IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 March 2023, accepted 4 April 2023, date of publication 12 April 2023, date of current version 21 April 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3266440

RESEARCH ARTICLE

Campus Abnormal Behavior Recognition With Temporal Segment Transformers

HAI CHUAN LIU[1,2], JOON HUANG CHUAH[1], (Senior Member, IEEE), ANIS SALWA MOHD KHAIRUDDIN[1], XIAN MIN ZHAO[2,3], AND XIAO DAN WANG[4]

[1]Department of Electrical Engineering, Faculty of Engineering, Universiti Malaya, Kuala Lumpur 50603, Malaysia
[2]Chongqing Key Laboratory of Public Big Data Security Technology, Chongqing 401420, China
[3]Chongqing College of Mobile Communication, Chongqing 401520, China
[4]Department of School of Management, Chongqing Open University, Chongqing 400052, China

Corresponding author: Joon Huang Chuah (jhchuah@um.edu.my)

This work involved human subjects or animals in its research. The authors confirm that all human/animal subject research procedures and protocols are exempt from review board approval.

ABSTRACT The intelligent campus surveillance system is beneficial to improve safety in school. Abnormal behavior recognition, a field of action recognition in computer vision, plays an essential role in intelligent surveillance systems. Computer vision has been actively applied to action recognition systems based on Convolutional Neural Networks (CNNs). However, capturing sufficient motion sequence features from videos remains a significant challenge in action recognition. This work explores the challenges of video-based abnormal behavior recognition on campus. In addition, a novel framework is established on long-range temporal video structure modeling and a global sparse uniform sampling strategy that divides a video into three segments of identical durations and uniformly samples each snippet. The proposed method incorporates a consensus of three temporal segment transformers (TST) that globally connects patches and computes self-attention with joint spatiotemporal factorization. The proposed model is developed on the newly created campus abnormal behavior recognition (CABR50) dataset, which contains 50 human abnormal action classes with an average of over 700 clips per class. Experiments show that it is feasible to implement abnormal behavior recognition on campus and that the proposed method is competitive with other peer video recognition in terms of Top-1 and Top-5 recognition accuracy. The results suggest that TST-L+ can improve campus abnormal behavior recognition, corresponding to Top-1 and Top-5 accuracy results of 83.57% and 97.16%, respectively.

INDEX TERMS Action recognition, campus abnormal behavior, computer vision, motion sequence features, temporal segment transformer.

I. INTRODUCTION

Campus abnormal behavior recognition refers to using surveillance devices and artificial intelligence to identify unusual or potentially threatening behavior on campus. Video understanding is a core technology [1], [2], [3], [4] in many scenarios of surveillance systems. Over the years, unexpected actions, such as fighting, accidents, falling, and suicides, have occurred frequently in schools, causing general concern. Recognizing abnormal behavior can achieve real-time and efficient warning, positively affecting school safety management. Researchers focus on directly exploring abnormal behaviors instead of relying heavily on pre-processing to classify video behaviors [5], [6]. Researchers have focused on applications in specific scenarios on campus, such as classrooms [45] and laboratories [46], [47]. However, there is little research on campus abnormal behavior recognition. Essentially, abnormal behavior is a wide range of applications of video understanding. Motivated by video understanding, this study aims to provide an effective solution for recognizing video-based abnormal behavior on campus.

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

Deep learning has become increasingly popular for image recognition [7], [8], [9], [10], [11]. Depending on deep learning, video understanding has emerged in an endless stream to push action recognition to a climax. However, there are challenges in representing temporal video features, mainly focusing on three popular categories: (1) two-dimensional (2D) networks, (2) three-dimensional (3D) networks, and (3) transformers. In the first category, 2D network success represented by two-stream networks [12] pushed video understanding into the deep learning era. The following versions [13], [14], [15] related to two-stream networks emerged within a year, which have a similar network structure. One spatial network branch learns spatial information; the other is an optical flow network representing temporal information. Two-stream networks exhibit superior performance in learning spatial and temporal features separately. Because of the complex optical flow calculation and high storage requirements for pre-processing [12], previous studies are unsuitable for large-scale training and real-time deployment.

Meanwhile, for the second category: 3D networks, video understanding is a 3D tensor composed of two spatial dimensions and one temporal dimension to extract spatial and temporal features. However, optimizing the 3D model requires more work and relies heavily on diverse data than 2D networks [12], [16]. This situation changes when an inflated 3D model (I3D) [17] develops. The I3D operation can inflate ImageNet's pre-trained 2D model to the corresponding 3D model, accelerating optimization. Research related to 3D convolutional neural networks (CNNs) followed the emergence of I3D [18], [19], [20], [21], [22]. The 3D network with natural temporal properties [16] and inflated operation [17], [18], [19] have a competitive effect on video recognition. Consequently, it has long-dominated action recognition.

In the third category, transformers [23], [24], [25], [26] challenge the dominance of CNNs in deep learning and break the barriers of computer vision and natural language processing models. Because of its excellent capabilities in capturing distant information, especially for medium-range and long-range video modeling. Therefore, researchers are interested in applying transformers from the image field to video understanding [27], [28], [29]. However, if an element of self-attention consists of each image pixel, self-attention cannot be directly calculated in a transformer model with relatively high complexity. Hence, a fundamental problem to be solved is reducing the sequence length and designing a self-attention method. Timesformer [27] applied this sequence of frame-level patches with a size of 16×16 pixels instead of every pixel in an image and explored five structures of self-attention. This demonstrates that the divided space-time attention method is faster to train than the 3D CNNs. Applying a transformer in video understanding on the kinetics 400 dataset achieved the best performance compared to CNNs [22] for the first time. However, the transformer has a common issue [23]: it is challenging to learn inductive bias owing to the lack of a large amount of pre-training data, like prior knowledge of the locality and translation equivariance in CNNs. Therefore, researchers have attempted to solve the inductive bias problem of pure transformers [28], [31], [32].

We aim to solve the problem of abnormal campus behavior recognition. Comparing different approaches with the results of the original paper above is challenging because a generic suite is needed to test these types of solutions using the new campus anomalous behavior dataset. Therefore, the proposed models are adequately compared with three relevant proposals: TSN [13], Slowfast [22], and Swin-B [31]. In addition, this work attempts to innovate abnormal behavior identification on campus. First, the backbone network consists of video shifted windows transformer [31], which effectively overcomes the inductive bias problem of the transformer: locality and translation equivariance. It also dramatically resolves the transformer sequence length issue and improves the global modeling ability of models by using a multi-scale shift window to calculate self-attention.

However, the problem with current models is their inability to model an entire video [13], [14]. Since they operate only on a single frame or a stack of frames within a short segment, they have limited access to the temporal context. The complex actions are illustrated in Fig. 1. Abnormal campus actions contain multiple segments with similar redundancies between the consecutive frames of one segment. Failure to use a long-range temporal structure for network training loses the ability to model the entire behavior. Motivated by this early work on video segmentation fusion TSN [13], we designed a campus abnormal behavior recognition framework called temporal segment transformer (TST) to exploit temporal action features and achieve video-level global modeling. Therefore, instead of working on a single frame or stacked frames, TST processes a sequence of snippets globally and is sparsely sampled from the entire video. Each snippet produces its initial class prediction, and a consensus function between the snippets is exported as the final prediction to enable global video dynamic modeling. It can remove redundant information and increase the difference between the behavior classes. Moreover, extensive experiments and discussions support a comparative study of these three methods. Overall, the main contributions of this study are summarized as follows:

- We propose a consensus of three temporal segment transformers (TST) based on the video Swin transformer for the new campus abnormal behavior recognition (CABR50) dataset. It enhances the ability to capture motion sequences and model long-range abnormal behavior on campus.
- We perform extensive comparative experiments with state-of-the-art methods for recognizing abnormal campus behaviors. The results show that it is feasible and can improve the accuracy of abnormal campus behavior recognition. In addition, we demonstrate the performance comparison of the TST and previous methods on

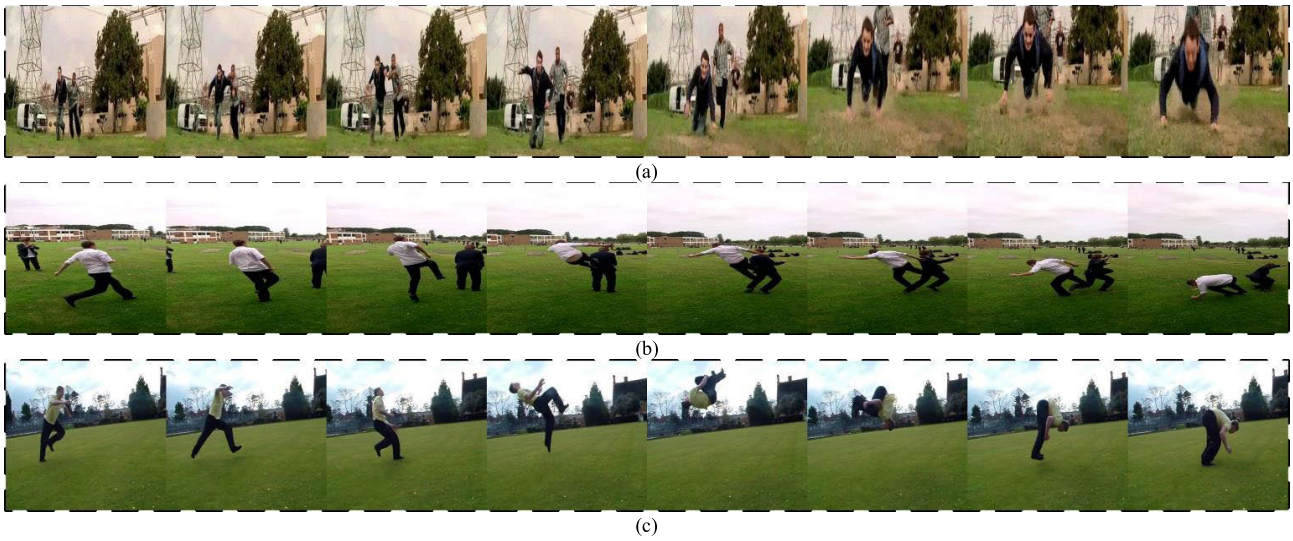


FIGURE 1. Complex and similar behaviors. (a) falling, (b) kicking, (c) backflipping. If only focusing on the first part of the video sequence and ignoring the following sequence, it is easy to confuse the classification of these behaviors.

the UCF-101 dataset. It indicates that our proposed TST model has acceptable generalization performance.

- Our research provides essential technical support for the identification and early warning of abnormal behavior on campus, which plays an essential role in intelligent campus surveillance systems.

In the rest of the paper, Section II summarizes the research methods related to our work. Section III outlines how the TST is employed for campus abnormal behavior recognition and introduces related components. Section IV presents the experimental results and discussion. Section V concludes the work and presents perspectives for future work.

II. RELATED WORKS

The standard solution is to extract features representing abnormal behavior for campus abnormal behavior recognition. For example, Xie et al. [45] used spatiotemporal representations to learn the posture estimation of college students to identify abnormal behavior. They analyzed the behavior of sleeping and using mobile phones in the classroom. Other researchers [46], [47] have explicitly looked at abnormal behavior in the laboratory. Rashmi et al. [46] apply YOLOv3 to locate and recognize student actions in still images from surveillance video in school laboratories. Unlike Rashmi's image recognition-based analysis of students' abnormal behavior in the laboratory, Banerjee et al. [47] used video. They propose a deep convolutional network architecture to detect and classify the behavioral patterns of students and teachers in computer-enabled laboratories. The above works can significantly demonstrate recognition of abnormal behavior in specific scenarios on campus. However, do not aim to reveal the feasibility of numerous abnormal behaviors in multiple scenarios on campus. Although there is limited research on campus aberrant behavior recognition, it is a form of video understanding. The following is a review of video understanding research relevant to our work.

Our methods are motivated by recent works [26], [27], [28], [29], [30], [31], [32] that use self-attention for video classification combined with a convolution operator or a pure transformer. Non-local [33] introduced a self-attention operation, where correlations between any two positions in the feature map are modeled to capture long-range dependencies. It achieved outstanding results on commonly-used video understanding datasets, demonstrating the success of the self-attention operation in long-range modeling interactions between spatiotemporal features. However, the non-local module introduces more parameters and requires a large amount of training data to adjust these parameters. AttentionNAS [34] used the neural architecture search method to automatically search for various self-attention designs and obtain an optimal self-attention operation unit. It can be directly inserted into the existing network and obtains more competitive results than non-local. However, they rely on the hardware's computing power and the search strategy. AttentionNAS designs attention units optimized for specific video classification tasks and may not generalize well to other tasks or domains.

Although self-attention is added as a submodule to CNNs to improve this temporal modeling video understanding, the ability of remote modeling can be made more robust by applying a pure transformer. These algorithms [27], [29], [30] are related to decomposing spatiotemporal self-attention using different factorized methods. They achieved better results than previous pure CNN and methods for adding self-attention units to videos. However, depending on the global attention modeling, these methods lead to a geometric increase in complexity.

Subsequently, MVTs [28] and the video Swin transformer [31] presented the idea of multi-scale hierarchical modeling by calculating the self-attention of multi-scale windows, which is much lower than the computational complexity of global self-attention. Moreover, it exceeds

the previous decomposition space-time modeling methods in terms of accuracy and efficiency. The latest research [32] proposed a multi-view transformer consisting of multiple independent encoders to represent different dimensional input views, fusing information across views through horizontal connections. Although they achieve state-of-the-art performance, their method relies on many unpublic datasets and trained views. It may limit their generalization performance when applied to new videos or tasks beyond their original training scope. Therefore, a fairer comparison is required. In the case of employing ImageNet-21K as pre-training data to initialize network weights, the transformer method established on multi-scale hierarchical modeling [31] has more competitive advantages in video understanding.

Our approach employs a video Swin transformer as the bases of the backbone network, which learns self-attention features with moving multi-scale windows. Its main advantage is a reduced sequence length by making cross-window connections between the upper and lower layers. While the video Swin Transformer excels at processing long-term spatiotemporal dependencies, it may not be the most suitable model for more complex video recognition tasks. Considering the excessive length of the video of abnormal behavior on campus and the similarity of previous action sequences, this work adapted from the opinion of TSN to perform global uniform sparse sampling of the video. Furthermore, this work proposes TST to calculate self-attention by joint spatiotemporally decomposing the entire video. Unlike the video Swin transformer that misclassifies long-range and similar behaviors, our method employs a more efficient temporal segment pure transformer for campus abnormal behavior recognition.

III. PROPOSED METHODOLOGY

This section describes the proposed TST based on the video Swin transformer [31] for abnormal campus behavior recognition. We first employ the essential components and calculation methods for spatiotemporal self-attention. Next, the overall network architecture shows how the backbone network is connected and the forward calculation process. Finally, the configuration of the TST architecture variables is defined to display the relevant variant structures.

A. COMPONENTS

Fig. 4 shows the overall architecture of the proposed TST. These required components consist of video pre-processing, converting the input video into patches, layer normalization [36], accelerating the training convergence of the network, and 3D (shifted) window-based multi-head self-attention operated to calculate self-attention using a sliding window. Thus, there exists a calculation strategy for self-attention. Finally, it presents how the video Swin transformer block connects to the previous components.

1) VIDEO PRE-PROCESSING

The input video clip defines the size $T \times H \times W \times 3$ sampled from the original video, where T denotes the length of the video clip, the frame width and height are represented by W and H, respectively, and 3 denotes the number of channels. Following [31], each patch is treated as a token to solve the video sequence length problem. It is characterized by the concatenation of the raw pixel RGB values, and its dimension is equal to $2\times P\times P\times 3$, P=4. THE input is decomposed into N=$T/2\times H/P\times W/P$ non-overlapping 3D patches spanning the video clip through a 3d patch partitioning layer, where each patch and token consists of a 96-dimensional feature. Then, a linear embedding layer is implemented using CONV3D [35] to project the features per token to an arbitrary dimension represented by C=96.

2) LAYER NORMALIZATION

The function of layer normalization (LN) is to speed up the convergence of the network, control gradient explosion, and prevent the gradient from disappearing. LN normalizes the elements in each sample and computes the layer normalization statistics over all hidden units in the same layer, as follows:

$$\mu = \frac{1}{N}\sum_{i=1}^{N} a_i^l \tag{1}$$

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(a_i^l - \mu^l)^2 + \varepsilon} \tag{2}$$

$$LN(a^l) = \frac{a^l - \mu}{\sigma} \odot \gamma + \beta \tag{3}$$

where μ and σ are the mean and variance of this layer, respectively, and N denotes the number of hidden units in a layer. All hidden units of the identical layer share the normalization terms μ and σ, but separate training cases have specific normalization terms [36]. a is the vector representation of the lth layer. ε refers to adding a smaller value to the variance to prevent division by zero. $\odot$ is the element-wise multiplication between two vectors. γ and β denote the learnable rescaling and the retranslation parameter.

3) 3D (SHIFTED) WINDOW-BASED MULTI-HEAD SELF-ATTENTION

The self-attention of each 3D head is calculated as follows:

$$Attention(Q, K, V) = SoftMax(\frac{QK^T}{\sqrt{d}} + B)V \tag{4}$$

where Q, K, and $V \in \mathbb{R}^{PM^2\times d}$ are the query, key, and value matrices, respectively, and d is the query and key feature dimensions. Moreover, PM^2 denotes the number of tokens in the 3D window [31]. B is learnable relative positional encoding. Note that the above process of computing attention only once is called head self-attention. Multi-head self-attention pays attention to the same Q, K, and V multiple times, obtaining multiple different outputs and then connecting these different outputs to a final output.

The 3D window-based multi-head self-attention (3D W-MSA) has the additional concept of a 3D window

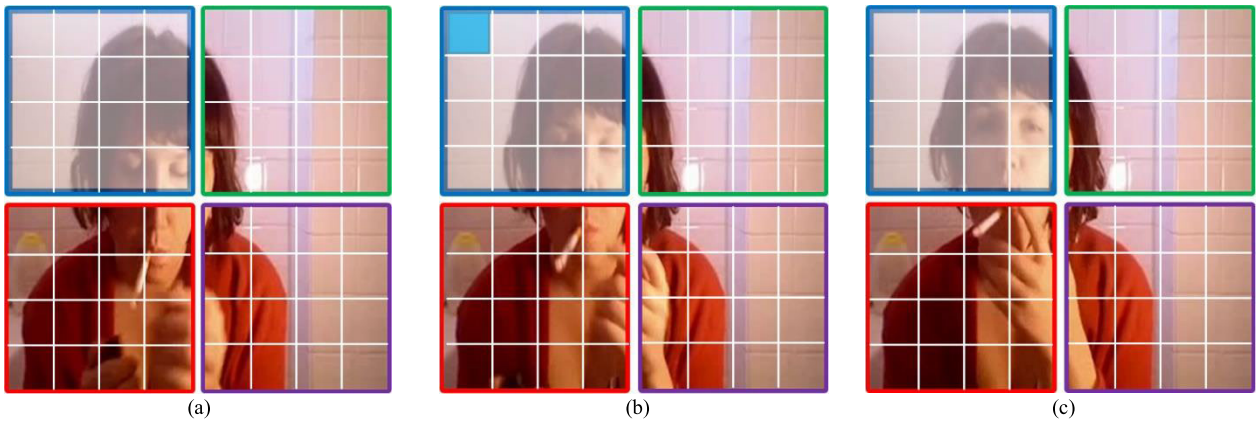


FIGURE 2. Visualization of the joint spatiotemporal attention studied in TST. For illustration, enumerate three consecutive frames, (a) frame t-1, (b) frame t, (c) frame $t+1$. Each frame is divided into four windows, each with sixteen patches to be distinguished by different colors. Use blue to represent a query patch, and shadowy shows how its self-attention spatiotemporal neighborhood is computed. Note that only compute self-attention with patches in windows with the same color; although the attention pattern is shown for only two adjacent frames, it extends in the same fashion to all frames of the clip.

compared with MSA. Its function is to reduce the computational complexity such that it is linearly related to the input image size [25]. Compute self-attention within a 3D local window, divide the clip input in a non-overlapping manner evenly, supposing the input 3D tokens with $T' \times H' \times W'$ and a 3D window size of $P \times M \times M$. Input tokens split into $\lceil T'/P \rceil \times \lceil H'/\mathrm{M} \rceil \times \lceil W'/M \rceil$ non-overlapping 3D windows. Because a multi-head self-attention mechanism is used per non-overlapping 3D window, a lack of connectivity between separate windows may restrict the representational capability of the model. Therefore, it adapts 3D-shifted window-based multi-head self-attention (3D SW-MSA). The window partition configuration is shifted along the temporal, height, and width axes by (P/2, M/2, M/2) tokens from that of the self-attention module of the preceding layer.

4) JOINT SPATIOTEMPORAL SELF-ATTENTION

There exist many related methods [27], [28], [29], [30], [31], [32], [44] for designing spatiotemporal self-attention. Motivated by [27], a design based on joint space-time self-attention is the best, adopting the joint spatiotemporal attention at each 3D window-based MSA layer. Fig. 2 shows blue as the query patch for joint spatiotemporal attention schemes. Self-attention must be paid to all patches in the same window between three consecutive frames. Therefore, the features learned through self-attention include spatial and temporal information. In addition, self-attention is computed based on a shifting window, and the complexity of the joint spatiotemporal attention approach is much lower than that of computing an entire image [27]. Each patch in the clip must calculate self-attention.

5) VIDEO SWIN TRANSFORMER BLOCK

As shown in Fig. 3, the backbone network mainly consists of two consecutive video Swin transformer blocks [31]. Specifically, the left block comprises a conventional 3D window-based MSA module (3D W-MSA) and feedforward network (FFN). The other block consists of only a shift-operated 3D shifted window MSA module (3D SW-MSA) followed by an FFN, which consists of two layers of MLP with the activation function GELU. LN is applied before each MSA module and FFN, and a residual connection is applied after each module. Therefore, each stage is composed of an even number. Two layers of blocks are required to be connected as a basic unit. Finally, calculating the self-attention between different windows makes the learned features multi-scale hierarchical.

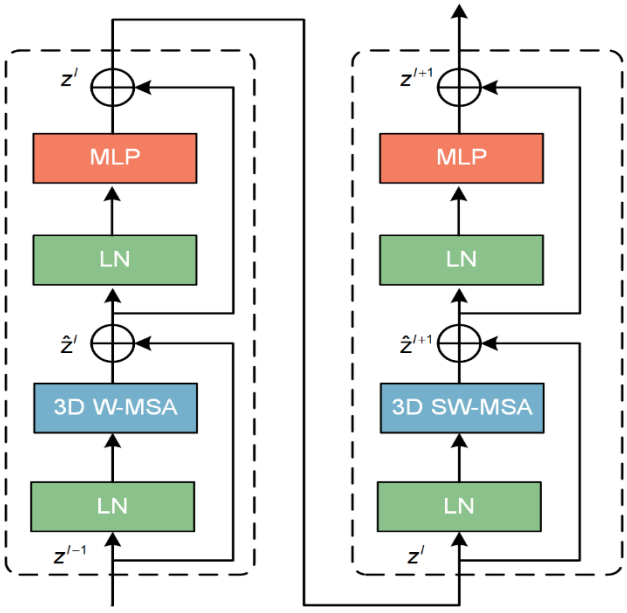


FIGURE 3. Video Swin transformer block with 3D window-based MSA module and 3D shifted window MSA module.

Two consecutive video Swin transformer blocks are computed as:

$$\hat{z}^l = 3DW - MSA(LN(z^{l-1})) + z^{l-1} \quad (5)$$

$$z^l = FFN(LN(\hat{z}^l)) + \hat{z}^l \quad (6)$$

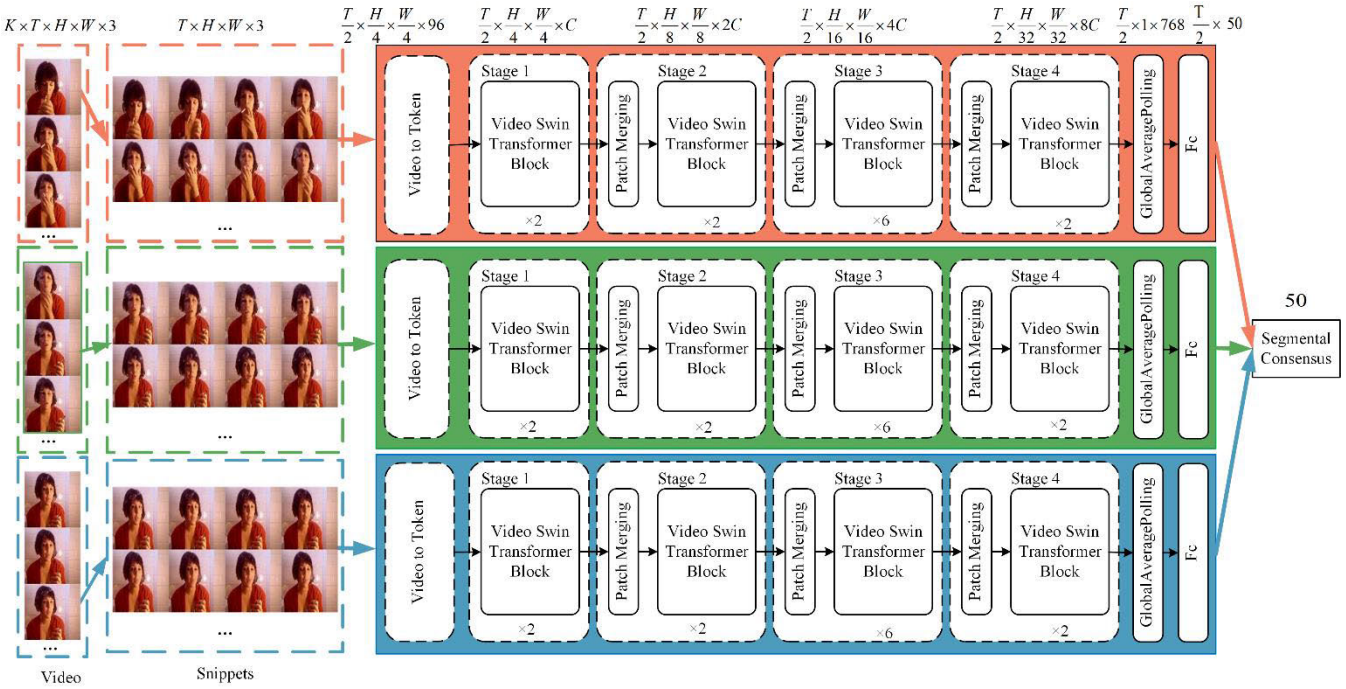


FIGURE 4. The overall architecture of the proposed temporal segment transformer network. The first column splits an input video into K segments and then uniformly selects a short snippet from each segment. A segmental consensus function fuses the class prediction of different segments to produce the final prediction. TST shares parameters for all fragments.

$$\hat{z}^{l+1} = 3DSW - MSA(LN(z^l)) + z^l \tag{7}$$

$$z^{l+1} = FFN(LN(\hat{z}^{l+1})) + \hat{z}^{l+1} \tag{8}$$

where $\hat{z}^l$ denotes the output features of the 3D (S)W-MSA module, z^l is the FFN module for block l. 3D W-MSA and 3D SW-MSA denote 3D window-based multi-head self-attention using standard and shifted window partitioning configurations, respectively.

B. OVERALL ARCHITECTURE OF THE PROPOSED WORK

Fig. 4 illustrates the overall architecture of the proposed TST. We apply a global sparse uniform sampling strategy. Given an input video, split it into K segments $\{S_1, S_2, \ldots, S_K,\}$ of identical durations from the original video, defined to be of size $K \times T \times H \times W \times 3$. Here, $(T_1, T_2, \ldots, T_K)$ is a sequence of snippets, and each snippet T_K is uniformly sampled from its related segment S_K, represented as $T \times H \times W \times 3$. The snippets input the backbone network. Note that the models share parameters for all the fragments. For example, assume that $K = 3$, $T = 32$, $H = 224$, $W = 224$, and the following forward process calculation of TST.

First, the frame is divided into patches using the 3D patch partition method. Because video is used as input, it increases the time dimension compared with the image. The 3D patch of size $2 \times 4 \times 4 \times 3$, and get $T/2 \times H/4 \times W/4$ 3D tokens, the input of size $16 \times 56 \times 56$, each patch/token contains 96-dimensional features. The following linear embedding layer modifies the dimension of the vector to a value: the hyperparameter C for the TST, $C = 96$. Through the linear embedding layer, the input size becomes $16 \times 56 \times 56 \times 96$. The input and output dimensions of the transformer block maintain consistency; therefore, through the two layers of Swin transformer blocks in Stage 1, the output is still $16 \times 56 \times 56 \times 96$.

Subsequently, Patch Merging is applied to construct a hierarchical transformer that obtains multi-scale features. It works like max pooling in CNNs, which connects the features of each 2×2 spatial neighbor patch and employs a linear layer to project concatenated features to half its dimensions. It is equivalent to projecting the $4C$ dimensions of a token onto $2C$-dimensional features. The concept of this model is like that of ResNet [10]. While deepening the model, it extracts advanced visual features and reduces computational complexity. Note that the temporal dimension remains consistent. In Stage 2, the output size is $16 \times 28 \times 28 \times 192$; Stage 2, Stage 3, and Stage 4 share the same structure, and the outputs of Stage 3 and Stage 4 are $16 \times 14 \times 14 \times 384$ and $16 \times 7 \times 7 \times 768$, respectively. Subsequently, by applying the global average pooling layer, the output size becomes $16 \times 1 \times 768$. The final step fuses sixteen frames in each snippet and obtains classification results for each segment through dropout and the output of fully connected layers. The specific calculation process is as follows.

$$TST(T_1, T_2, \ldots, T_K) = P(g(F(T_1; W), F(T_2; W), \ldots, F(T_K; W))) \tag{9}$$

$F(T_K; W)$ is a function showing a transformer that achieves the short snippet T_K and generates initial class scores for all

TABLE 1. Detailed Architecture Configurations.

Stage	TST-T		TST-H		TST-L		Output size
Stage1	$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=96 & head=3 \end{bmatrix}\times2$	$\times$ 3	$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=96 & head=3 \end{bmatrix}\times2$	$\times$ 3	$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=128 & head=4 \end{bmatrix}\times2$	$\times$ 3	$16\times(56\times56)$
Stage2	$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=192 & head=6 \end{bmatrix}\times2$		$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=192 & head=6 \end{bmatrix}\times2$		$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=256 & head=8 \end{bmatrix}\times2$		$16\times(28\times28)$
Stage3	$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=384 & head=12 \end{bmatrix}\times6$		$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=384 & head=12 \end{bmatrix}\times18$		$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=512 & head=16 \end{bmatrix}\times18$		$16\times(14\times14)$
Stage4	$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=768 & head=24 \end{bmatrix}\times2$		$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=768 & head=24 \end{bmatrix}\times2$		$\begin{bmatrix} win-size= & 8\times7\times7 \\ dim=1024 & head=32 \end{bmatrix}\times2$		$16\times(7\times7)$

classes of actions. Segmental consensus function g produces a class hypothesis consensus among K short segments by combining the outputs from these snippets. The prediction function P predicts the probability of each category for the entire video based on this consensus. Here, the widely used SoftMax function for P. Combined with the standard categorical cross-entropy loss, the final loss function regarding the segmental consensus $G = g(F(T_1; W), F(T_2; W),\ldots, F(T_K; W))$ is as follows:

$$L(\mathrm{y}, G) = -\sum_{i=1}^{n} y_i(G_i - \log \sum_{j=1}^{n} \exp G_j) \qquad (10)$$

n and y_i represent the number of categories and ground-truth labels corresponding to class i. The class score G_i is calculated from scores of the same class on all the snippets using an aggregation function g. Following previous work [13], this work chose the weighted averaging method with a better effect.

C. ARCHITECTURE VARIANTS

Three versions of TST are produced to explore the model size and depth. Table 1 lists the detailed configurations, where the input video size is assumed to be $32 \times 224\times224$ for all architectures. The "win-size=$8 \times 7\times7$" represents a multi-head self-attention module with a temporal window size of $8\times7\times7$. The "dim" and "head" denote the output dimension and the number of heads in multi-head self-attention. The "output size" indicates the size of each stage. Finally, the architecture hyper-parameters of these model variants:

TST-T(tiny):$K=3$, $C=96$, layer numbers$=\{2, 2, 6, 2\}$
TST-H(huge):$K=3$, $C=96$, layer numbers$=\{2, 2, 18, 2\}$
TST-L(large):$K=3$, $C=128$, layer numbers$=\{2, 2, 18, 2\}$

where K shows the number of segments, and C represents the number of channels in the hidden layer in the first stage. The window size default is $P = 8$ and $M = 7$. The query dimension of each head is $d = 32$, and the expansion layer of each MLP is $\alpha = 4$.

TABLE 2. Summary of Characteristics of Cabr50.

Characteristics	Value
Actions	50
Clips	36,968
Mean Clip Length	9.30 s
Max Clip Length	88.96 s
Frame Rate	25 to 30 fps
Resolution	340*256

IV. EXPERIMENTS

In this section, experiments are performed to evaluate the proposed approach. We introduce the CABR50 dataset, evaluation protocol, and model settings and then present model configuration studies of the TST. Furthermore, performance comparisons between TST and other state-of-the-art methods are presented on the CABR50 and UCF-101. Finally, we discuss the separability and analyze the results.

A. DATASET AND IMPLEMENTATION DETAILS

1) DATASET

The experiments are conducted on the self-proposed campus abnormal behavior recognition dataset: CABR50, which contains 50 categories and 36,968 video clips of abnormal behaviors on different campus scenes. These data were collected mainly from YouTube, Baidu, and public datasets [38], [41]. We used OpenCV to read the collected video data and see if it opened correctly. Irrelevant videos were removed. The part of the video that indicates abnormal behavior was cropped. The resolution of the video was normalized to 340∗256. Table 2 summarizes the characteristics of the datasets. Indoor scenes include classrooms, laboratories, study rooms, libraries, student residences, student canteens, and lifts. Outdoor scenes refer to the outdoor public areas, corridors and aisles, and the top floors of buildings. Fig. 5(a) uses a bar chart to illustrate the number of clips for every abnormal behavior, where the color on each bar shows the duration of different

TABLE 3. The Division Information of the Cabr50.

Version	Train	Test	Total
Split1	24,382	12,586	36,968
Split2	24,764	12,204	36,968
Split3	24,790	12,178	36,968

clips contained in that class. A potential characteristic is that most action clips beyond 10 s are long-range videos [37].

The chart shown in Fig. 5(b) displays the average clip length (orange) and the total duration of the clips (blue) for each abnormal behavior. The total video duration exceeds 6,000 seconds for most classes, except for a few: falling, fighting, and standing. The problem of data imbalance is common in abnormal behavior detection and identification. This paper uses a resampling approach to reduce the impact of data imbalance. For a fairer comparison, the experiment follows the evaluation scheme of the industry standard for video datasets [37], [38], which adapts the three train/test splits for cross-evaluation. This paper adds the constraint of separating videos from the same group during training and testing. Because some videos in the group are obtained from a single long video, sharing videos from the same group in the training and test sets provide high performance. Table 3 lists the split video data.

2) EVALUATION PROTOCOL

The unified evaluation standard is employed in behavior recognition to evaluate the model's performance and apply the recognition accuracy of the test dataset to evaluate the performance. Top-1 and Top-5 recognition accuracy are commonly used to characterize classification performance. Top-1 accuracy indicates that the model prediction with the highest probability must be the exact expected answer. Top-5 accuracy means that any of our model's top five highest probability answers match the expected answer. A confusion matrix is used to present the performance for each category. Each column of the confusion matrix represents the predicted category, and the total number of each column indicates the number of predicted as this category. Each row represents the actual category, and the sum in each row indicates the number of data instances in this category. The model recall rate for each category analyzes the categories that are difficult to distinguish.

3) MODEL SETTING AND IMPLEMENTATION DETAILS

For CABR50, TST applies an AdamW [39] optimizer for 20 epochs and uses the cosine decay learn rate scheduler and 2.5 epochs of linear warm-up, setting the batch size to 16. Unless otherwise noted, the entire video is evenly divided into three segments, sampled 32-frame from each short segment, a temporal stride of 2, and a spatial size of 224 × 224. Following previous work [25], the increasing stochastic depth and weight decay for better models [40], such as 0.1, 0.2, and 0.3 stochastic depth rates, 0.02, 0.02, and 0.05 weight decay for TST, TST-H, and TST-L, respectively. For each clip, experiments use three crops (top-left, center, bottom-right) from one segment and obtain the prediction by averaging the scores for these three crops. The results of these three segments are then averaged. Unless otherwise noted, the final score is the average prediction of the three segments.

TABLE 4. The Effective of the Temporal Dimension of 3D Tokens and Temporal Window Size with Backbone Network on the First Split of Cabr50.

Temporal Dimension	Window Size	Top-1	Top-5	FLOPs	Param
16	16×7×7	73.55	92.42	106	28.3
8	8×7×7	72.36	91.90	44	27.9
4	4×7×7	68.89	90.35	20	27.7
16	16×7×7	73.55	92.42	106	28.3
16	**8×7×7**	**75.73**	**93.67**	**88**	**27.9**
16	4×7×7	74.33	92.98	79	27.7

In this paper, the experiments are performed using the same gpushare.com cloud node on a system with an AMD EPYC 7302 CPU and an NVIDIA GeForce RTX 3090 GPU. The node system runs on Ubuntu 20.04.3 LTS with Linux kernel 5.4.0-91-generic. This work develops the algorithm using Python 3.8.10 with the PyTorch: 1.7.1+cu110 framework.

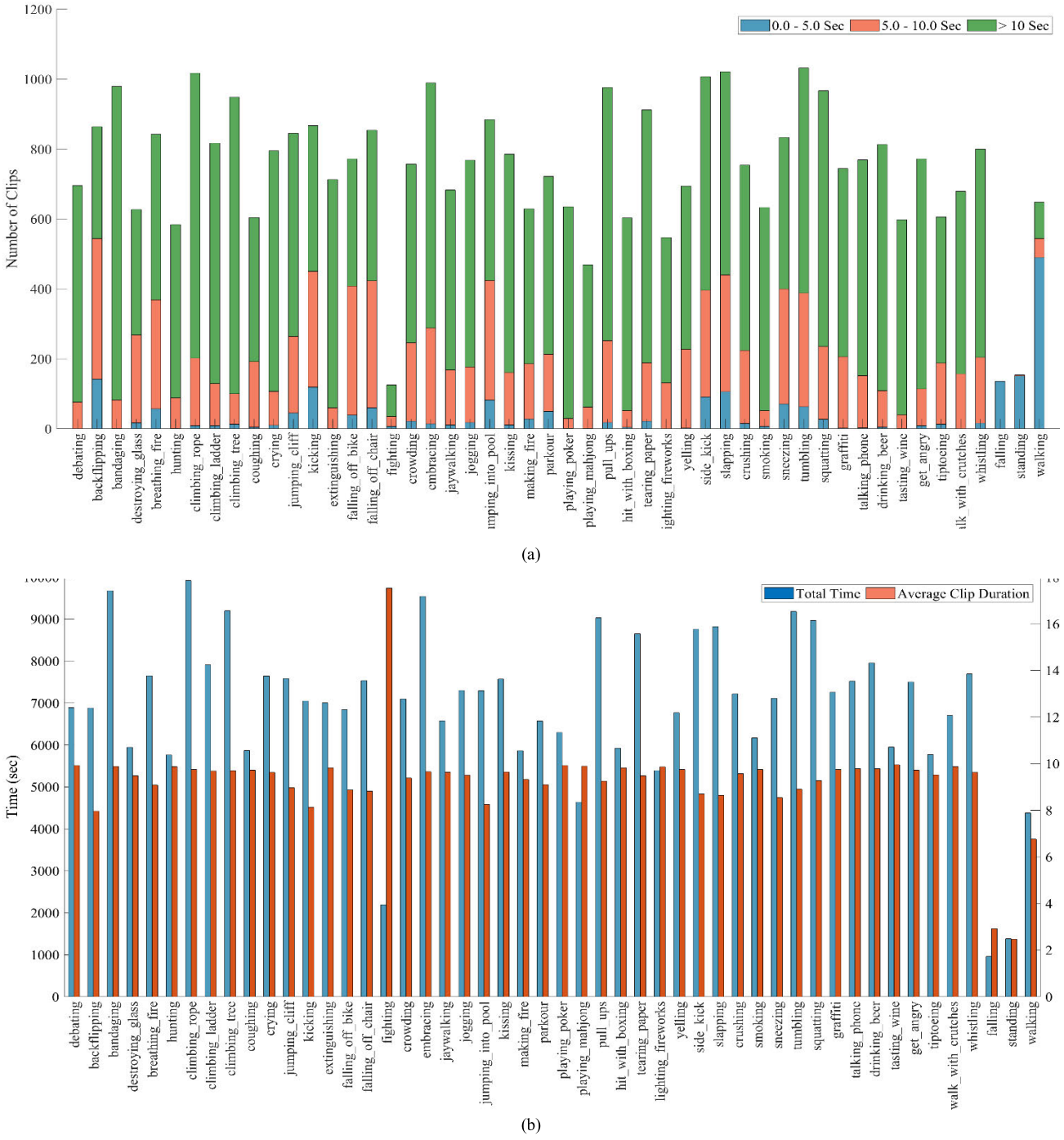
B. MODEL CONFIGURATION STUDIES

1) THE EFFECT OF THE TEMPORAL DIMENSION OF 3D TOKENS AND TEMPORAL WINDOW SIZE

Although the backbone network can significantly learn the Kinetics Human Action Video Dataset [41], the effect on the CABR50 dataset depends on the temporal dimension of the 3D tokens and the temporal window size. Therefore, extensive experiments are performed to choose the combination of the temporal dimension and temporal window size to perfectly match the CABR50 dataset.

Experiment is conducted on the temporal dimension of 3D tokens on a global temporal scale and sets the temporal dimension of the 3D tokens equal to the temporal window size. The results for the backbone network on CABR50 are listed in the upper part of Table 4. This indicates that a larger temporal dimension prompts higher top-1 and top-5 accuracy, which can learn more temporal sequence features with higher corresponding computational costs and a slower inference speed. Hence, the temporal dimension of the 3D tokens is sixteen.

Subsequently, experiments with temporal window sizes of 4/8/16 are completed, fixing the temporal dimension of the 3D tokens to 16. The lower part of Table 4 shows the results for the backbone network of CABR50. It can be observed that the performance is better for a smaller temporal window size than for a temporal window size of 16. Compared to the backbone models with temporal window sizes of 8 and 4, the computation amount increased by 11.39% (88 vs. 79), and the accuracy increased by 1.4%. Therefore, considering the



(a)

(b)

FIGURE 5. **(a) The number of clips per abnormal behavior. A stacked bar chart is used to show three distributions of clip durations and illustrate them with different colors. (b) the total time of videos for each class is illustrated by applying the blue bars. Orange bars represent the average length of the clips for each class.**

trade-off between accuracy and complexity, the best combination is 3D tokens with a temporal dimension of 16 and a temporal window size of $8 \times 7 \times 7$.

2) THE EFFECT OF THE NUMBER OF SEGMENTS ON THE TST

As described in the Introduction, abnormal campus actions include multiple temporal segments, and different classes have similar redundancies between consecutive frames of one segment. The experiments are performed by applying multiple segments instead of the clip as the input and exploring the number of more suitable segments for CABR50. As shown in Table 5, TST with two segments achieved 2.18% better performance compared to TST with one in Top-1 accuracy. With the increasing number of segments, the TST with three, four,

TABLE 5. The Effective of the Number of Segments with TST on the First Split of Cabr50.

Number of Segments	Top-1	Top-5	FLOPs	Param
1	75.73	93.67	88	27.9
2	77.91	94.80	176	27.9
3	**78.82**	**95.22**	**264**	**27.9**
4	78.85	95.35	352	27.9
5	78.90	95.26	440	27.9

TABLE 6. Comparing the Effects of Pre-training Datasets on the First Split of Cabr50.

Method	Pre-training	Top-1	Top-5	FLOPs	Param
TST-T	ImageNet-1K	78.82	95.22	264	27.9
TST-H	ImageNet-1K	79.99	95.43	443	49.6
TST-L	ImageNet-1K	81.19	96.16	786	87.7
TST-L+	**ImageNet-21K**	**83.60**	**97.11**	**786**	**87.7**

and five segments gain 3.09%, 3.12%, and 3.17%, respectively. In general, the accuracy is dramatically improving with a more significant number of segments in the TST.

Furthermore, it probes reasonable values for K. The results indicate that the three-segment model for CABR50 is better in terms of accuracy and complexity. The Top-1 accuracy of the four and five segments improved by 0.03% and 0.08%, respectively. However, computation increased significantly by 33% and 66%, respectively. Therefore, given the trade-off between accuracy and complexity, three are the optimal segmentation values in the TST network.

3) THE EFFECT OF DIFFERENT PRE-TRAINING DATASETS ON TST

Training TST from scratch requires much time to converge, owing to the many TST parameters. Therefore, this work refers to the common practice in deep learning and initializes it with the weights learned from the pre-training data. We evaluate the results of ImageNet-1K and ImageNet-21K as pre-training initialization weights for CABR50. These experiments are conducted in three TST variants, and the specific design details are described in Section III. Table 6 shows that TST-T with ImageNet-1K as the pre-training weight reaches 78.82% in Top-1 accuracy, with the same pre-training data TST-H and TST-L achieving higher values of 79.99% and 81.19%, respectively. ImageNet-21K is then used for pre-training. It can achieve a performance improvement of 2.41% over TST-L pre-trained with ImageNet-1K. It is verified again that the larger the dataset size, the better the initialization effect of the model.

C. COMPARATIVE STUDY

Comparative experiments facilitate finding the most suitable algorithm for abnormal behavior recognition on campus. The current representative methods in the video domain are evaluated for their performance. Table 7 summarizes the average results for the CABR50. It substantially compares the most exemplary methods in three directions: the 2-D network represented by TSN, Slowfast related 3-D network, and the transformer represented by Swin-B. For a fair comparison, unless otherwise noted, all the above models use ImageNet-1K as a pre-training to initialize the network weights. TSN and Slowfast obtain 69.27% and 68.85%, respectively, in terms of Top-1 accuracy. The state-of-the-art transformer represented by Swin-B achieves 76.65%, which is significantly better than those of the previous two models. The baseline model TST-T improved better than Swin-B, and the Top-1 and Top-5 accuracy increased by 1.79% and 1.26%, respectively. The complexity of the baseline model is slightly lower compared to Swin-B. With the same pre-training dataset, TST-H and TST-L gain 2.85% and 5.29% Top-1 accuracy and a corresponding significant complexity enhancement over the previous algorithms.

More importantly, for TST-L+, the ImageNet-21K pre-training results in a 1.63% growth over training on ImageNet-1K. TST-L+ obtained state-of-the-art results of 83.57% and 97.16% in terms of Top-1 and Top-5 accuracy, respectively. Furthermore, experiments with the above methods effectively demonstrate the feasibility of identifying abnormal campus behaviors on CABR50, showing that the created abnormal behavior datasets are separable and effective. Therefore, the proposed TST model can improve the performance of abnormal campus behavior recognition.

We show the performance comparison of the TST model and previous methods on the UCF-101 dataset in Table 8. One can see that our proposed TST performs well on the UCF-101 dataset. The baseline model TST-T achieves an accuracy of 96.13%, a relatively promising result. However, the performance of TST-T is slightly weaker than I3D [17]. Compared with I3D, the pre-processing work is reduced. That is, the extraction of optical flow is avoided. Our variant model TST-L+ achieves a TOP-1 accuracy of 98.20%, which is also a competitive result. In conclusion, one can see from the experiments that our proposed model performs exceptionally well on CABR50 and has competitive results on UCF-101. Therefore, our TST model has acceptable generalization performance.

D. DISCUSSION

1) SEPARABILITY

Abnormal campus behavior usually includes the following characteristics: context-dependent, unpredictable, sudden, short-term temporal, local spatiotemporal, and global consistency [42]. Therefore, it is essential to determine whether the proposed model can extract useful information and classify it accurately. To intuitively illustrate the separability of the proposed approach, the TST-L+ model is used to visualize the first split features of CABR50. The results are illustrated in Fig. 6. A point represents a video sample. Videos in the identical action category have the same color, and the category is marked near them. In this study, a T-distributed

TABLE 7. Comparative Study on the Cabr50 (over three splits).

Method	Pretraining	Split1		Split2		Split3		Average		FLOPs	Param
		Top-1	Top-5	Top-1	Top-5	Top-1	Top-5	Top-1	Top-5		
TSN [13]	ImageNet-1K	71.08	91.67	68.54	91.15	68.18	90.95	69.27	91.26	31	23.6
Slowfast [22]	ImageNet-1K	69.23	90.11	69.12	90.18	68.21	90.39	68.85	90.23	36	33.7
Swin-B [31]	ImageNet-1K	77.36	93.98	76.28	93.91	76.30	94.06	76.65	93.98	282	87.0
TST-T	ImageNet-1K	78.82	95.22	78.34	95.33	78.17	95.18	78.44	95.24	264	27.9
TST-H	ImageNet-1K	79.99	95.43	79.61	95.61	78.89	95.07	79.50	95.37	443	49.6
TST-L	ImageNet-1K	81.19	96.16	82.35	96.73	82.29	97.75	81.94	96.88	786	87.7
TST-L+	**ImageNet-21K**	**83.60**	**97.11**	**83.51**	**97.25**	**83.59**	**97.13**	**83.57**	**97.16**	**786**	**87.7**

TABLE 8. A Comparative Study on the UCF-101 Datasets, averaged over three splits.

Model	Top-1	Top-5
TSN [13]	94.20	/
I3D [17]	97.80	/
TST-T	96.13	99.43
TST-H	96.86	99.65
TST-L	97.36	99.82
TST-L+	**98.20**	**99.87**

Stochastic Neighbor Embedding (T-SNE) [43] visualized algorithm is used to reduce the dimensionality of the features and project the high-dimensional representation into a two-dimensional space. Nearby points model similar samples, and distant points with high probability model dissimilar samples.

Fig. 6 shows that a small part of the classification results of standing and falling, crushing, and destroying-glass coincide. Another evident phenomenon is that get-angry, crying, sneezing, coughing, yelling, debating, and slapping are closely distributed. Because these abnormal behaviors have highly similar sequences in local spatiotemporal and global consistency, their 2D-dimensional visualization looks tight, as shown in Fig. 6. However, they can be distinguished. Based on this illustration, other actions display a strong classification effect because their features appear to aggregate within the class and are separated between classes.

Therefore, separability indicates that our model can learn intra-class similarity and inter-class difference features from the campus abnormal behavior dataset. Through training, it carries out strong abnormal behavior information representation, which provides a powerful tool for campus safety supervision.

2) RESULTS ANALYSIS

To analyze these hard-to-recognize abnormal behaviors on campus, we visualize the confusion matrix and per-class recall for the CABR50 classification. The left plot in Fig. 7 shows the confusion matrix obtained using the TST-L+ model. It is observed that the proposed TST-L+ fits CABR50 well. However, some classes have smaller values.

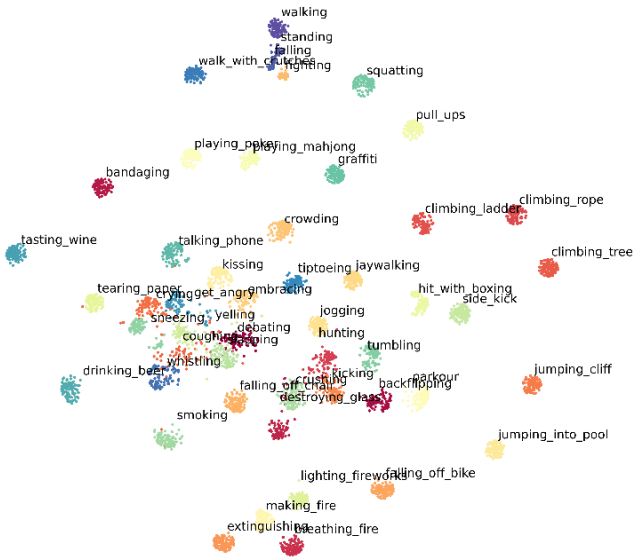


FIGURE 6. Feature visualization of TST-L+ with T-SNE on the first split of CABR50.

Then, the per-class recall is visualized in Fig. 7(right plot) and separately analyzed classes related to recall less than 0.5. The worst class recall corresponds to the coughing class, which is confused with the sneezing class for two possible reasons. First, from the perspective of the spatial dimension, it may be caused by the existence of faces in both classes. Second, from the temporal dimension, both actions contain identical motion patterns (mouth opening and closing, hand movement).

In addition, there are poor performance categories: yelling and debating, confusing each other. The yelling category is misjudged to be slapping or debating. The debating category is misclassified as slapping. The first reason may be that these classes have a significant relationship with human expressions and numbers in the spatial dimension. Another possible reason is the continuous behavioral motion (expression changes from normal to surprised and angry).

In order to intuitively observe the performance of the comparison algorithms used in each category of the three split datasets. Recall curves are drawn by taking the mean value of each category of the three split datasets, as shown in Fig. 8. It uses a stacked bar chart to illustrate the average recall of the

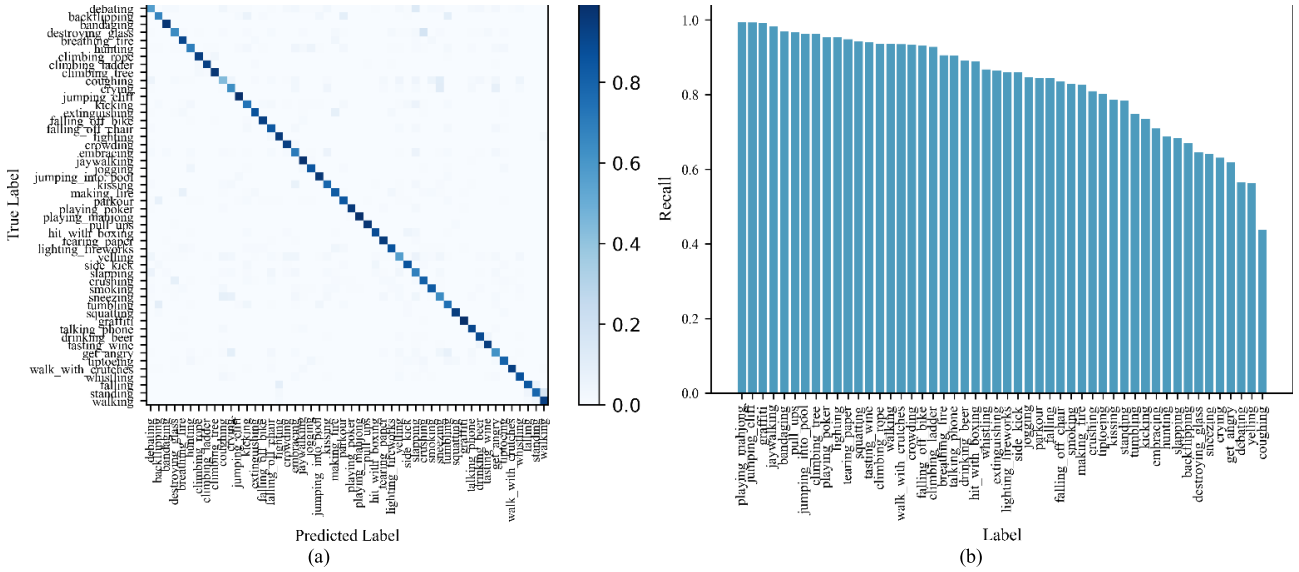


FIGURE 7. Confusion matrix and per-class recall of TST-L+ on the first split of CABR50. (a) the confusion matrix results, (b) the per-class recall results, and the results in ascending order.

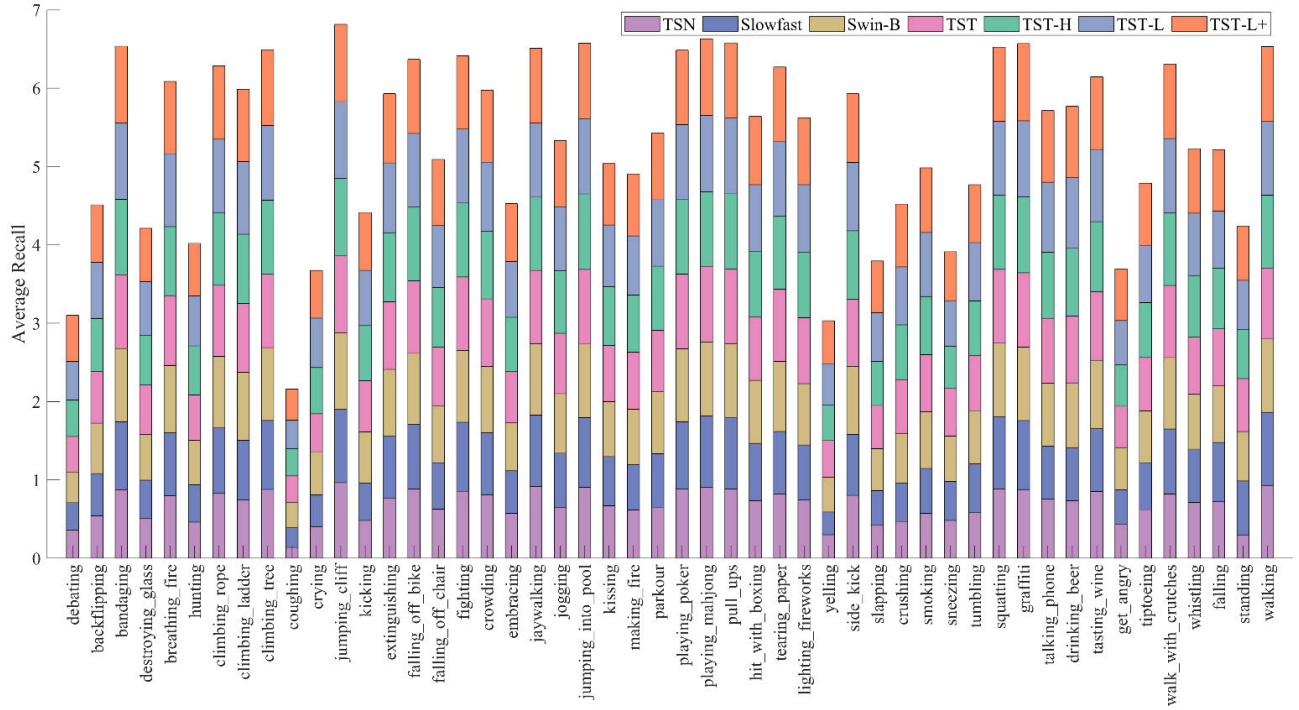


FIGURE 8. Per-class average recall on the three splits of CABR50.

seven algorithms on three split datasets. Each color represents every algorithm, and its values range from zero to seven. This value indicates the overall recognition performance of a category. The overall performance of the proposed TST in most categories is more robust than that of the other models in Table 7, demonstrating the effectiveness and separability of the proposed TST models.

In addition, one can observe that coughing, debating, and yelling performs poorly for each algorithm, which hardly means that the TST does not work. Because the nature of these categories is challenging to recognize, these data are similar to the hard classes defined in the UCF-101 [37]. However, the performance of the proposed TST is significantly enhanced in the hard classes, indicating the effective mitigation of hard classes in CABR50. In the future, the project needs to study the effective representation of the features of the hard categories. Further recognizing these categories is also conducive to improving the performance of intelligent campus surveillance systems.

V. CONCLUSION

Abnormal behavior recognition plays a significant role in intelligent campus surveillance systems. In this study, a CABR50 dataset was created, and a framework named temporal segment transformers was proposed to address the problem of identifying abnormal behavior on campus. Specifically, TST divides the input video into three segments of equal duration from the original video and obtains snippets uniformly sampled from its segment. These snippets are used as the inputs for the backbone network. Each snippet produces its initial prediction of the class, followed by a consensus function between the snippets exported as the final prediction, enabling dynamic global video modeling. Extensive experiments are carried out on the three split CABR50 datasets to verify the classification accuracy: TSN, Slowfast, Swin-B, and TST methods. In addition, the superiority of the proposed method in classifying abnormal behaviors on campus is verified in terms of the analysis result. To explore the model's generalization performance, we experimented with it on the UCF-101 dataset and achieved promising results.

In summary, this work demonstrates the feasibility of using abnormal campus behavior recognition. In addition, our proposed TST can effectively model long-range behavior and achieve competitive results on CABR50. However, the model should perform better on the abnormal campus behavior categories of coughing, debating, and yelling that belong to hard data. In the future, we will try to combine multimodal approaches, such as extracting audio features from videos, to assist in classifying hard data for abnormal campus behavior. In addition, we can take an efficient approach to reduce the complexity of the model and overcome the problem of imbalance in the corresponding category data, such as GAN, which generates new training data for unbalanced categories.

ACKNOWLEDGMENT

The authors would like to thank the editors and reviewers for their valuable comments and suggestions. They are also grateful to gpushare.com for providing arithmetic support.

REFERENCES

[1] Q. Hao and L. Qin, "The design of intelligent transportation video processing system in big data environment," *IEEE Access*, vol. 8, pp. 13769–13780, 2020.

[2] K. Muhammad, J. Ahmad, Z. Lv, P. Bellavista, P. Yang, and S. W. Baik, "Efficient deep CNN-based fire detection and localization in video surveillance applications," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 7, pp. 1419–1434, Jul. 2019.

[3] L. Simoni, A. Scarton, C. Macchi, F. Gori, G. Pasquini, and S. Pogliaghi, "Quantitative and qualitative running gait analysis through an innovative video-based approach," *Sensors*, vol. 21, no. 9, p. 2977, Apr. 2021.

[4] M. Shorfuzzaman, M. S. Hossain, and M. F. Alhamid, "Towards the sustainable development of smart cities through mass video surveillance: A response to the COVID-19 pandemic," *Sustain. Cities Soc.*, vol. 64, Jan. 2021, Art. no. 102582.

[5] A. B. Mabrouk and E. Zagrouba, "Abnormal behavior recognition for intelligent video surveillance systems: A review," *Expert Syst. Appl.*, vol. 91, pp. 480–491, Jan. 2018.

[6] A. B. Tanfous, H. Drira, and B. B. Amor, "Sparse coding of shape trajectories for facial expression and action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 10, pp. 2594–2607, Oct. 2020.

[7] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. NIPS*, Dec. 2012, pp. 1097–1105.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 1–9.

[9] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

[10] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[11] H. Zhang, C. Wu, Z. Zhang, Y. Zhu, H. Lin, Z. Zhang, Y. Sun, T. He, J. Mueller, R. Manmatha, M. Li, and A. Smola, "ResNeSt: Split-attention networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2022, pp. 2735–2745.

[12] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 27, Dec. 2014, pp. 1–9.

[13] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. van Gool, "Temporal segment networks: Towards good practices for deep action recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, Oct. 2016, pp. 20–36.

[14] Z. Lan, Y. Zhu, A. G. Hauptmann, and S. Newsam, "Deep local video feature for action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jul. 2017, pp. 1219–1225.

[15] A. Diba, V. Sharma, and L. Van Gool, "Deep temporal linear encoding networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1541–1550.

[16] 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.

[17] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 6299–6308.

[18] 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.

[19] C. Feichtenhofer, "X3D: Expanding architectures for efficient video recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 200–210.

[20] 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)*, Sep. 2018, pp. 305–321.

[21] M. Zolfaghari, K. Singh, and T. Brox, "ECO: Efficient convolutional network for online video understanding," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Sep. 2018, pp. 713–730.

[22] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "SlowFast networks for video recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6202–6211.

[23] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby, "An image is worth 16 × 16 words: Transformers for image recognition at scale," 2020, *arXiv:2010.11929*.

[24] M. Naseer, K. Ranasinghe, S. Khan, M. Hayat, F. S. Khan, and M.-H. Yang, "Intriguing properties of vision transformers," in *Proc. NeurIPS*, Dec. 2021, pp. 23296–23308.

[25] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 10012–10022.

[26] H. Touvron, M. Cord, M. Douze, F. Massa, A. Sablayrolles, and H. E. J'egou, "Training data-efficient image transformers & distillation through attention," in *Proc. Int. Conf. Mach. Learn. (ICML)*, Dec. 2020, pp. 10347–10357.

[27] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" in *Proc. Int. Conf. Mach. Learn. (ICML)*, Feb. 2021, pp. 813–824.

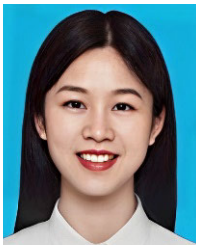
[28] H. Fan, B. Xiong, K. Mangalam, Y. Li, Z. Yan, J. Malik, and C. Feichtenhofer, "Multiscale vision transformers," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6804–6815.

[29] Y. Zhang, X. Li, C. Liu, B. Shuai, Y. Zhu, B. Brattoli, H. Chen, I. Marsic, and J. Tighe, "VidTr: Video transformer without convolutions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 13557–13567.
[30] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lucic, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6816–6826.
[31] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3192–3201.
[32] S. Yan, X. Xiong, A. Arnab, Z. Lu, M. Zhang, C. Sun, and C. Schmid, "Multiview transformers for video recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 3323–3333.
[33] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2018, pp. 7794–7803.
[34] X. Wang, X. Xiong, M. Neumann, A. Piergiovanni, M. S. Ryoo, A. Angelova, K. M. Kitani, and W. Hua, "AttentionNAS: Spatiotemporal attention cell search for video classification," 2020, *arXiv:2007.12034*.
[35] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Jan. 2013.
[36] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," 2016, *arXiv:1607.06450*.
[37] K. Soomro, A. R. Zamir, and M. Shah, "UCF101: A dataset of 101 human actions classes from videos in the wild," 2012, *arXiv:1212.0402*.
[38] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. Int. Conf. Comput. Vis. (ICCV)*, Nov. 2011, pp. 2556–2563.
[39] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014, *arXiv:1412.6980*.
[40] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Oct. 2016, pp. 611–646.
[41] J. Carreira, E. Noland, C. Hillier, and A. Zisserman, "A short note on the kinetics-700 human action dataset," 2019, *arXiv:1907.06987*.
[42] R. A. Shatalin, V. R. Fidelman, and P. E. Ovchinnikov, "Abnormal behaviour detection method for video surveillance applications," *Comput. Opt.*, vol. 41, no. 1, pp. 37–45, 2017.
[43] L. Van Der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 11, pp. 2579–2625, Nov. 2008.
[44] Z. Li, P. Xu, J. Xing, and C. Yang, "SDFormer: A novel transformer neural network for structural damage identification by segmenting the strain field map," *Sensors*, vol. 22, no. 6, p. 2358, Mar. 2022.
[45] Y. Xie, S. Zhang, and Y. Liu, "Abnormal behavior recognition in classroom pose estimation of college students based on spatiotemporal representation learning," *Traitement du Signal*, vol. 38, no. 1, pp. 89–95, Feb. 2021.
[46] M. Rashmi, T. S. Ashwin, and R. M. R. Guddeti, "Surveillance video analysis for student action recognition and localization inside computer laboratories of a smart campus," *Multimedia Tools Appl.*, vol. 80, no. 2, pp. 2907–2929, Jan. 2021.
[47] S. Banerjee, T. S. Ashwin, and R. M. R. Guddeti, "Multimodal behavior analysis in computer-enabled laboratories using nonverbal cues," *Signal, Image Video Process.*, vol. 14, no. 8, pp. 1617–1624, Nov. 2020.

HAI CHUAN LIU received the M.S. degree from Sichuan Normal University, China, in 2019. He is currently pursuing the Ph.D. degree with the Department of Electrical Engineering, Universiti Malaya. He is a Researcher with the Chongqing Key Laboratory of Public Big Data Security Technology. His research interests include pattern recognition, computer vision, and action recognition.

JOON HUANG CHUAH (Senior Member, IEEE) received the B.Eng. degree (Hons.) from Universiti Teknologi Malaysia, the M.Eng. degree from the National University of Singapore, and the M.Phil. and Ph.D. degrees from the University of Cambridge. He is currently the Head of the VIP Research Group and an Associate Professor with the Department of Electrical Engineering, Faculty of Engineering, Universiti Malaya. His research interests include image processing, computational intelligence, IC designs, and scanning electron microscopy. He is a fellow and was the Honorary Secretary of the Institution of Engineers, Malaysia (IEM). He was the Honorary Treasurer of IEEE Computational Intelligence Society (CIS) Malaysia Chapter and the Honorary Secretary of IEEE Council on RFID Malaysia Chapter. He is the Chairperson of the Institution of Engineering and Technology (IET) Malaysia Network. He is a Chartered Engineer registered under the Engineering Council, U.K., and a Professional Engineer registered under the Board of Engineers, Malaysia.

ANIS SALWA MOHD KHAIRUDDIN received the bachelor's degree in electrical engineering from Universiti Tenaga Nasional, Malaysia, in 2008, the master's degree in computer engineering from the Royal Melbourne Institute of Technology (RMIT), Australia, in 2010, and the Ph.D. degree in electrical engineering from Universiti Teknologi Malaysia, in 2014. She is currently a Senior Lecturer with the Department of Electrical Engineering, Faculty of Engineering, Universiti Malaya, Malaysia, where she is also the Head of the Centre of Intelligent Systems for Emerging Technology (CISET), Faculty of Engineering. Her research interests include intelligent systems and image processing.



XIAN MIN ZHAO received the M.S. degree from Sichuan Normal University, China, in 2019. She is currently pursuing the Ph.D. degree with the Faculty of Education, Universiti Malaya. She is a Researcher with the Chongqing Key Laboratory of Public Big Data Security Technology. Her research interests include educational and digital media technology and computer vision.

XIAO DAN WANG received the M.S. degree from Chongqing Technology and Business University, China, in 2019. She is currently a Lecturer with Chongqing Open University. Her research interests include artificial intelligence, fault diagnosis, and the health management of manufacturing systems.