow the advanced framework of object detection, and it can be divided into two-stage methods and one-stage methods.

For two-stage methods, a large number of region proposals that may contain actions are generated firstly, and then the features corresponding to these regions are fed into a classifier and regressor to obtain the final predictions, including the action labels and temporal boundaries. The performances of these algorithms depend on the quality of the generated proposals, which makes current researches mainly focus on how to generate high-quality proposals. Shou *et al*. [26] employed a binary classifier to sort the proposals generated by the sliding window and then filtered them. Gao *et al*. [27] used the idea of boundary regression to precisely adjust the location of the sliding window beyond the binary classifier. Xu *et al*. [28] followed the framework of Faster R-CNN [29] and proposed an end-to-end architecture based on 3D CNNs. They extracted features for the input video and adopted the anchor-based method to generate proposals. After that, they applied the classification head and regression head to recognize actions and refine boundaries of actions, respectively. Besides, Zhao *et al*. [30] proposed an actionness score group method according to the watershed algorithm for proposal generation, which improved the efficiency significantly. More recently, Lin *et al*. [31] directly predicted the possible start and end locations on the actionness curve and generated proposals by matching these locations. In [32], a boundary matching mechanism was further proposed to improve the quality and efficiency of proposal generation. The sliding window based methods can cover the entire video and the actionness curve method can accurately locate actions. Gao *et al*. [33] proposed to combine them by using a complementary classifier. Besides, some other methods focused on modifying the detection strategy. Chao *et al*. [34] proposed a multi-scale architecture to align the receptive field with the action duration through the multi-tower structure and dilated convolutions. Zeng *et al*. [35] used a graph to model the relationship between proposals, and aggregated their features by a graph convolutional network.

Although two-stage methods can achieve better detection accuracy, they are complicated and slower in the inference stage. One-stage methods directly generate action categories and boundaries from the video simultaneously. Lin *et al*. [36] generalized the idea of SSD [37]. They extracted the snippet-level features, and then adopted 1D convolution layers for predicting localization offsets and action categories. Long *et al*. [38] utilized the characteristics of the Gaussian kernel to model the action composition. Piergiovanni *et al*. [39] also proposed a model using gaussian kernels. They constructed a convolutional module called temporal Gaussian mixture (T-GM) layer, which can capture longer temporal dependencies, to replace the traditional convolutional layer. Also, they used a soft attention mechanism to learn the parameters of the mixed gaussian kernel, which makes the TGM layer pay more attention to the periods that are helpful for the final classification. They achieved state-of-the-art performance on THUMOS'14 [40] dataset.

The above deep-learning-based temporal detection methods for visible videos have been broadly investigated, but there is no method to apply them to infrared data currently. Unlike these methods designed for the visible domain, we focus on generalizing this task to the infrared domain and narrow the large gap between different modalities.

2) spatio-temporal action detection: For spatio-temporal action detection, most of current frameworks implement it by two steps—frame-level detection and linking. Hou *et al*. [41] generalized the Faster R-CNN [29] architecture and linked action tubes across video clips. Huang *et al*. [42] explored an online approach for action detection and prediction. Köpüklü *et al*. [43] used a unified framework for frame-level action detection, which was inspired by the one-stage object detector in the image level. Some methods [7], [44], [45] investigated the relationship between objects and context. Duarte *et al*. [46] proposed an integrated framework to train the model end-to-end. It generalized the capsule network [47] from 2D to 3D, and directly processed the input video to get the final results.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3050069, IEEE Transactions on Multimedia

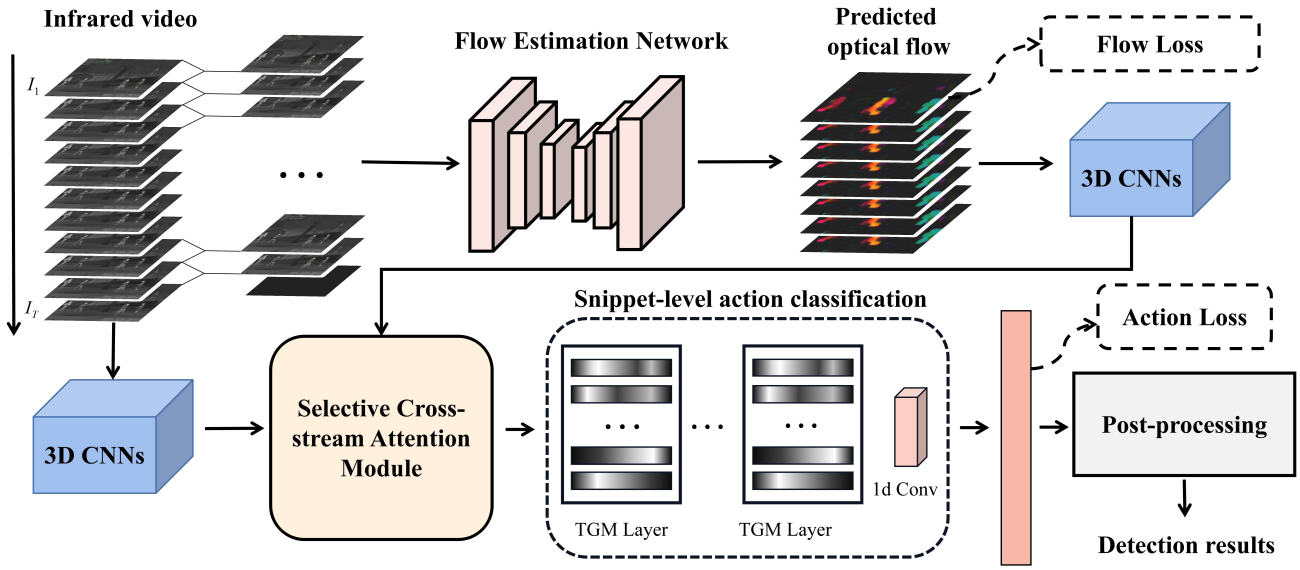

Fig. 3. Overview of our proposed framework, which is composed of five modules: Flow Estimation Network (FEN), feature extractor, Selective Cross-stream Attention (SCA) module, classifier, and post-processing.

III. METHODOLOGY

The framework of our proposed method is illustrated in Fig. 3. It takes the whole video as input, *i.e.*, $\{I_1, \cdots, I_T\}, I \in \mathbb{R}^{h \times w}$, where $h \times w$ is the spatial size of each frame and T denotes the video length, namely the number of frames. The original infrared frames are gray images, and we copy them three times in the channel to get the shape of $h \times w \times 3$. We employ a light architecture called Flow Estimate Network (FEN) to estimate the optical flow *i.e.*, $\{F_1, \cdots, F_{T-1}\}, F \in \mathbb{R}^{h \times w \times 2}$ for each timestamp t, where F_t is computed from infrared video frames *i.e.*, I_t and I_{t+1}. Then we follow a feature extractor (*e.g.* the I3D [14] network) to extract the feature for every l-frames snippet of $\{I_1, \cdots, I_T\}$ and $\{F_1, \cdots, F_T\}$, respectively. We set $l = 8$ in this paper. Let $X_I \in \mathbb{R}^{T' \times d}$ be the infrared feature and $X_F \in \mathbb{R}^{T' \times d}$ be the flow feature. d is the output dimension of 3D CNNs. T' is the temporal length of the output feature, which can be computed as $\frac{T}{l}$. After feature extraction, both the infrared feature and flow feature are fed to SCA module to obtain the enhanced feature $Y \in \mathbb{R}^{T' \times d}$ using a generalized attention mechanism. We further feed Y into the snippet-level classifier to get action scores $S \in \mathbb{R}^{T' \times C}$, where C is the number of action classes. Finally, we link these snippet-level predictions to generate final detection results.

A. Flow Estimation Network

The two-stream architecture is compatible with the infrared data, and it has been used in many methods [1], [5], [22] since the infrared video is insufficient in detailed texture information. The traditional optical flow estimator is off-the-shelf and can not be optimized jointly with neural networks. Meanwhile, current deep-learning-based flow estimators, like FlowNet [10] and FlowNetV2 [11], introduce too many parameters and computations. Thus, we design a lightweight network to estimate the optical flow, which guarantees controllable computations and training difficulties for our model.

Our FEN is specified in Table I. The dimensions of kernels, output channels, stride, and padding are formed as $\{k \times k, n_k, smpn\}$. For convolutional layers and pooling layers, $smpn$ means stride=m and padding=n. *Res* layers in Table I follow the skip connection in [48]. The *InsN* means instance normalization layer [49], and *ReLU* means rectified linear units [50]. We concatenate adjacent frames I_t, I_{t+1} along channels as the input. The encoder network maps the input data into a latent feature space to obtain the feature encoding. The estimated optical flow F_t for I_t and I_{t+1} can be formulated as:

$$F_t = tanh\left(Decoder\left(Encoder\left(concat\left(I_t, I_{t+1}\right)\right)\right)\right). \quad (1)$$

After that, we restore the resolution and detailed texture information by the decoder network and then generate the optical flow using a 1×1 convolutional layer. To reduce computations, the output resolution of FEN is a quarter of the input frame, and it will be restored to the original resolution after flow estimation. We use a $tanh$ function to limit the output values to $(-1, 1)$, which keeps the same preliminary normalization as infrared frames.

Inspired by [51], we jointly train FEN with the action classification loss. It helps to optimize the whole framework end-to-end, and the FEN will focus on task-related regions and ignore the background clutter. More details for training can be seen in Section III-D. After optical flow estimation, we use two I3D networks [14] to extract the infrared feature X_I and the flow feature X_F, respectively.

B. Selective Cross-stream Attention Module

Considering the entire input video, we can localize the scope of each action from a global perspective as long as

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3050069, IEEE Transactions on Multimedia

TABLE I
THE SPECIFIC CONFIGURATION OF THE FEN ARCHITECTURE

Stage	layer	Specification	output size
input	-	-	6×56×56
Encoder	*conv_1*	conv:{7×7, 64, s2, p3} *InsN* *ReLU*	64×28×28
	pool_1	max,3×3, s2p1	64×14×14
	Res_1	conv:{3×3, 128, s1p1} *InsN* *ReLU* conv:{3×3, 128, s1p1} *InsN* *ReLU*	128×14×14
	Res_2	conv:{3×3, 128, s2p1} *InsN* *ReLU* conv:{3×3, 128, s1p1} *InsN*	128×7×7
	Res_3	conv:{3×3, 128, s1p1} *InsN* *ReLU* conv:{3×3, 128, s1p1} *InsN*	256×7×7
	Res_4	conv:{3×3, 128, s1p1} *InsN* ReLU conv:{3×3, 128, s1p1} *InsN*	256×7×7
Decoder	*deconv_1*	transconv:{4×4, 128, s2p1} InsN LeakyReLU	128×14×14
	deconv_2	transconv:{4×4, 64, s2p1} InsN LeakyReLU	64×28×28
	deconv_3	transconv:{4×4, 32, s2p1} InsN LeakyReLU	32×56×56
flow prediction	*flow_conv*	conv:{1×1, 3, s1p0}	3×56×56

the computation and storage are sufficient. However, since the actions are submerged in massive unrelated background frames, it is necessary to measure the relationship of each snippet in the temporal dimension and enhance the combined feature based on these relationships. To effectively achieve the above properties, the SCA module is designed for leveraging both infrared and flow features. As shown in Fig. 4, our SCA module consists of two parts—a selective head and a cross-attention module.

1) ***Selective head****:* We design a selective head to automatically prefer different modalities. As shown in the left of Fig. 4, we feed the infrared feature X_I and the flow feature X_F as input. We concatenate them alone the channel and denote the output feature as X_c, and then we feed it into a regression network consisting of convolutional layers and fully connected layers to obtain a weighted scalar W. The sigmoid function is used to limit W to the range $(0, 1)$. We calculate two complementary streams for cross-attention as follows:

$$X_1 = X_I \cdot W + X_F\,(1 - W)\,, \tag{2}$$

$$X_2 = X_F \cdot W + X_I\,(1 - W)\,, \tag{3}$$

where we denote X_1 as the mainstream and X_2 as the auxiliary stream, respectively. The auxiliary stream is used to calculate attention maps in the next step. In this manner, the input of the cross-attention module is adaptive for different occasions that either infrared or flow feature is more discriminative.

2) ***Cross-attention****:* Self-attention is derived from [52] and can be applied in many other video analysis methods [7], [53]. It can enhance the feature for an input sequence by relating different positions in the temporal or spatio-temporal axis, and it aims to capture the global dependencies of features. However, in the self-attention mechanism, the input feature can be infrared feature X_I or flow feature X_F, but it can not directly utilize the two-stream data. When considering two streams, previous models [31], [36], [39] adopted some simple fusion strategies, like late fusion, summation, concatenate, *etc*. But these methods treat each snippet-level feature equally, without considering the different impact of snippets from another stream. Furthermore, based on the basis of self-attention, we generalize it to process the two-stream data. Because we leverage the correlation of two features to calculate the attention map, which corresponds to the self-correlation concept in the self-attention mechanism, we call it the cross-attention mechanism.

The implementation of the cross-attention mechanism is simple, which only needs to change the input data of the self-attention. Specifically, as shown in the right part of Fig. 4, given input features X_1 and X_2, we regard X_1 as the auxiliary stream, and X_2 as the mainstream. First, we apply three 1×1 convolutional layers named $query\_conv, key\_conv, value\_conv$ for input feature to reduce the channels d. The output features are formulated as follows:

$$F_q = query\_conv\,(X_1)\,, \tag{4}$$

$$F_k = key\_conv\,(X_2)\,, \tag{5}$$

$$F_v = value\_conv\,(X_1)\,. \tag{6}$$

The dimension for F_q, F_k, F_v is $T' \times d'$, where $d' = \frac{d}{8}$. Then a matrix multiplication between the transpose of F_q and F_k is performed to produce a matrix $G \in \mathbb{R}^{T' \times T'}$, which can be seen as the cross-correlation across time and streams:

$$G = F_k {F_q}^T. \tag{7}$$

The attention map can be produced by normalizing each row of G using a softmax function:

$$A_{ij} = \frac{exp\,(G_{ij})}{\sum_{j=1}^{T'} exp\,(G_{ij})}, \tag{8}$$

where A_{ij} is a score that measures the j^{th} snippet's impact on the i^{th} snippet. G_{ij} represents the inner product between i^{th} vector of F_k and j^{th} vector of F_q. To impose this impact to the original feature, a matrix multiplication between A and F_v is carried out. The enhanced output feature combines this result with the original input features:

$$P = AF_v, \tag{9}$$

$$Y = \gamma P + X, \tag{10}$$

where P denotes the weighted feature X_1 that relies on each snippet's impact of the X_2. Thus, it contains the global relationship across X_1 and X_2. We add this weighted feature to X_2. γ is a learnable scalar, and we set it to 0 at the beginning

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3050069, IEEE Transactions on Multimedia

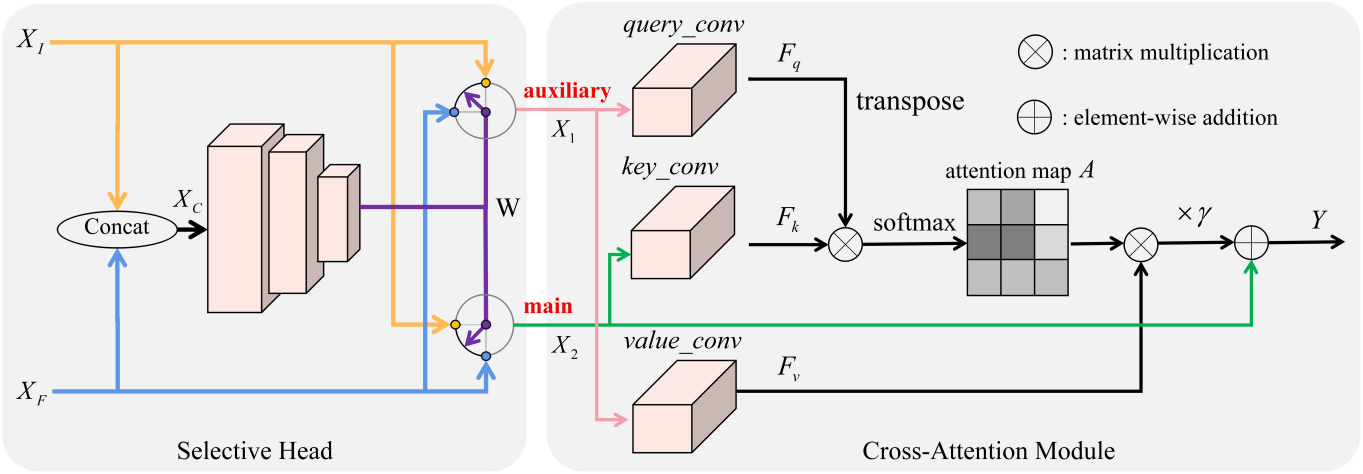

Fig. 4. Illustration of the selective cross-stream attention module. Left: We utilize a selective head to determine the mainstream and auxiliary stream automatically. X_I denotes the RGB stream and X_F is the flow stream. Right: The cross-attention module accepts two streams as input and computes the attentions of different snippets across streams. X_1 is the auxiliary stream for calculating attention map, and X_2 is the mainstream to be enhanced.

of the training phase.

C. Action detection

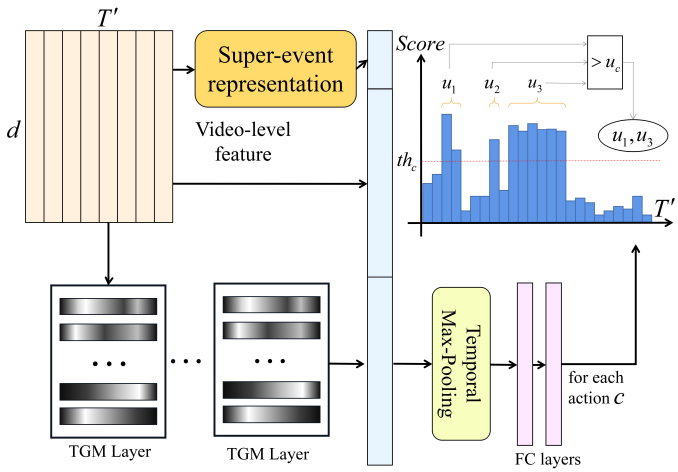

Fig. 5. Illustration of the detection pipeline. The snippet-level features are fed into temporal convolution layers and fully connected layers to obtain action scores. We generate detection results by grouping continuous snippets and filtering short detections.

For the enhanced feature Y, we use a snippet-level classifier to get the class-specific score S for each snippet at timestamp $t \in \{1, \cdots T'\}$. We adopt the commonly-used action score based procedure for the final detection. Specifically, as shown in Fig. 5, we first feed the enhanced feature into several TGM layers to capture the long-term temporal representation. We also adopt the super-event [54] branch, which has been verified effective for capturing latent representation between actions. The output features are concatenated with the original feature in the channel for an embedded feature *i.e.*, $Y_e \in \mathbb{R}^{T' \times (2 \times d + C)}$. Then we followed a global max pooling and two fully connected layers to obtain snippet-level action scores as follows:

$$S = fc(maxpool(Y_e)). \tag{11}$$

The final detection results are generated by aggregating continuous snippets that have larger scores than the action specified threshold of th_c. We remove detection results with too short durations. As shown in the upper right of Fig. 5, there are three action instances belonging to action c, and their durations are u_1, u_2, u_3, respectively. We only output u_1 and u_3 since $u_2 < u_c$, where u_c is the half of the average duration in the training set.

D. Loss Function

We have two subtasks in our framework, *i.e.*, flow estimation and snippet-level classification. For classification, we minimize the binary cross-entropy loss:

$$L_{cls} = -\frac{1}{T'} \sum_{t,c} \hat{S}_{t,c} \log\left(S_{t,c}\right) + \left(1 - \hat{S}_{t,c}\right) \log\left(1 - S_{t,c}\right), \tag{12}$$

where $\hat{S}_{t,c}$ is the ground truth label, which is 1 if action c is occurring at time t. $S_{t,c}$ is the output of our model for class c at time t. Since there is no ground truth flow for infrared videos, we use the optical flow generated by the TV-L1 algorithm [55] as ground truth $\hat{F}$. Then we compute the L1 loss:

$$L_{flow} = \frac{1}{T} \sum_{t} \sum_{i,j} |F_t(i,j) - \hat{F}_t(i,j)|. \tag{13}$$

Thus, the total loss is computed as $L = L_{cls} + \alpha L_{flow}$, which is the sum of classification loss and flow estimation loss with a weight factor α. We set α as 0.7.

IV. Experiments

In this section, we first introduce the proposed dataset and related evaluation metrics used in our experiments. And then we present the implementation details. After that, we compare the proposed method with the state-of-the-art action detection methods designed for visible videos on the proposed dataset. Additionally, we evaluate our SCA module and jointly

trained FEN in the ablation study. Qualitative visualization and extended experiments on other infrared action recognition datasets are also conducted to further validate the effectiveness of our method.

A. Dataset and Evaluation Metric

1) Dataset: To evaluate the temporal action detection methods for infrared videos, we collect a dataset called **InfDet**. It is derived from 11 hours of infrared videos and densely sampled from the night environment. The frame resolution is 293×256 for each infrared video. Since the original videos are too long, we divide them into 234 video clips. For every single video clip, we provide multi-instance annotations and all instances in this clip belong to the same action. Each action instance only occupies an average 5.5% duration of this clip. The minimal, maximal, and average durations of video clips are 24.08s, 108.56s, 57.68s, respectively. The distribution of action durations is shown in Table II. We note that some existing

TABLE II
DURATION DISTRIBUTION OF ACTION INSTANCES PER CLASS

Action	Duration		
	0-5s	5-10s	>10s
drop	279	2	0
fight	75	123	12
hug	332	17	3

video analysis benchmarks [56]–[58] contain a large number of infrared videos. However, most of them are captured by normal cameras with the near-infrared (NIR) device, which is quite different from ours, namely the thermal infrared sensor. As shown in Fig. 6, we compare InfDet dataset with NTU RGB-D 120 [56], which captured by Kinect V2 cameras. To get thermal infrared videos, we mainly consider the night scenes at distance, which is intractable both for RGB and ordinary NIR cameras (*e.g*. Kinect V2).

2) Metrics: A satisfactory prediction of the action detection model should meet two criteria: 1) the predicted action category is consistent with the ground truth action; 2) the temporal Intersection over Union (tIoU) is large enough. The definition of the IoU is:

$$IoU(a_i, \hat{a}_j) = \frac{a_i \bigcap \hat{a}_j}{a_i \bigcup \hat{a}_j}, \tag{14}$$

where the $\hat{a}_j$ is the i^{th} instance of ground truth actions $\hat{A} = (\hat{a}_1, \hat{a}_2, \cdots)$ and a_i is the j^{th} instance of predicted actions $A = (a_1, a_2, \cdots)$. The tIoU computes the intersection and union of actions on the temporal dimension. Similar to the evaluation of visible temporal action detection, we use mean Average Precision (mAP) as the metric, where the Average Precision (AP) is calculated on each action. A predicted action is considered to be correct if its tIoU with the ground truth instance is larger than a certain threshold, and the predicted category is the same as the corresponding ground truth instance. As the same as THUMOS14 [40], we choose tIoU thresholds from 0.1, 0.2, 0.3, 0.4, 0.5 on our InfDet dataset for evaluation.

B. Implementation details

Our model is implemented by the Pytorch framework. For the ground truth of FEN, we utilize the optical flow computed by TV-L1 and downsample it with factor 4, which can save the GPU memories and computations. We employ two I3D networks and load the weights pre-trained on Kinetics-400. We freeze them both in the training and testing phases to extract features from infrared videos and estimated optical flow, respectively. Like the I3D, we resize all input frames to 224×224 and normalize their values to (-1,1). In the training phase, we use Adam optimizer with a learning rate of $6 \times 0.01 \times \frac{batch\_size}{number\_of\_samples}$. The learning rate is decreased by a factor of 10 if the loss plateaued after 10 epochs. All the modules are trained in an end-to-end manner with 4 GPUs. In addition, we use the apex library to accelerate the training procedure. The code and pre-trained weight for this work will be available at GitHub [1].

C. Comparisons with State-of-the-art Methods

TABLE III
COMPARE WITH STATE-OF-THE-ART METHODS. '†' MEANS THE BACKBONE IS FINE-TUNED ON INFRARED VIDEOS.

Method	mAP@tIoU=α(%)				
	0.1	0.2	0.3	0.4	0.5
R-C3D [28]	4.8	4.4	4.0	3.6	3.1
BSN [31]	28.9	11.7	5.3	2.8	0.6
BMN [32]	17.6	15.6	13.7	12.4	11.2
TGM [39]	41.83	40.12	38.59	36.70	34.05
†TGM [39]	41.45	40.9	38.35	38.35	34.12
Ours	42.41	**42.12**	**41.69**	**40.64**	**37.83**
†Ours	**44.57**	42.09	39.7	38.47	37.09

Since there is no temporal action detection framework for infrared videos, we compare the proposed method with state-of-the-art methods in the visible domain, including R-C3D [28], BSN [31], BMN [32], and TGM [39]. For a fair comparison, we adopt these methods and fine-tune them on our InfDet dataset. For all of the compared methods, we use their official implements and follow the default configurations. Specifically, the R-C3D utilizes a C3D pre-trained on Sport-1M [59] as the feature extractor. The Two-Stream Network [12] pre-trained on ActivityNet [60] is employed by both of the BSN and BMN. For TGM, the I3D pre-trained on Kinetics [14] is adopted. Apart from the feature extractor, R-C3D, BSN, and BMN adopt the two-stage structure, *i.e*., they first generate proposals and then add a classification head and a boundary regression head. TGM generates the final detections directly based on the snippet-level action scores.

As shown in Table III, our method is superior to other methods for all the thresholds, and it is 3.78% higher than TGM with the threshold of 0.5. It demonstrates that our model is more adaptive for variational characteristics in infrared videos. The performance of R-C3D is obviously lower than other methods except for the threshold of 0.5. This phenomenon is probably caused by three factors: 1) R-C3D uses a shallower

[1]https://github.com/LannCX/InfDetNet

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3050069, IEEE Transactions on Multimedia

Fig. 6. Infrared frames of NTU RGB-D 120 (left) and InfDet (right). The infrared images in NTU RGB-D 120 are more like RGB images, and it only captures images in a close range. The infrared images in InfDet are quite different from RGB images and can capture images from distance, which is crucial in the out-door surveillance scenes.

C3D network as the backbone, which has a limited feature extraction capability compare to other 3D CNNs. 2) The temporal receptive fields of proposals are relatively short since the anchor-based temporal range is limited. 3) Except for R-C3D, all the other methods used additional optical flow data in the original paper, which can improve performance naturally. BSN uses a two-stage architecture, and it considers more information in the proposal generation stage. Surprisingly, the performance of BSN decreases sharply as the threshold rises, even lower than R-C3D for the threshold of 0.5. BMN is an enhanced version of BSN. Under the same framework, a more effective boundary matching mechanism is adopted for BMN to improve the quality of proposals. Since the performance of BMN does not significantly degrade, it can be judged that the sharply decreased performance of BSN is caused by the poor quality of proposal generation. Among the visible temporal detection methods, TGM is the best method that achieves 34.05% mAP@tIoU=0.5. It is in accordance with the comparison in the THUMOS'14 dataset. One notable reason is that TGM utilizes more discriminative features provided by I3D pre-trained on a large scale dataset, which makes TGM have more advantages when compared with BSN and BMN. We further compare the average precision with the threshold of 0.5 on different actions in Fig. 7. It can be seen that our method outperforms the other methods for all three classes. The performance of the 'hug' is more than twice of the TGM.

For a more comprehensive comparison, we fine-tune the I3D on the Infrared-Visible [61] dataset with a learning rate of 0.01, then we use it as a feature extractor for TGM and the proposed

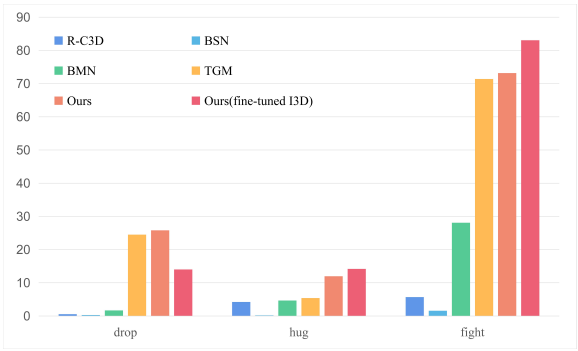

Fig. 7. Comparison of different methods per class at tIoU=0.5.

method. As shown in TABLE III, the fine-tuned backbone do not achieve promising improvements, and even worse in some cases. The fine-tuned TGM can improve mAP@tIoU=0.2, 0.4, 0.5, but fails at 0.1 and 0.3. For the fine-tuned model of the proposed method, it only improves mAP@tIoU=0.1. Overall, there are no obvious differences between the original model and the fine-tuned one. We further analyze the per-class average precision in Fig. 7 and find that the fine-tuned model can improve the performance of 'fight' and 'hug', but is worse for 'drop', which makes the average AP slightly lower than the original model. Therefore, fine-tuning can transfer the existing prior information in the visible domain to infrared videos, but the benefits for actions are different. For some hard examples, it may even harm the performance.

We measure the running time of our method using a

machine equipped with an Intel Xeon E5 2.2GHz and 4 Tesla V100. Timing is averaged over 100 runs, and we fed a 100-frames dummy video for each run. Finally, it can achieve 290 frames per second (FPS). However, we can not give a direct comparison for other action detection methods since they adopt action detection upon the off-the-shelf feature extraction for both RGB and flow streams, while our framework is end-to-end. Nevertheless, we can make a brief comparison with TGM. The classification module in our method keeps the same computations with TGM, and our cross-stream attention module can achieve roughly 3% gain while only introduce 135M extra multiplication and addition operations per 8 frames.

D. Ablation Study

To investigate the effectiveness of modules used in our method, we employ TGM as the baseline method, and then implement different strategies on it. The results are shown in Table IV, and we report the mAP at tIoU of 0.5 for an intuitive comparison. For convenience, we use SA, CA denote self-attention, and cross-attention, respectively. First of all, feeding infrared or flow data independently can not achieve high performance, and the self-attention mechanism influences the performance slightly. The result of the cross-attention mechanism is close to self-attention since both of them can not choose the input streams for different occasions. Our SCA module can get a 2.94% improvement compared with the two-stream baseline. It is an important component that can increase the discrimination of features. In addition, using the task specified optical flow can further bring 0.84% improvement.

As shown in Table V, we also compare different feature combination methods. Common multi-stream feature combination for infrared data include late fusion, element-wise maximization, element-wise minimization, concatenate *etc*. For element-wise maximization and minimization, we just take the two features from different streams and compare the value of them one-by-one. It is simple to implement and can obtain promising improvement because the maximum or minimum operation can increase differences of feature values, which makes the combined feature more discriminative. For the concatenate method, we concatenate the two features in the channel and change the input channel of TGM. That is, we do not change any values of input features. It does not bring a significant improvement, and its gain is lower than simple operations. It is worth mentioning that, we fuse the final snippet-level classification scores of two streams and it only gets 0.28% gain. This may be caused by the poor performance of flow data. We can observe from the Table V that all the combination methods can get gains upon the baseline. Among them, our SCA module gets a 3.22% gain, which is higher than the common feature combination methods.

We compare the number of parameters and theoretical floating-point operations per second (FLOPs) with other optical flow estimators. The input frames are resized to 256×256, and we calculate per frame computation for all compared methods. During testing, we adopt pytorch implementation of the FlowNetv2[2] and calculate FLOPs by using OpCounter toolbox[3]. We omit the computation of correlation operations for FlowNetC and LiteFlow v1-v3. As shown in Table VI, we first show the model size of three variations in FlowNet [10] and FlowNet V2 [11]. There are two variations in FlowNet—FlowNetS, and FlowNetC. In FlowNetV2, there is a lightweight variation FlowNetSD, and other variations explore more complicated strategies of stacking FlowNetS and FlowNetC than FlowSD, so we just compare with this one. As we can see from Table VI, our FEN only introduces much less parameters and computations than other general learnable optical flow estimators.

In the second section, we compare with the lightweight LiteFlowNet [62]–[64]. Our FEN is still better in terms of model size and FLOPs. Other optical flow estimators, like LiteFlowNet v1 and v2, have to design the architecture carefully to achieve good estimation accuracy, which limits the further reduction of model size. Different from these methods, we pursuit light architecture since we need to feed the whole video as input. The proposed FEN in this paper is mainly designed to complement the action-related motion representation. Thus we can force it as light as possible.

TABLE IV
COMPARISONS OF DIFFERENT MODULES. THE SA, CA DENOTE SELF-ATTENTION AND CROSS-ATTENTION, RESPECTIVELY.

Method	Flow	mAP@tIoU=0.5
infrared only	-	33.77
flow only	TV-L1	25.69
two-stream	TV-L1	34.05
infrared+SA	-	33.81
two-stream+CA	TV-L1	33.82
two-stream+SCA	TV-L1	36.99
two-stream+SCA	FEN	**37.83**

TABLE V
COMPARISONS OF DIFFERENT FEATURE COMBINATION METHODS. THE BASELINE METHOD IS TGM THAT USES INFRARED DATA ONLY.

Method	mAP@tIoU=0.5	Gain
baseline	33.77	-
late fusion	34.05	0.28
min	35.49	1.72
max	34.60	0.83
concatenate	34.16	0.39
our SCA	**36.99**	**3.22**

E. Qualitative Analysis

1) The detection results: We show three different detections in Fig. 8, in which the blue lines represent the real actions, while the green lines are predicted actions. The first and second rows are the detections of 'hug' and 'fight', respectively. It can be observed that our model is accurate in terms of action category prediction and boundary localization. Among them,

[2]https://github.com/NVIDIA/flownet2-pytorch
[3]https://github.com/wzmsltw/pytorch-OpCounter

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2021.3050069, IEEE Transactions on Multimedia

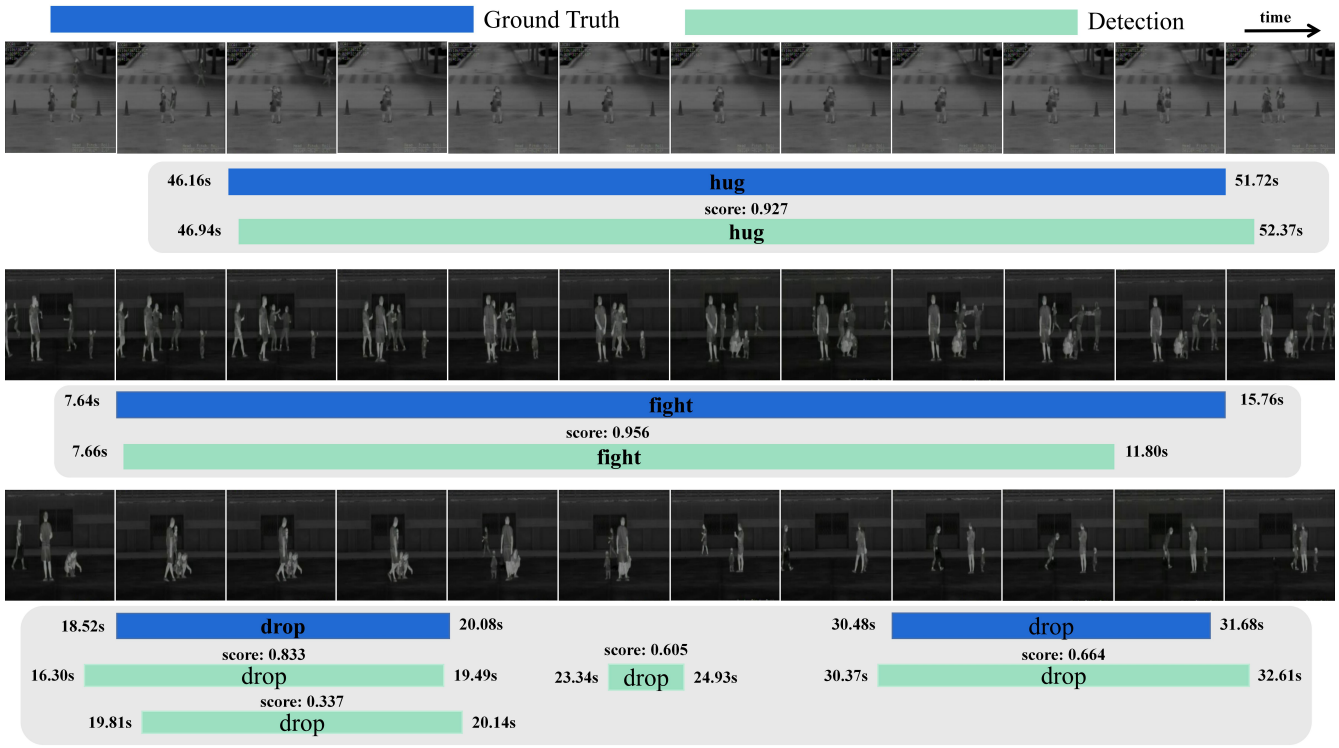

Fig. 8. Qualitative examples of high-quality detections (top two rows) and low-quality detections (bottom row) generated by our method on the InfDet dataset.

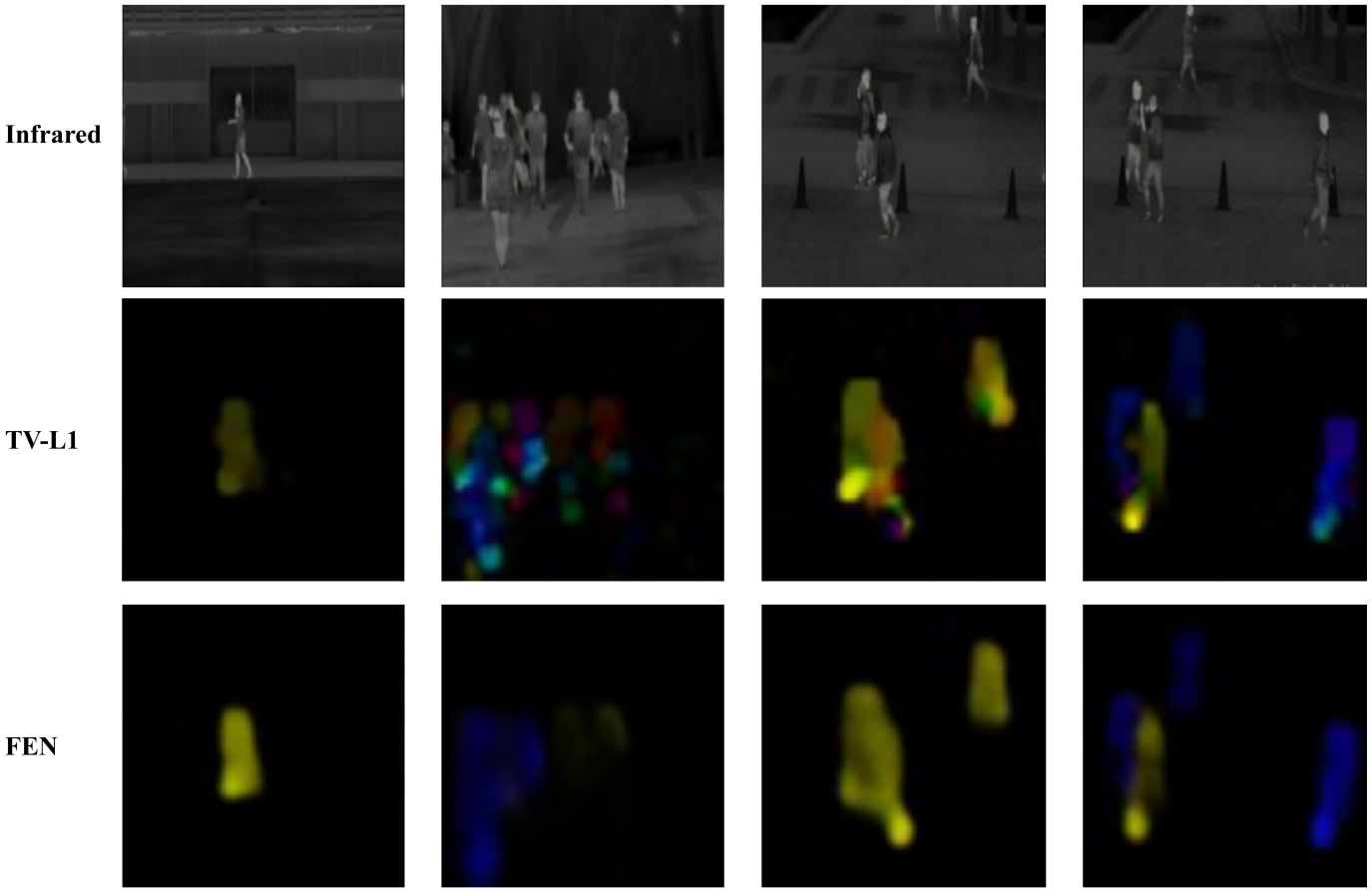

Fig. 9. Qualitative examples of the optical flow generated by the TV-L1 (second row) and FEN (third row). The first row shows the corresponding original infrared images.

TABLE VI
COMPARISONS OF DIFFERENT OPTICAL FLOW ESTIMATORS.

Model	Model Size	FLOPs
FlowNetS [10]	38.67M	8.9G
FlowNetC [10]	39.17M	11.1G
FlowNetSD [11]	45.37M	12.1G
LiteFlowNet v1 [62]	5.37M	18.9G
LiteFlowNet v2 [63]	6.42M	7.3G
LiteFlowNet v3 [64]	5.28M	9.9G
FEN	**3.41M**	**4.4G**

although there is some ambiguity in the boundary determination of the 'fight' action, the model still works well. It is worth noting that the 'drop' actions shown in the third row have a serious occlusion problem and the amplitude of the actor is not obvious enough. At the same time, the interval between the two actions is very short, which makes the detection difficult. Due to the similarity of the intermediate frames, a false alarm appears. However, our model still matches the real actions.

2) The estimated optical flow: We visualized the optical flow predictions in Fig. 9. The first row shows infrared images, while the second row shows the optical flow calculated by the TV-L1 algorithm according to the current frame and the next frame. The third row shows optical flow images estimated by the FEN. It can be seen that the optical flow calculated by the TV-L1 algorithm involves all pixels, but at the same time, it also introduces many noises, which is harmful to action recognition. On the contrary, the optical flow estimation module that we optimize together with the entire network will focus on the vicinity of the action occurring.

F. Extended experiments for infrared recognition

To compare the effectiveness of our feature representation with other methods in the infrared domain, we conduct extended experiments on InfAR [5] and Infrared-Visible [61], both of which are broadly used by infrared action recognition methods. The frame resolution for both two datasets is 293×256, which is the same as InfDet. We average the output action scores of the model over time and remove the original post-processing operations so that it can be used for action recognition. For a fair comparison, we unfreeze the parameters of 3D CNNs and fine-tune them on the target dataset.

1) Experiments on Infrared-Visible: Table VII shows the results on the Infrared-Visible dataset. All the methods use infrared data or optical flow generated from infrared data. The PM-GANs trains a generator to obtain fully-modal representation, which includes the infrared feature and visible feature. It gets a higher performance thanks to the visible information. The performance of our method is 88.18%, which is higher than the state-of-the-art PM-GANs [61] by 10.18%. We also utilize the fixed I3D and just fine-tune the last classification layer, and it achieves 78.9% top-1 accuracy, which is better than the previous state-of-the-art method (78%) and two-stream methods. Therefore, we can conclude that the I3D pre-trained on Kinetics is powerful to capture motion information, even applied to infrared videos. In order to remove the performance gain brought by the powerful I3D, we separately train the I3D network for infrared action recognition. Specifically, we load the weights pre-trained on Kinetics-400 and use SGD to optimize all the parameters by 300 epochs. As can be seen from TableVII, just using I3D can get a 6% improvement compared with PM-GANs, and our method can further improve performance by 4.18%, which shows the superiority of our feature representation.

TABLE VII
COMPARISONS OF DIFFERENT METHODS ON THE INFRARED-VISIBLE DATASET. '*' DENOTES THE MODEL IS NOT FINE-TUNED EXCEPT THE FINAL CLASSIFICATION LAYER.

Method	Accuracy(%)
iDT [24]	72.33
C3D [2]	69.67
Two-stream 2D-CNN [5]	68.00
Two-stream 3D-CNN [22]	74.67
PM-GANs [61]	78.00
*I3D [14]	78.90
I3D [14]	84.00
Ours	**88.18**

2) Experiments on InfAR: We train and test our model following the suggestions in [5], and the results are shown in Table VIII. For CDFAG [23], it has done a lot of experiments, and each result has the upper and lower bounds of the performance. What we list in the table is the upper bound of the best results in all experiments. Our method outperforms all the previous networks, including that using traditional features [20], [24], general two-stream networks [5], [22], three-stream [3], and even four-stream networks [25].

TABLE VIII
COMPARISONS OF DIFFERENT METHODS ON THE INFAR DATASET.

Method	Average accuracy(%)
HOF [5]	68.58
DT [20]	68.66
iDT [24]	71.83
Two-stream 2D-CNN [5]	76.66
Two-stream 3D-CNN [22]	77.50
CDFAG [23]	78.55
TSTDDs [3]	79.25
Four-stream CNN [25]	83.50
Ours	**84.25**

V. CONCLUSION

In this paper, we first explore temporal action detection in the low-light environment. We collect a temporal action detection dataset, which is densely sampled from the night environment, to promote future researches for this task. We generalize the temporal action detection task from the visible domain to the infrared domain and commit to narrow the gap between the two modalities. In our framework, we design a light Flow Estimation Network for generating optical flow and we jointly train the flow loss with snippet-level classification

loss. Furthermore, a Selective Cross-stream Attention module is proposed to automatically select and combine different streams. The experiments show that the proposed method achieves state-of-the-art performance on the infrared temporal action detection task and infrared action recognition task, which demonstrate the effectiveness of our model.

REFERENCES

[1] A. Ulhaq, "Action recognition in the dark via deep representation learning," in *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*. IEEE, 2018, pp. 131–136.

[2] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 4489–4497.

[3] Y. Liu, Z. Lu, J. Li, T. Yang, and C. Yao, "Global temporal representation based cnns for infrared action recognition," *IEEE Signal Processing Letters*, vol. 25, no. 6, pp. 848–852, 2018.

[4] X. Ma, L.-P. Chau, K.-H. Yap, and G. Ping, "Convolutional three-stream network fusion for driver fatigue detection from infrared videos," in *2019 IEEE International Symposium on Circuits and Systems (ISCAS)*. IEEE, 2019, pp. 1–5.

[5] C. Gao, Y. Du, J. Liu, J. Lv, L. Yang, D. Meng, and A. G. Hauptmann, "Infar dataset: Infrared action recognition at different times," *Neurocomputing*, vol. 212, pp. 36–47, 2016.

[6] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 23, no. 3, pp. 257–267, 2001.

[7] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 244–253.

[8] Z. Fan, X. Zhao, T. Lin, and H. Su, "Attention-based multiview re-observation fusion network for skeletal action recognition," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 363–374, 2018.

[9] J. Li, X. Liu, W. Zhang, M. Zhang, J. Song, and N. Sebe, "Spatio-temporal attention networks for action recognition and detection," *IEEE Transactions on Multimedia*, 2020.

[10] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. Van Der Smagt, D. Cremers, and T. Brox, "Flownet: Learning optical flow with convolutional networks," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 2758–2766.

[11] E. Ilg, N. Mayer, T. Saikia, M. Keuper, A. Dosovitskiy, and T. Brox, "Flownet 2.0: Evolution of optical flow estimation with deep networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2462–2470.

[12] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Advances in neural information processing systems*, 2014, pp. 568–576.

[13] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 6450–6459.

[14] J. Carreira and A. Zisserman, "Quo vadis, action recognition? a new model and the kinetics dataset," in *proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 6299–6308.

[15] S. Sun, Z. Kuang, L. Sheng, W. Ouyang, and W. Zhang, "Optical flow guided feature: A fast and robust motion representation for video action recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1390–1399.

[16] S. Zhang, S. Guo, W. Huang, M. R. Scott, and L. Wang, "V4d: 4d convolutional neural networks for video-level representation learning," *arXiv preprint arXiv:2002.07442*, 2020.

[17] D. Li, T. Yao, L.-Y. Duan, T. Mei, and Y. Rui, "Unified spatio-temporal attention networks for action recognition in videos," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 416–428, 2018.

[18] A. Klaser, M. Marszałek, and C. Schmid, "A spatio-temporal descriptor based on 3d-gradients," 2008.

[19] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional sift descriptor and its application to action recognition," in *Proceedings of the 15th ACM international conference on Multimedia*, 2007, pp. 357–360.

[20] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *CVPR 2011*. IEEE, 2011, pp. 3169–3176.

[21] D.-M. Tsai, W.-Y. Chiu, and M.-H. Lee, "Optical flow-motion history image (of-mhi) for action recognition," *Signal, Image and Video Processing*, vol. 9, no. 8, pp. 1897–1906, 2015.

[22] Z. Jiang, V. Rozgic, and S. Adali, "Learning spatiotemporal features for infrared action recognition with 3d convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2017, pp. 115–123.

[23] Y. Liu, Z. Lu, J. Li, C. Yao, and Y. Deng, "Transferable feature representation for visible-to-infrared cross-dataset human action recognition," *Complexity*, vol. 2018, 2018.

[24] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 3551–3558.

[25] J. Imran and B. Raman, "Deep residual infrared action recognition by integrating local and global spatio-temporal cues," *Infrared Physics & Technology*, vol. 102, p. 103014, 2019.

[26] Z. Shou, D. Wang, and S.-F. Chang, "Temporal action localization in untrimmed videos via multi-stage cnns," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 1049–1058.

[27] J. Gao, Z. Yang, K. Chen, C. Sun, and R. Nevatia, "Turn tap: Temporal unit regression network for temporal action proposals," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 3628–3636.

[28] H. Xu, A. Das, and K. Saenko, "R-c3d: Region convolutional 3d network for temporal activity detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5783–5792.

[29] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91–99.

[30] Y. Zhao, Y. Xiong, L. Wang, Z. Wu, X. Tang, and D. Lin, "Temporal action detection with structured segment networks," in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2914–2923.

[31] T. Lin, X. Zhao, H. Su, C. Wang, and M. Yang, "Bsn: Boundary sensitive network for temporal action proposal generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 3–19.

[32] T. Lin, X. Liu, X. Li, E. Ding, and S. Wen, "Bmn: Boundary-matching network for temporal action proposal generation," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3889–3898.

[33] J. Gao, K. Chen, and R. Nevatia, "Ctap: Complementary temporal action proposal generation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 68–83.

[34] Y.-W. Chao, S. Vijayanarasimhan, B. Seybold, D. A. Ross, J. Deng, and R. Sukthankar, "Rethinking the faster r-cnn architecture for temporal action localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1130–1139.

[35] R. Zeng, W. Huang, M. Tan, Y. Rong, P. Zhao, J. Huang, and C. Gan, "Graph convolutional networks for temporal action localization," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 7094–7103.

[36] T. Lin, X. Zhao, and Z. Shou, "Single shot temporal action detection," in *Proceedings of the 25th ACM international conference on Multimedia*, 2017, pp. 988–996.

[37] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[38] F. Long, T. Yao, Z. Qiu, X. Tian, J. Luo, and T. Mei, "Gaussian temporal awareness networks for action localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 344–353.

[39] A. Piergiovanni and M. Ryoo, "Temporal gaussian mixture layer for videos," in *International Conference on Machine Learning*, 2019, pp. 5152–5161.

[40] Y.-G. Jiang, J. Liu, A. R. Zamir, G. Toderici, I. Laptev, M. Shah, and R. Sukthankar, "Thumos challenge: Action recognition with a large number of classes," 2014.

[41] R. Hou, C. Chen, and M. Shah, "Tube convolutional neural network (t-cnn) for action detection in videos," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5822–5831.

[42] J. Huang, N. Li, T. Li, S. Liu, and G. Li, "Spatial-temporal context-aware online action detection and prediction," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2019.

[43] O. Köpüklü, X. Wei, and G. Rigoll, "You only watch once: A unified cnn architecture for real-time spatiotemporal action localization," *arXiv preprint arXiv:1911.06644*, 2019.

[44] J. Pan, S. Chen, Z. Shou, J. Shao, and H. Li, "Actor-context-actor relation network for spatio-temporal action localization," *arXiv preprint arXiv:2006.07976*, 2020.

[45] C.-Y. Wu, C. Feichtenhofer, H. Fan, K. He, P. Krahenbuhl, and R. Girshick, "Long-term feature banks for detailed video understanding," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 284–293.

[46] K. Duarte, Y. Rawat, and M. Shah, "Videocapsulenet: A simplified network for action detection," in *Advances in Neural Information Processing Systems*, 2018, pp. 7610–7619.

[47] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," in *Advances in neural information processing systems*, 2017, pp. 3856–3866.

[48] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[49] D. Ulyanov, A. Vedaldi, and V. Lempitsky, "Instance normalization: The missing ingredient for fast stylization," *arXiv preprint arXiv:1607.08022*, 2016.

[50] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Proceedings of the 27th international conference on machine learning (ICML-10)*, 2010, pp. 807–814.

[51] L. Sevilla-Lara, Y. Liao, F. Güney, V. Jampani, A. Geiger, and M. J. Black, "On the integration of optical flow and action recognition," in *German Conference on Pattern Recognition*. Springer, 2018, pp. 281–297.

[52] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[53] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[54] A. Piergiovanni and M. S. Ryoo, "Learning latent super-events to detect multiple activities in videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5304–5313.

[55] C. Zach, T. Pock, and H. Bischof, "A duality based approach for realtime tv-l 1 optical flow," in *Joint pattern recognition symposium*. Springer, 2007, pp. 214–223.

[56] J. Liu, A. Shahroudy, M. Perez, G. Wang, L.-Y. Duan, and A. C. Kot, "Ntu rgb+d 120: A large-scale benchmark for 3d human activity understanding," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[57] I. Kitware, "The multiview extended video with activities (meva) dataset," https://mevadata.org/ Accessed May 29, 2020.

[58] A. Shahroudy, J. Liu, T.-T. Ng, and G. Wang, "Ntu rgb+d: A large scale dataset for 3d human activity analysis," in *IEEE Conference on Computer Vision and Pattern Recognition*, June 2016.

[59] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*, 2014.

[60] F. Caba Heilbron, V. Escorcia, B. Ghanem, and J. Carlos Niebles, "Activitynet: A large-scale video benchmark for human activity understanding," in *Proceedings of the ieee conference on computer vision and pattern recognition*, 2015, pp. 961–970.

[61] L. Wang, C. Gao, L. Yang, Y. Zhao, W. Zuo, and D. Meng, "Pm-gans: Discriminative representation learning for action recognition using partial-modalities," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 384–401.

[62] T.-W. Hui, X. Tang, and C. C. Loy, "LiteFlowNet: A Lightweight Convolutional Neural Network for Optical Flow Estimation," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018, pp. 8981–8989. [Online]. Available: http://mmlab.ie.cuhk.edu.hk/projects/LiteFlowNet/

[63] X. T. Tak-Wai Hui and C. C. Loy, "A Lightweight Optical Flow CNN - Revisiting Data Fidelity and Regularization," 2020. [Online]. Available: http://mmlab.ie.cuhk.edu.hk/projects/LiteFlowNet/

[64] T.-W. Hui and C. C. Loy, "LiteFlowNet3: Resolving Correspondence Ambiguity for More Accurate Optical Flow Estimation," 2020.