

Article

HSTNet: Violent Action Detection

Fanying Meng, Lian Zou * , Jian Lin and Ziao Liu

School of Electronic Information, Wuhan University, Wuhan 430072, China; fanying518619@gmail.com (F.M.)

* Correspondence: zoulian@whu.edu.cn

Abstract

To enhance public safety and safeguard lives and property, the automatic detection of anomalous and violent behaviors in video has become a key task in intelligent surveillance systems. Violent actions are often abrupt, rapid, and irregular, posing considerable challenges to conventional approaches. Existing methods based on hand-crafted features and convolutional neural networks still exhibit limitations in spatiotemporal feature extraction, recognition accuracy, and model robustness. To address these issues, this paper proposes HSTNet, a hybrid neural architecture that integrates Spiking Neural Networks (SNNs) with Transformers. The framework adopts a dual-branch design: the SNN branch models temporal dynamics in video, while the Transformer branch extracts spatial structural information. A feature interaction module is further introduced to enable deep cross-modal fusion. Experiments on multiple datasets including UCF101, HMDB51, Hockey Fight, and Movies Fight demonstrate that HSTNet achieves significantly higher accuracy than state-of-the-art baselines, indicating strong performance and promising application potential.

Keywords: violent action detection; hybrid neural networks; SNN; transformer

1. Introduction

Violent incidents in public venues such as urban transportation hubs, metro stations, campuses, and shopping malls are highly abrupt and harmful. Consequently, the automatic detection of violent behavior in video has become an urgent problem to address. As a major research area in computer vision, video action recognition aims to model spatiotemporal patterns in videos to automatically identify specific behaviors (e.g., assault, shoving, fighting), and it plays a pivotal role in intelligent security, behavior analysis, and anomaly detection [1].

However, violent action recognition faces the following challenges:

1. **Abruptness and irregularity:** Violent behaviors are typically short lived and intense, with rapid spatiotemporal changes, making it difficult for conventional convolutional models to effectively capture their dynamics [2].
2. **Complex background interference:** Real-world surveillance videos often contain background noise and occlusions, which divert the model's attention from the primary actor and undermine its focus on the action of interest [3].

Existing deep learning-based action recognition methods can be broadly categorized into CNN, RNN, GCN, and Transformer-based approaches. Conventional two-dimensional convolutional neural networks (2D CNNs) are effective in extracting spatial appearance features from individual video frames, but they lack the capability to model temporal dynamics [4]. Three-dimensional convolutional neural networks (3D CNNs) extend convolution operations to the temporal dimension, enabling the joint modeling of local spatiotemporal features; however, such models usually involve high computational complexity and



Academic Editor: Pedro Couto

Received: 10 January 2026

Revised: 7 February 2026

Accepted: 11 February 2026

Published: 12 February 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and

conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

substantial training costs [5]. Recurrent neural networks (RNNs) and their variants, such as long short-term memory (LSTM) and gated recurrent units (GRUs), explicitly model temporal dependencies, but they often suffer from gradient vanishing issues and show limited effectiveness in handling long-term temporal sequences [6]. Graph convolutional networks (GCNs) have achieved promising performance in skeleton-based action recognition by modeling the topological structure of human joints, yet they are sensitive to noise and variations in skeletal structures [7]. In recent years, Transformer-based architectures have demonstrated strong capability in capturing long range dependencies through self attention mechanisms; nevertheless, their quadratic computational complexity poses significant challenges when processing long video sequences [8]. In addition, many state-of-the-art methods adopt multi-branch artificial neural network (ANN) frameworks to fuse RGB and optical-flow modalities, but such approaches typically rely on late fusion strategies and incur high computational and energy costs, thereby limiting their applicability in resource-constrained deployment scenarios [9].

To address the above issues, we propose HSTNet, a hybrid neural architecture that combines Spiking Neural Networks (SNNs) with Transformers. The model adopts a dual-branch design: the SNN branch leverages its event-driven, low-power characteristics to model temporal dynamics in video, while the Transformer branch exploits its global modeling capability to extract spatial structural features. To overcome the modality independence and shallow fusion commonly seen in multi-branch approaches, we introduce a feature interaction module that enables deep inter-modal fusion and noise suppression, thereby enhancing recognition robustness. We conduct comparative experiments on multiple public datasets (UCF101, HMDB51, Hockey Fight, Movies Fight). The results show that HSTNet consistently achieves higher recognition accuracy than mainstream methods (3D CNN, TSN, ViT) and, in particular, demonstrates superior performance and strong generalization on violent action recognition.

The main contributions are as follows:

1. We propose HSTNet, a hybrid neural architecture that integrates Spiking Neural Networks (SNNs) and Transformers, effectively combining their complementary strengths in temporal modeling and spatial representation.
2. We design a feature interaction mechanism that deepens cross-modal fusion, suppresses background interference, and enhances the model's discriminative ability for violent behaviors.
3. Extensive experiments on multiple public action recognition and violence datasets validate the effectiveness of the proposed method, demonstrating superior overall performance and promising application prospects.

2. Related Work

2.1. Two-Stream and 3D CNN-Based Methods

In recent years, deep learning methods have been widely applied to action recognition [10], with two-stream networks and 3D CNNs standing out in both research and practice. Simonyan et al. [11] first proposed the two-stream architecture, which extracts features in separate temporal and spatial branches and fuses them either at intermediate layers or at the final classification layer to generate action predictions by combining temporal and spatial information. Subsequently, Feichtenhofer et al. [12] introduced an early fusion strategy to enable joint optimization of features. Wang et al. [13] proposed Temporal Segment Networks (TSNs), which divide a video into several segments for feature extraction and fusion, effectively alleviating the limitations of two-stream networks in modeling long-range temporal information. In addition, Ma et al. [14] designed a human action recognition model that combines a spatiotemporal two-stream fusion network with additive margin Softmax, further improving recognition accuracy; Ding et al. [15] proposed

a heterogeneous two-stream architecture to reduce redundant interference and enhance feature representation quality.

Despite their strong performance across many tasks, two-stream networks still encounter bottlenecks when handling complex or fine-grained actions. As a result, models based on 3D convolutions have gained traction, and 3D CNNs have been widely adopted for action recognition. Tran et al. proposed the R(2+1)D architecture [16], which factorizes a 3D convolution into separate spatial and temporal convolutions, thereby improving network performance. Feichtenhofer et al. [17] introduced the scalable X3D network, which enhances recognition accuracy while reducing computational cost. For violence detection specifically, Ding et al. [15] jointly modeled spatial and temporal cues with 3D CNNs and performed classification using fully connected layers. Cheng et al. [18] innovatively combined an optical-flow gating mechanism with 3D CNNs, optimizing the max-pooling process to improve feature extraction effectiveness and recognition performance.

2.2. Skeleton-Based Methods

Skeleton-based action recognition analyzes the trajectories of human keypoints and has developed rapidly in recent years. Since skeleton sequences naturally contain joint spatiotemporal information, effectively modeling their temporal dynamics is crucial for recognition performance. Traditional approaches rely primarily on hand-crafted features such as geometric relations among joints, velocity, and acceleration [19] which depend on domain expertise and often generalize poorly across scenarios or large-scale datasets [20]. With the advent of deep learning, powerful feature modeling has been introduced to skeleton action recognition. Existing methods can be broadly categorized into four families: recurrent neural networks (RNNs) [21], convolutional neural networks (CNNs) [22], graph convolutional networks (GCNs) [23], and Transformers [24], each with distinct advantages for different recognition settings.

Zhao et al. [25] proposed an improved RNN structure that achieves faster convergence and lower computational cost compared with conventional LSTMs, and combined with 3D CNNs yields significant performance gains. Li et al. [26] further introduced an adaptive learning framework (RNN-T) that effectively addresses the difficulty of distinguishing fine-grained action categories, and validated its superiority on multiple large-scale datasets. Xu et al. [27] proposed EnsemNN, which builds a main network with one-dimensional convolutional residual blocks and introduces multiple subnetworks to separately process temporal and spatial features of the skeleton. Wang et al. [28] encoded joint trajectories as pseudo-images and used 2D CNNs to extract discriminative features for action classification. Within the GCN paradigm, joints are treated as graph nodes and their connections as edges, enabling effective modeling of the skeleton's topology. Yan et al. [29] were the first to introduce GCNs into skeleton-based action recognition with ST-GCN (SpatioTemporal Graph Convolutional Network), which significantly improved performance through a spatiotemporal graph representation and laid the groundwork for subsequent research. Building on GCNs, Lee et al. [30] proposed LD-GCN (Layered Decomposition GCN), which models action sequences using only joint and bone streams, avoiding redundant computation common in multistream designs and achieving state-of-the-art results on several datasets. Hao et al. [31] further incorporated hypergraph neural networks, using hyper-edges to capture higher-order dependencies and designing hypergraph attention together with improved residual structures. This approach addresses the limitation of focusing only on local physical connections, markedly enhancing modeling accuracy and generalization.

2.3. Transformer-Based Methods

Originally proposed for natural language processing [32], Transformers have been increasingly introduced to computer vision (CV) due to their advantages in handling long-range dependencies and computational efficiency, advancing a wide range of vision tasks. Dosovitskiy et al. proposed the Vision Transformer (ViT) [33], which partitions an image into patches and uses positional encoding for sequence modeling, substantially improving image classification performance. A variety of improved models followed: Lee et al. [34] introduced a multi-path vision Transformer that enhances representational capacity by fusing multi-scale features; Xia et al. [35] incorporated deformable attention to better focus on salient information; and Liu et al. [36] proposed Swin Transformer V2, which, with multiple optimizations, set new records across several vision benchmarks. In action recognition, Transformers likewise perform strongly. Ahn et al. [37] proposed a spatiotemporal cross-attention Transformer that efficiently fuses spatial and temporal cues. Ma et al. [14] applied Transformer-based models to driver action recognition and designed a voting mechanism for precise localization, validating their effectiveness in complex scenarios.

Nevertheless, the high computational complexity of Transformers still constrains their deployment on long videos and high-resolution inputs. Moreover, strategies for spatiotemporal feature fusion remain immature, and the substantial redundancy in video data increases learning difficulty. Designing lightweight, efficient Transformers with coordinated spatiotemporal modeling thus remains an active research direction.

2.4. Hybrid Neural Network-Based Action Recognition

Spiking Neural Networks (SNNs) [38] offer event-driven, low-power computation and exhibit advantages in temporal modeling, making them a useful complement to conventional artificial neural networks (ANNs). Spiking neurons are the fundamental computational units of Spiking Neural Networks (SNNs). The leaky integrate-and-fire (LIF) model is commonly adopted as the neuron model in SNNs. In the LIF model, a neuron updates its membrane potential by integrating incoming inputs over time, while the potential gradually decays due to a leakage mechanism. When the membrane potential exceeds a predefined firing threshold, the neuron emits a spike and the membrane potential is reset. This integrate-and-fire mechanism enables neurons to encode dynamic information through spike timing and firing patterns, thereby endowing SNNs with inherent temporal modeling capability. Moreover, SNNs operate in an event driven manner, where spikes are generated only when the membrane potential reaches the firing threshold, resulting in sparse neural activations. Compared with conventional artificial neural networks that rely on continuous valued computations, this event-driven mechanism significantly reduces the update frequency of neurons and synapses, leading to lower computational complexity and energy consumption. As a result, SNNs exhibit clear advantages for deployment on low-power hardware and resource-constrained edge devices.

However, SNNs still face challenges such as suboptimal accuracy and training difficulty. Sparse spiking signals can lead to vanishing gradients and slow convergence. Although learnable thresholds [39,40] and surrogate gradients [41] provide improvements, SNNs often lag behind ANNs in stability and representational power. To this end, researchers have proposed ANN-SNN hybrid architectures. One strategy uses an SNN branch to process event streams and an ANN for classification [42]; another adopts a dual branch design that separately extracts frame-based and event-based information and fuses them via weighting or interaction mechanisms [43]. For example, Aydin et al. [44] developed an energy-efficient hybrid model for pose estimation, while Zhao's team [43] integrated hybrid modeling into tracking frameworks, demonstrating practical utility.

Despite these advances, many methods still suffer from shallow modality fusion and a lack of explicit interaction mechanisms. Building hybrid networks with deep information interaction to fully exploit the complementarity of ANNs and SNNs is therefore key to achieving high-accuracy, low-power action recognition.

3. Proposed Method

3.1. SNN–Transformer Hybrid Neural Network Architecture

As shown in Figure 1, the proposed HSTNet comprises three components: an SNN branch, an ANN branch, and a feature interaction module. HSTNet adopts two independent branches: the SNN branch is implemented as a Spiking Neural Network based on an MS-ResNet architecture [45], while the ANN branch employs a Swin Transformer [46]. The ANN branch focuses on extracting spatial representations, whereas the SNN branch captures temporal dynamics and interframe motion cues. As illustrated in Figure 1, the blue blocks correspond to the four hierarchical feature extraction stages of the SNN branch, whereas the red blocks indicate the four hierarchical feature extraction stages of the ANN branch. The input video is split into RGB frames and residual (difference) frames. RGB frames are fed to the ANN branch (Swin Transformer) to extract spatial features F_1, F_2, F_3, F_4 ; residual frames are fed to the SNN branch (MS-ResNet) to extract temporal features S_1, S_2, S_3, S_4 . Residual frames provide a lightweight approximation of inter-frame motion and can be directly derived from RGB inputs, enabling effective temporal modeling while avoiding the additional computational overhead introduced by optical flow estimation or event-based preprocessing. This design is consistent with the event-driven and efficiency-oriented characteristics of SNNs, and is particularly suitable for scenarios that emphasize computational efficiency and deployability. At corresponding stages, the two branches exchange complementary information via the feature interaction modules. To reduce computation and broaden the global receptive field, the first two interaction modules are computed at a reduced spatial resolution (average pooling), followed by upsampling to restore the original size. The fused multi-scale features are then passed to a multi-scale enhancement module (inter-scale enhancement, intra-scale enhancement) and a Feature Pyramid Network (FPN) for further cross-scale integration. Finally, the ANN branch's prediction and the multi-branch fused prediction are combined in the classification head to produce the final action label.

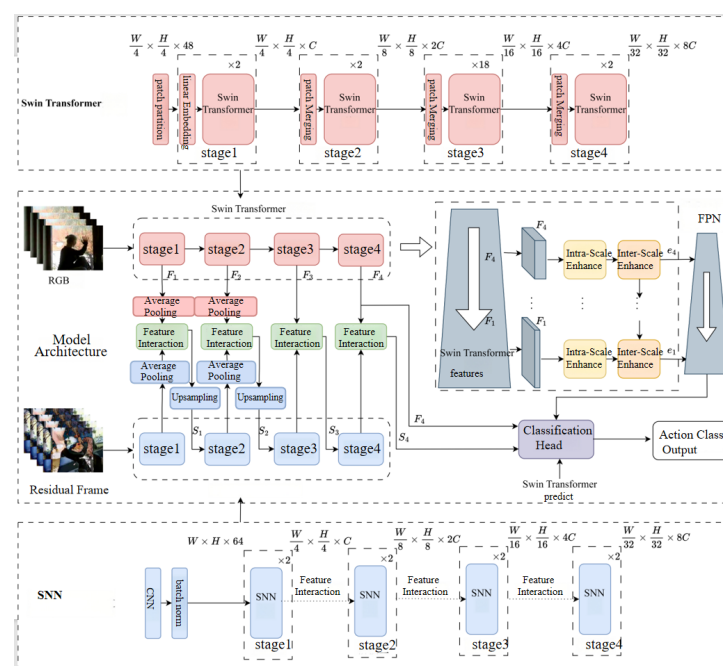


Figure 1. HSTNet network.

3.2. SNN

The SNN branch focuses on modeling temporal dynamics and inter-frame motion patterns in an efficient, event-driven manner, leveraging its suitability for temporal signal processing. As shown in Figure 2, the SNN branch adopts MS-ResNet. Compared with the standard spiking ResNet (vanilla ResNet), MS-ResNet introduces key architectural modifications: it removes the inter-block LIF (leaky integrate-and-fire) neuron modules and adds cross-layer identity mapping shortcut connections that span the entire network. In the vanilla spiking ResNet, as depth increases, model accuracy saturates after stacking about 20 SNN layers, making further improvement difficult. By eliminating the LIF modules, MS-ResNet provides an unobstructed pathway for information flow and effectively prevents the accumulation of ineffective residual representations. These changes markedly enhance scalability, enabling the network to scale to 482 layers on CIFAR-10 and to 104 layers on ImageNet [45]. They not only improve information transmission in deep SNNs, but also substantially enhance the model's scalability and stability. The output feature dimensions of the SNN branch at different stages are detailed in Table 1. As shown, the SNN branch adopts a hierarchical design in which the spatial resolution is progressively reduced while the feature dimensionality is gradually increased across stages. This structure enables multi-level temporal feature extraction with increasing semantic abstraction. In the early stages, average pooling and upsampling are employed to balance computational efficiency and temporal modeling capacity, while maintaining alignment with the corresponding stages of the ANN branch. In deeper stages, higher-dimensional feature representations are learned at lower spatial resolutions, allowing the network to capture high-level temporal semantics and long-term motion patterns.

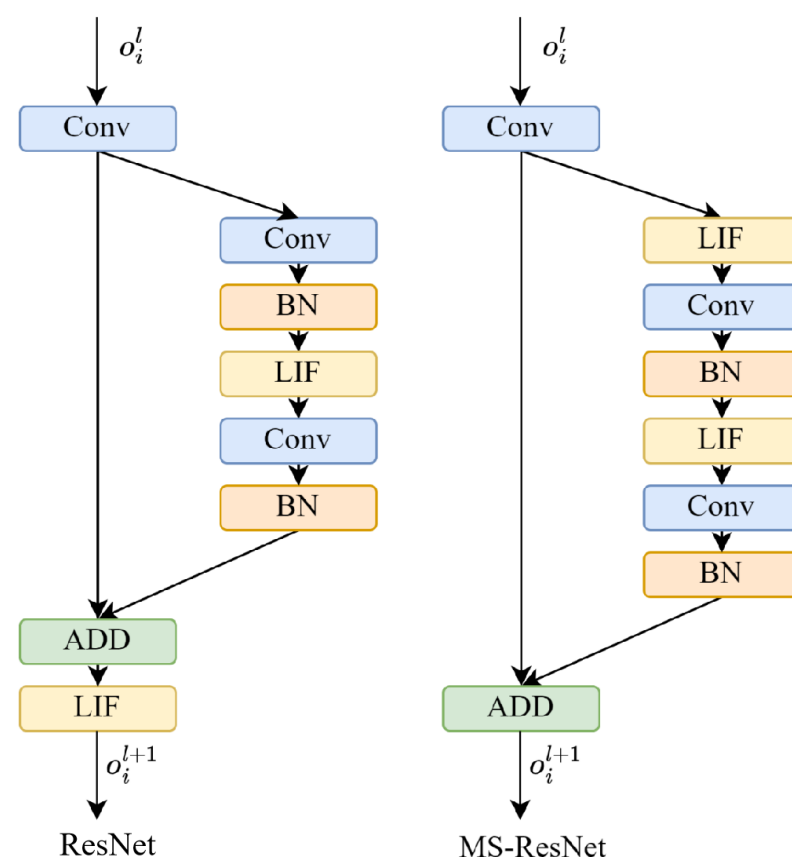


Figure 2. Resnet and MS-Resnet.

Table 1. SNN branch.

Layer	Modules	Feature Maps	Output
Input	–	3	224×224
Conv	1	64	224×224
Stage1	2	128	56×56
Average pooling 1	1	128	14×14
Upsampling 1	1	128	56×56
Stage 2	2	256	28×28
Average pooling 2	1	256	14×14
Upsampling 2	1	256	28×28
Stage 3	2	512	14×14
Stage 4	2	1024	7×7

3.3. ANN

The ANN branch is designed to provide strong spatial representation capability, which is essential for recognizing complex human actions in cluttered scenes. Purely temporal modeling is insufficient for action recognition, as many actions are characterized by subtle spatial configurations and object–human interactions. The ANN branch adopts a Swin Transformer for feature extraction. Compared with a standard Transformer, Swin introduces a local attention mechanism that focuses on local structures while reducing computational complexity and improving efficiency. Specifically, Swin Transformer first applies Patch Partition to the input $H \times W \times 3$ RGB image, splitting it into non-overlapping $M \times M$ windows and computing self-attention independently within each window. For example, given an input RGB frame F , when the patch size is set to 4×4 , the original feature map $F \in \mathbb{R}^{H \times W \times C}$ is converted to $F \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C}$. Next, through a Linear Embedding Layer, these patches are projected into a unified C -dimensional vector space. For each window $X \in \mathbb{R}^{M^2 \times C}$, Swin Transformer computes self-attention within this local region, thereby effectively extracting local features while avoiding the high computational cost of global attention in standard Transformers. The computation is given by:

$$\begin{cases} \text{SA}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}} + B\right)V, \\ Q = XW^Q, \quad K = XW^K, \quad V = XW^V \end{cases} \quad (1)$$

where SA denotes the self-attention operation, and $W^Q \in \mathbb{R}^{C \times C}$, $W^K \in \mathbb{R}^{C \times C}$, and $W^V \in \mathbb{R}^{C \times C}$ are learnable linear projection matrices used to generate the query Q , key K , and value V , respectively; these projections share parameters across windows. B denotes a learnable relative positional encoding, and d_k is the dimensionality of the keys.

To further enhance the model's representational capacity, the local window self-attention is extended to multi-head self-attention (Window Multi-Head Self-Attention, W-MSA). However, relying solely on local attention may restrict information exchange across different windows. To remedy this, Swin Transformer introduces Shifted Window Multi-Head Self-Attention (SW-MSA), which alternates with W-MSA to enable cross-window interaction and strengthen global modeling. The overall structure is shown in Figure 3.

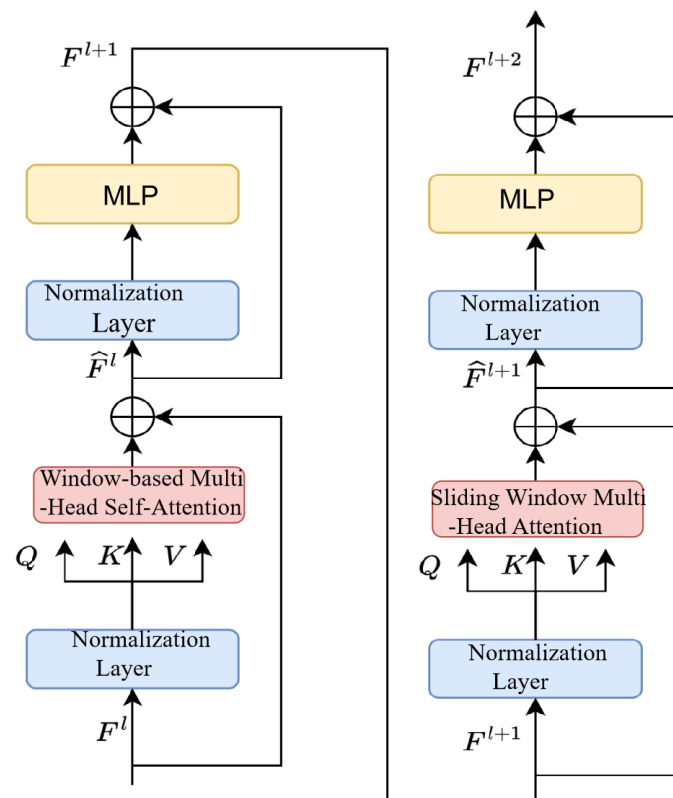


Figure 3. Two consecutive Swin Transformer blocks.

The computation of the alternating cascade of W-MSA and SW-MSA can be expressed as:

$$\begin{cases} \hat{F}^{(\ell)} = \text{W-MSA}(\text{LN}(F^{(\ell)})) + F^{(\ell)}, \\ F^{(\ell+1)} = \text{MLP}(\text{LN}(\hat{F}^{(\ell)})) + \hat{F}^{(\ell)}, \\ \hat{F}^{(\ell+1)} = \text{SW-MSA}(\text{LN}(F^{(\ell+1)})) + F^{(\ell+1)}, \\ F^{(\ell+2)} = \text{MLP}(\text{LN}(\hat{F}^{(\ell+1)})) + \hat{F}^{(\ell+1)}. \end{cases} \quad (2)$$

where LN (LayerNorm) and MLP denote layer normalization and multilayer perceptron, respectively, and l indexes the l -th layer of the Swin Transformer encoder. By cascading W-MSA and SW-MSA, the model effectively enables information exchange across windows, thereby enhancing global feature modeling. In addition, to further capture multi-scale information, Swin Transformer introduces a resolution-reducing feature merging (Patch Merging) operation at each stage to progressively extract higher-level semantic features.

3.4. Feature Interaction Module

While dual-branch architectures can separately model spatial and temporal information, treating the two branches as independent often leads to shallow fusion and limited cross-modal complementarity. In many existing hybrid frameworks, spatial and temporal features are only combined at high-level or decision stages, which restricts their ability to mutually refine each other. To overcome this limitation, we introduce a feature interaction module that enables explicit and bidirectional information exchange between the SNN and Transformer branches at corresponding semantic levels. This module allows temporal motion cues extracted by the SNN branch to guide spatial attention in the ANN branch, while spatial semantic information helps suppress background noise in temporal representations. As a result, the interaction module provides a unique contribution by deepening cross-modal fusion and improving robustness in complex and dynamic scenes. As shown in Figure 4, in the first two dynamic interaction modules we introduce

an average-pooling layer before interaction to reduce spatial resolution, and an upsampling layer after interaction to restore the features to their original resolution. This design reduces computational complexity: after downsampling, the dimensionality of the feature maps decreases, so attention is computed in a more compact feature space, substantially lowering the computational cost. Performing feature interaction at a low spatial resolution biases cross-modal fusion toward global characteristics rather than local details, which facilitates feature complementarity. The subsequent upsampling restores the features to the initial resolution, preserving representational capacity and thereby improving the final recognition performance.

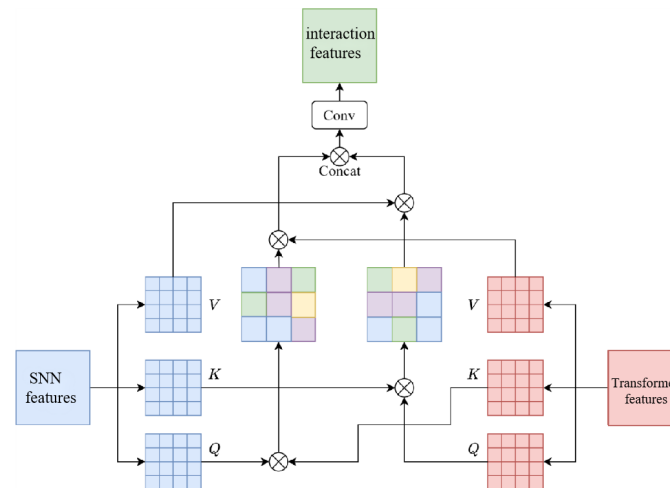


Figure 4. Feature interaction module.

Assume the SNN features are $X_s \in \mathbb{R}^{T \times C \times H \times W}$ and the Transformer features are $X_t \in \mathbb{R}^{T \times C \times H \times W}$. The computation can be expressed as:

$$\begin{cases} A_{s \rightarrow t} = \text{Softmax}\left(\frac{Q_s K_t^T}{\sqrt{d}}\right) \in \mathbb{R}^{T \times T}, \\ A_{t \rightarrow s} = \text{Softmax}\left(\frac{Q_t K_s^T}{\sqrt{d}}\right) \in \mathbb{R}^{T \times T}, \\ Q_s = W_s^Q X_s, \quad K_s = W_s^K X_s, \quad V_s = W_s^V X_s, \\ Q_t = W_t^Q X_t, \quad K_t = W_t^K X_t, \quad V_t = W_t^V X_t \end{cases} \quad (3)$$

$W^Q, W^K, W^V \in \mathbb{R}^{C \times d}$ are learnable projection matrices, and d is the projection dimension. The attention weight matrix $A_{s \rightarrow t}$ indicates how the SNN features attend to the Transformer features, and $A_{t \rightarrow s}$ does the opposite. This feature interaction module effectively suppresses background noise: when the SNN detects an action such as an arm swing at time step t , the Swin Transformer correspondingly strengthens its focus on the arm region, increasing the feature weights there and thereby enabling more precise spatiotemporal fusion.

$$\begin{cases} Z_s = A_{s \rightarrow t} V_t, \\ Z_t = A_{t \rightarrow s} V_s, \\ Z = \text{Conv}(\text{Concat}(Z_s, Z_t)) \end{cases} \quad (4)$$

Z_s denotes the new SNN features after interaction with the Transformer, and Z_t denotes the new Transformer features after interaction with the SNN; Concat denotes concatenation along the channel dimension, and Conv denotes a 1×1 convolutional layer.

3.5. Scale Enhancement Module

Human actions often exhibit significant variations across temporal and spatial scales. Relying on features from a single scale may limit the model's ability to recognize actions with different motion amplitudes and temporal durations. In particular, violent behaviors and complex actions may involve both fine-grained local movements and coarse global motion patterns. To address this issue, we enhance the intermediate features of the Swin Transformer both within-scale and across scales to strengthen multi-scale representations. Specifically, the within-scale enhancement optimizes features at a single scale, whereas the cross-scale enhancement focuses on fusing features from different scales. The detailed structures are shown in Figures 5 and 6.

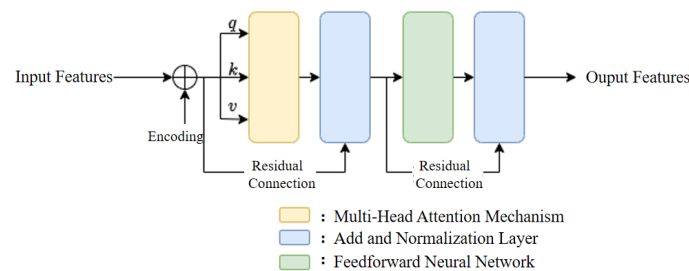


Figure 5. Within-scale enhancement module.

The computation can be formulated as follows:

$$\tilde{X}_i = X_i + PE_i \quad (5)$$

X_i denotes the input features and PE_i the positional encoding. The within-scale enhancement applies a Transformer encoder to single-scale features to improve their representational capacity at that scale.

$$Z_i = \text{TransformerEncoder}(\tilde{X}_i) \quad (6)$$

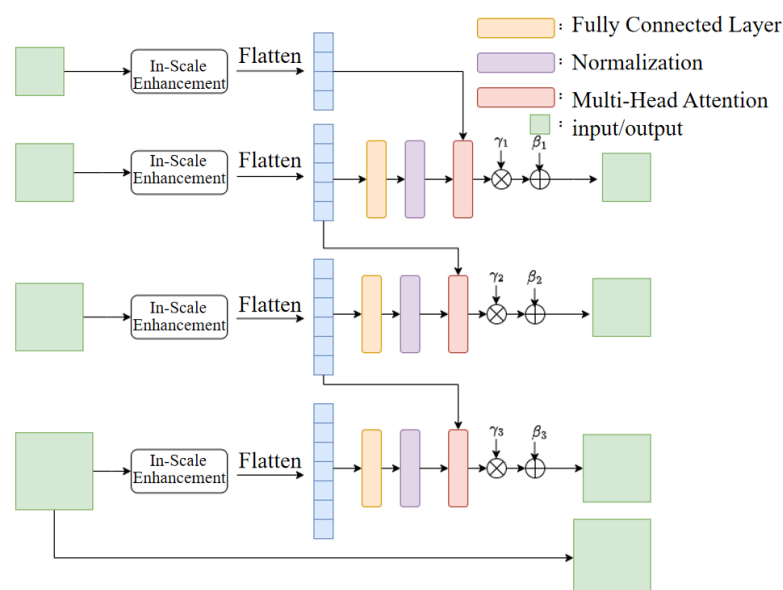


Figure 6. Cross-scale enhancement module.

The cross-scale enhancement leverages a multi-head attention mechanism to capture cross-scale dependencies; it restricts attention to adjacent scales, thereby enabling complementary integration of features at different resolutions.

$$\begin{cases} Z_{i-1}^{(\text{proj})} = W_{i-1} Z_{i-1}, \\ \mu_{i-1}, \sigma_{i-1} = (Z_{i-1}^{(\text{proj})}), \\ \mu_i, \sigma_i = (Z_i), \\ Z_{i-1}^{(\text{norm})} = \frac{Z_{i-1}^{(\text{proj})} - \mu_{i-1}}{\sigma_{i-1} + \varepsilon} \times \sigma_i + \mu_i \end{cases} \quad (7)$$

W_{i-1} is a linear layer; μ and σ denote the mean and variance, respectively; and ε is a small constant to prevent division by zero. γ_i and β_i are learnable parameters, and $\odot$ denotes element-wise multiplication.

$$\begin{cases} Z_i^{\text{attn}} = (Q = Z_i, K = Z_{i-1}^{(\text{norm})}, V = Z_{i-1}^{(\text{norm})}), \\ Z_i^{\text{enhanced}} = \gamma_i \odot Z_i^{\text{attn}} + \beta_i \end{cases} \quad (8)$$

3.6. Classification Head

As shown in Figure 7, the classification head integrates multi-branch information and employs an adaptive weighting mechanism to generate the final prediction. The operator $\otimes$ denotes element-wise multiplication, while the $\otimes$ marked as Concat represents tensor concatenation along a specified dimension. Here, S_4 and F_4 denote the output features from the SNN branch and the ANN branch, respectively. First, S_4 and F_4 are concatenated along the feature dimension, followed by average pooling to reduce feature redundancy and improve computational efficiency. Meanwhile, the feature outputs provided by the Feature Pyramid Network (FPN) are also subjected to average pooling to ensure scale consistency. Subsequently, the pooled FPN features are concatenated with the previously fused features to further integrate multi-source information. Next, a fully connected layer is applied to compute the weighting factor α , which is used to perform weighted feature fusion. This adaptive weighting mechanism enables the model to dynamically adjust the relative importance of features from different sources. Finally, the fused feature representation is fed into a fully connected layer for classification, and the resulting prediction is added to the prediction from the ANN branch to produce the final output.

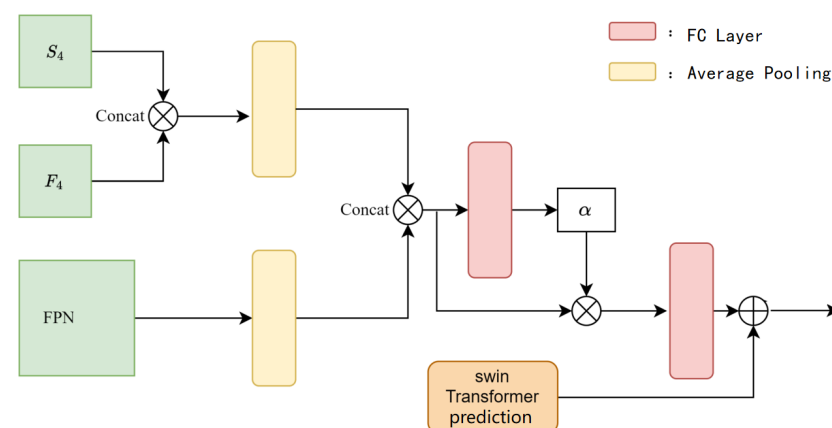


Figure 7. Classification head module.

4. Experiments

4.1. Datasets

The proposed method is evaluated on four public datasets (UCF101, HMDB51, Hockey Fight, and Movies Fight), covering two types of scenarios: general action recognition and violent behavior detection. UCF101 and HMDB51 encompass various action types such as sports, dance, and daily activities, with complex backgrounds and diverse motion subjects. Hockey Fight and Movies Fight are specifically designed for violent action detection. These four datasets differ significantly in terms of their sources, background complexity, and action types, making them suitable for a comprehensive evaluation of the model's spatiotemporal modeling ability and robustness.

UCF101 consists of 13,320 video clips spanning 101 action categories, mainly collected from YouTube, including sports, dance, and daily activities. The dataset is characterized by complex backgrounds, diverse motion subjects, significant camera motion, and frequent occlusions, posing high demands on the model's spatiotemporal modeling capability and robustness.

HMDB51 contains 6766 video clips across 51 action classes, sourced from movies, television programs, and online videos. Compared with UCF101, HMDB51 features more diverse action categories and more severe background clutter, making it a challenging benchmark for evaluating action recognition performance in complex scenarios.

The Hockey Fight dataset is specifically designed for violent behavior detection and includes 1000 video clips extracted from National Hockey League (NHL) games, with 500 clips depicting real fight events and the remaining 500 clips representing non-violent gameplay. All videos are captured under a uniform scene background and consistent camera viewpoints, allowing the model to focus on action-related cues rather than background variations.

The Movies Fight dataset is also used for violence detection, where fight scenes are collected from movies, and non-violent clips are selected from other action recognition datasets. Although relatively small in scale, this dataset involves diverse characters, motion patterns, and background variations, serving as a complementary benchmark to evaluate the model's recognition ability under non-uniform and complex visual conditions.

4.2. Experimental Settings and Evaluation Metrics

Input modality. HSTNet uses only RGB video as input.

Implementation details. All experiments are implemented using the PyTorch deep learning framework (version 1.8.0) with Python 3.9 on an Ubuntu 20.04 operating system. Model training is performed using an NVIDIA A100-SXM4 GPU with CUDA 11.1 support, while the CPU used is an AMD EPYC 7543.

Preprocessing. We adopt fixed-interval sampling to extract contiguous clips of 16 frames from each original video, with a stride of 16 to ensure broad temporal coverage. Each sampled clip is resized to a standard spatial resolution of 224×224 pixels. Pixel values are then normalized to the range $[0, 1]$.

Training configuration. The initial learning rate is set to 3×10^{-4} with a batch size of 8. For HMDB51 and UCF101, the models are fine-tuned from pretrained backbones rather than trained from scratch. In our preliminary experiments, we observed that the training loss and validation accuracy converge rapidly within the first few epochs, and further increasing the number of epochs brings marginal performance improvements while significantly increasing computational cost. Therefore, to ensure fair comparison across methods and maintain computational efficiency, we adopt a limited number of training epochs in the reported experiments. This setting follows common practice in recent action recognition works that rely on pretrained Transformer backbones. Training is conducted for

5 epochs. A cosine annealing learning rate schedule is applied to balance fast convergence in early stages with fine-grained optimization in later stages.

Metrics. We report accuracy and recall to assess recognition performance.

- (1) **Accuracy.** Measures the overall proportion of correctly classified samples—the more action instances whose labels are predicted correctly, the higher the accuracy. Here, *Accuracy* denotes the model's classification accuracy for actions; TP (true positive), TN (true negative), FP (false positive), and FN (false negative).

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \quad (9)$$

- (2) **Recall.** An important metric for a classifier's ability to identify positive samples, defined as the proportion of actual positives that are correctly detected by the model. Here, TP (true positive) is the number of samples correctly predicted as positive, and FN (false negative) is the number of samples that are actually positive but predicted as negative. Recall ranges from 0 to 1; a higher value indicates fewer missed positives and better detection performance.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (10)$$

- (3) **FLOPs.** FLOPs refer to the total number of floating-point operations required to perform a single forward or backward pass of a neural network, and are widely used to evaluate computational complexity. FLOPs can be computed by summing the operation counts of all layers in the network.
- (4) **SYOps.** Synaptic Operations (SYOps) are primarily used to measure the computational cost of Spiking Neural Networks (SNNs). They represent the number of synaptic computations occurring between neurons, the number of times neurons receive input spikes and perform synaptic processing.
- (5) **Params.** Params denotes the total number of trainable parameters in a neural network and is commonly used to measure the model's storage requirements. In general, a larger number of parameters leads to higher memory consumption and increased computational complexity.

4.3. Main Results

We compare the proposed method with mainstream models on the HMDB51 and UCF101 datasets. The results are summarized in Table 2, where “–” denotes missing entries; the best results are highlighted in bold. The compared baselines cover representative ANNs, SNNs, and hybrid neural networks, including 2D CNN-based ResNet18 [47] and ResNet50 [47]; 3D CNN-based C3D [48] and Res3D [49]; Transformer-based ViT-B [33], ViT-L [33], Swin-B [46], and Swin-L [46]; and the Spiking Neural Network MS-ResNet [45] and the hybrid neural network ReSpike (built upon ResNet18/ResNet50) [50].

From the results in Table 2, HSTNet outperforms most state-of-the-art action recognition models on the HMDB51 dataset. In terms of recognition accuracy, HSTNet improves over ResNet18 by 12.70% and over ResNet50 by 9.12%. From the results in Table 3, HSTNet maintains superior performance on the UCF101 dataset as well. Owing to the richer action content in UCF101, the performance gains are even more pronounced: HSTNet improves over ResNet18 by 11.78% and over ResNet50 by 8.73%. These results indicate that HSTNet is able to achieve competitive recognition accuracy while offering favorable computational efficiency and a more lightweight model design.

Table 2. Performance comparison of different action recognition methods on the HMDB51 dataset.

Type	Method	Accuracy (%)	FLOPs (G)	SYOps (G)	Params (M)
2D ANN	ResNet18	50.70	29.18	—	11.70
2D ANN	ResNet50	54.28	66.11	—	25.60
3D ANN	C3D	—	154.19	—	78.20
3D ANN	Res3D	54.14	163.42	—	33.20
Transformer	ViT-B	59.91	269.81	—	85.86
Transformer	ViT-L	62.25	954.97	—	303.15
Transformer	Swin-B	61.17	242.70	—	86.68
SNN	MS-ResNet	40.76	0.35	3.19	0.56
Hybrid	ReSpike (ResNet18)	52.70	14.47	1.93	30.08
Hybrid	ReSpike (ResNet50)	58.20	23.57	1.93	44.58
Hybrid	HSTNet	63.40	377.56	9.32	151.04

Note: The bold values indicate the best results.

To analyze the recognition performance of HSTNet across different action categories, we further investigate the classification accuracy of each action class on the HMDB51 dataset. Figures 8 and 9 illustrate the class-wise recognition results of HSTNet. The results show that HSTNet achieves relatively high accuracy on most action categories, particularly for actions with large motion amplitudes (e.g., class 6 “Climb” and class 16 “Golf”), where the model demonstrates superior performance. In contrast, the recognition accuracy is relatively lower for daily actions with small motion amplitudes and subtle variations (e.g., class 2, class 5, and class 43, such as “Stand” and “Clap”). Considering that actions with small motion amplitudes in real-world scenarios are generally less harmful, whereas violent events are often accompanied by large and abrupt motion changes, this recognition characteristic highlights the practical value of the proposed method for real-world applications.

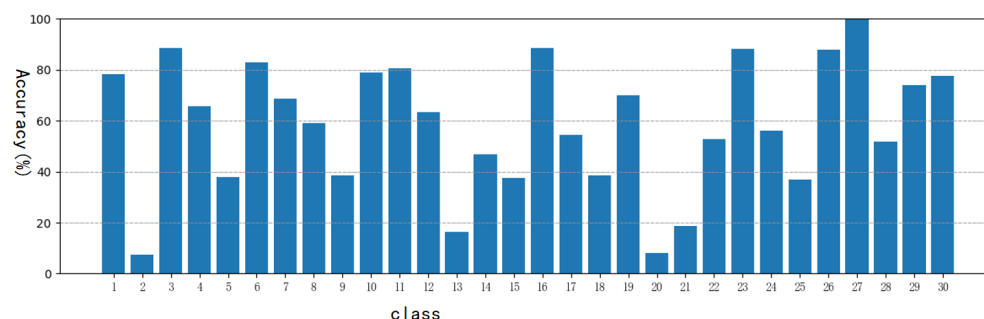
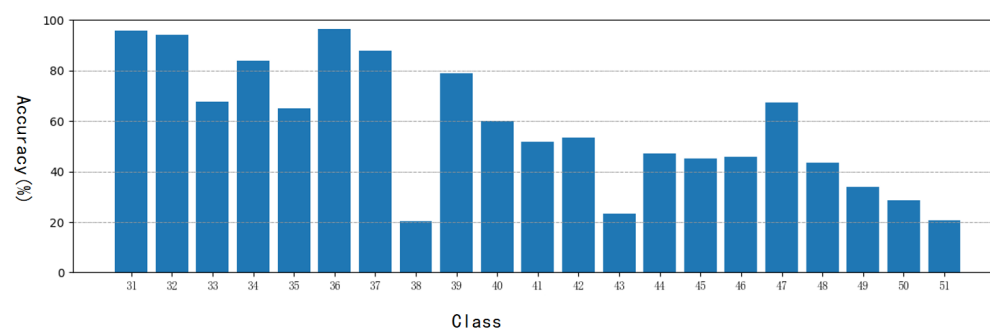
**Figure 8.** Accuracy of HSTNet on the first 30 action classes of the HMDB51 dataset.**Figure 9.** Accuracy of HSTNet on the remaining 21 action classes of the HMDB51 dataset.

Table 3. Accuracy on UCF101.

Category	Method	Accuracy (%)
2D ANN	ResNet18	79.14
2D ANN	ResNet50	82.20
3D ANN	C3D	85.23
3D ANN	Res3D	85.82
Transformer	Vit-B	85.53
Transformer	Vit-L	87.43
Transformer	Swin-B	90.03
SNN	MS-ResNet	60.55
Hybrid	ReSpike (ResNet18)	70.50
Hybrid	ReSpike (ResNet50)	85.60
Hybrid	HSTNet	90.93

Note: The bold values indicate the best results.

4.4. Ablation Studies

We conduct an in-depth study of how the dynamic feature interaction module, the within-scale enhancement module, the cross-scale enhancement module, and the Feature Pyramid Network (FPN) affect recognition accuracy. By progressively inserting these components into a baseline model, we quantify the contribution of each and further analyze the overall effect of different module combinations. As shown in Table 4, the baseline attains an accuracy of 56.09%. After introducing only the dynamic feature interaction module, HSTNet reaches 58.10%, indicating that the proposed interaction effectively fuses spatiotemporal features and enhances action recognition. Adding the within-scale enhancement module and FPN yields a further absolute gain of 1.14%, demonstrating the positive role of multi-scale feature extraction in video action recognition. With the cross-scale enhancement module added, accuracy increases to 59.46%, confirming that the synergy between cross-scale and within-scale enhancement effectively integrates multi-scale information and strengthens the model's representational capacity. Finally, after incorporating the classification head to fuse multi-branch features, the model achieves an accuracy of 63.40%.

Table 4. Ablation study on module effectiveness.

Feature Interaction	Within-Scale Enhancement	Cross-Scale Enhancement	Classification Head	Accuracy (%)
X	X	X	X	56.08
✓	X	X	X	58.10
✓	✓	X	X	59.24
✓	✓	✓	X	59.49
✓	✓	✓	✓	63.40

Note: The bold values indicate the best results.

The classification head in this experiment fuses feature outputs from different branches. Following the order of fusion inside the head, we denote the three fusion points as Position 1, Position 2, and Position 3. To further validate the effectiveness of the fusion strategy, we compare two alternatives—channel-wise concatenation and element-wise addition—to examine their information-integration capability and impact on overall performance in our network. The results are summarized in Table 5, where Add denotes element-wise addition and Concat denotes channel-wise concatenation. The experimental results indicate that the designed classification head fully exploits the complementary information of SNN and ANN; together with the multi-scale representational power of FPN and an adaptive weighting mechanism, it yields more accurate and robust classification decisions.

Table 5. Ablation study of different fusion strategies.

Position1	Position2	Position3	Accuracy (%)
Add	Concat	Add	62.35
Concat	Add	Add	62.05
Concat	Concat	Concat	59.83
Concat	Concat	Add	63.40

Note: The bold values indicate the best results.

4.5. Real-World Scenarios

To comprehensively validate the performance of the proposed algorithm in high-threat public scenarios—especially for violence detection—we conduct comparative experiments on the Hockey Fight and Movies Fight datasets against a range of mainstream methods. The compared approaches include representative models such as ViF [51], ViF+OVIF [52], ResNet50 [47], and C3D [48]. The HSTNet results are obtained from our experiments under the settings in Section 4.2, while the results of ViF, ViF+OVIF, ResNet50, and C3D are directly taken from their respective publications on the same datasets. The results are reported in Tables 6 and 7.

Table 6. Accuracy on Hockey Fight.

Method	Accuracy (%)	Recall
ViF	82.00	–
ViF+OVIF	87.50	–
ResNet50	95.50	0.96
C3D	93.50	0.92
HSTNet	96.50	0.97

Note: The bold values indicate the best results.

Table 7. Accuracy on Movies Fight.

Method	Accuracy (%)	Recall
ViF	–	–
ViF+OVIF	–	–
ResNet50	96.00	0.98
C3D	82.00	0.79
HSTNet	98.00	1.00

Note: The bold values indicate the best results.

The videos in the Hockey Fight dataset are all taken from a uniform scene background, with consistent camera angles and visual style, which aligns with the characteristics of real-world video surveillance environments. The results in Table 6 show that the proposed HSTNet achieves the best performance with a 96.50% accuracy and the highest recall rate on this dataset, fully demonstrating its sensitivity in capturing the intense motion changes of violent behavior in single-scene video surveillance. To visually present the recognition results of the HSTNet algorithm, the confusion matrices for HSTNet and C3D are plotted as shown in Figure 10. It can be observed that HSTNet shows significant advantages in recognizing violent actions. Table 7 presents the experimental results on the Movies Fight dataset. This dataset covers a variety of complex fight scenes, with non-uniform background environments, making it highly challenging. The experimental results indicate that the HSTNet model achieves a high accuracy of 98.00% and the highest recall rate, which fully proves its strong adaptability and recognition performance in handling complex and varied individual behaviors.

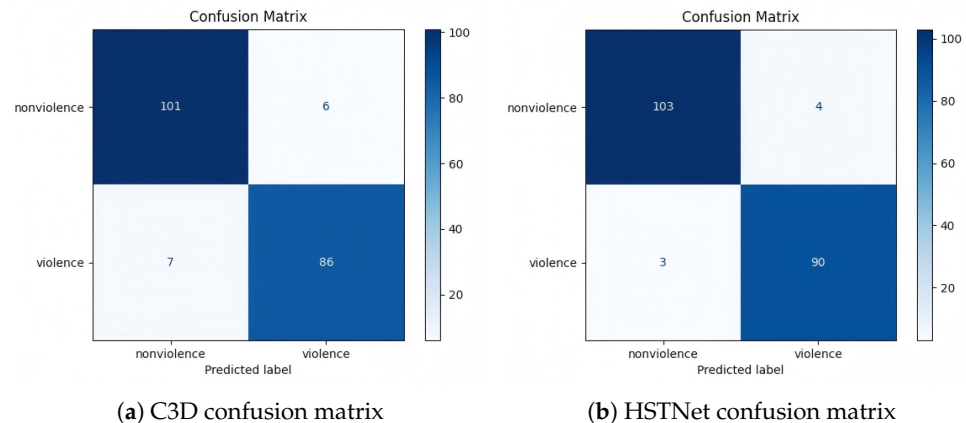


Figure 10. Comparison of confusion matrices of C3D and HSTNet.

5. Conclusions

We proposed HSTNet, a hybrid architecture that integrates a Spiking Neural Network (MS-ResNet) with a Transformer (Swin Transformer) for video action recognition. To enable efficient information exchange between the spiking and artificial neural branches, we designed a feature interaction module that facilitates deep fusion of spatiotemporal cues while suppressing background noise. To further exploit multi-scale information, we introduced within-scale and cross-scale enhancement modules together with a Feature Pyramid Network (FPN), collectively improving feature expressiveness. In addition, a dedicated classification head aggregates multi-branch outputs to enhance recognition accuracy. Experiments on two public video datasets, HMDB51 and UCF101, show that HSTNet achieves substantial gains in accuracy. Moreover, in high-public-risk scenarios of violence detection, the model exhibits strong performance, underscoring its practical applicability. These performance gains are not achieved at the cost of a substantial increase in model parameters or computational complexity. Systematic ablation studies further indicate that, although each functional module contributes a relatively modest improvement individually, their effects are stable and consistent in enhancing overall performance. This demonstrates a well-balanced design trade-off between recognition accuracy, computational efficiency, and model complexity. Despite these gains, there remains room for improvement in capturing long-range temporal dependencies within the hybrid framework. Although the SNN branch effectively models long-term temporal cues, training SNNs can be challenging; more efficient learning algorithms are needed to accelerate training and improve convergence. Future work may explore conversion-based ANN-SNN frameworks that leverage advances in conventional neural networks while reducing the number of timesteps and speeding up inference. Building on this, we also plan to investigate on-device deployment of HSTNet on standalone compute platforms. Finally, the present study focuses primarily on RGB and residual frames, whereas real-world videos are often multi-modal (e.g., RGB, depth, skeleton, optical flow). Future research will consider cross-modal alignment mechanisms and optimized fusion strategies to enhance adaptability and robustness, thereby improving generalization in complex environments.

Author Contributions: Conceptualization, L.Z.; Methodology, F.M. and L.Z.; Software, F.M. and Z.L.; Validation, F.M. and J.L.; Formal analysis, F.M.; Resources, L.Z.; Data curation, F.M. and Z.L.; Writing—original draft, F.M.; Writing—review & editing, L.Z., F.M., J.L. and Z.L.; Visualization, F.M.; Supervision, L.Z.; Project administration, L.Z.; Funding acquisition, L.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Publicly available datasets were analyzed in this study. The HMDB51 dataset can be found at <http://serre-lab.clps.brown.edu/resource/hmdb-a-large-human-motion-database/>, accessed on 7 February 2026. The UCF101 dataset can be found at <https://www.kaggle.com/datasets/matthewjansen/ucf101-action-recognition>, accessed on 7 February 2026.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Naimi, S.; Bouachir, W.; Bilodeau, G.A.; Mishara, B. SSTAR: Skeleton-Based Spatio-Temporal Action Recognition for Intelligent Video Surveillance and Suicide Prevention in Metro Stations. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), Waikoloa, HI, USA, 3–7 January 2025; pp. 813–823.
2. Alomar, K.; Aysel, H.I.; Cai, X. CNNs, RNNs and Transformers in Human Action Recognition: A Survey and a Hybrid Model. *Artif. Intell. Rev.* **2025**, *58*, 387. [CrossRef]
3. Li, H.; Lo, J.T.Y. A Review on the Use of Top-View Surveillance Videos for Pedestrian Detection, Tracking and Behavior Recognition across Public Spaces. *Accid. Anal. Prev.* **2025**, *215*, 107986. [CrossRef] [PubMed]
4. Lin, C.; Ma, T.; Wu, F.; Qian, J.; Liao, F.; Huang, J. Application of Temporal Action Detection Technology in Abnormal Event Detection of Surveillance Video. *IEEE Access* **2025**, *13*, 26958–26972. [CrossRef]
5. Zhang, Y.; Fan, K.; Yu, Y. Research on Passengers' Behavior Recognition Method in Public Transport Vehicles Based on Efficient 3D CNN. *Multimed. Syst.* **2025**, *31*, 30. [CrossRef]
6. Mahamkali, A.; Gali, M.; Jena, S.R.; Sreenivas, V. Convolutional Attention-Based Bidirectional Recurrent Neural Network for Human Action Recognition. *Comput. Intell.* **2025**, *41*, e70049. [CrossRef]
7. Chen, D.; Chen, M.; Wu, P.; Wu, M.; Zhang, T.; Li, C. Two-stream spatio-temporal GCN-transformer networks for skeleton-based action recognition. *Sci. Rep.* **2025**, *15*, 4982. [CrossRef] [PubMed]
8. Liu, X.; Wang, Q.; Hu, Y.; Tang, X.; Zhang, S.; Bai, S.; Bai, X. End-to-end temporal action detection with transformer. *IEEE Trans. Image Process.* **2022**, *31*, 5427–5441. [CrossRef] [PubMed]
9. Liu, X.; Bai, S.; Bai, X. An empirical study of end-to-end temporal action detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 20010–20019.
10. Mahbub, U.; Imtiaz, H.; Ahad, M.A.R. An optical flow based approach for action recognition. In Proceedings of the 14th International Conference on Computer and Information Technology (ICCIT 2011), Dhaka, Bangladesh, 22–24 December 2011; pp. 646–651.
11. Nam, J.; Alghoniemy, M.; Tewfik, A.H. Audio-visual content-based violent scene characterization. In Proceedings of the 1998 International Conference on Image Processing, Chicago, IL, USA, 4–7 October 1998; Volume 1, pp. 353–357.
12. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional two-stream network fusion for video action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 1933–1941.
13. Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; Gool, L.V. Temporal segment networks: Towards good practices for deep action recognition. In Proceedings of the European Conference on Computer Vision, Amsterdam, The Netherlands, 11–14 October 2016; pp. 20–36.
14. Ma, Y.; Yuan, L.; Abdelraouf, A.; Han, K.; Gupta, R.; Li, Z.; Wang, Z. M2DAR: Multi-view multi-scale driver action recognition with vision transformer. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Vancouver, BC, Canada, 17–24 June 2023; pp. 5287–5294.
15. Ding, C.; Fan, S.; Zhu, M.; Feng, W.; Jia, B. Violence detection in video using 3D convolutional neural networks. In *Advances in Visual Computing, ISVC 2014*; Springer: Cham, Switzerland, 2014; pp. 551–558.
16. Tran, D.; Wang, H.; Torresani, L.; Ray, J.; LeCun, Y.; Paluri, M. A closer look at spatiotemporal convolutions for action recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6450–6459.
17. Feichtenhofer, C. X3D: Expanding architectures for efficient video recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 13–19 June 2020; pp. 203–213.
18. Cheng, M.; Cai, K.; Li, M. RWF-2000: An open large scale video database for violence detection. In Proceedings of the 25th International Conference on Pattern Recognition, Milan, Italy, 10–15 January 2021; pp. 4183–4190.
19. Wang, T.; Liu, H.; Ding, R.; Li, W.; You, Y.; Li, X. Interweaved graph and attention network for 3D human pose estimation. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, Rhodes Island, Greece, 4–10 June 2023; pp. 1–5.

20. Wang, L.; Huynh, D.Q.; Koniusz, P. A comparative review of recent Kinect-based action recognition algorithms. *IEEE Trans. Image Process.* **2019**, *29*, 15–28. [[CrossRef](#)] [[PubMed](#)]
21. Wang, H.; Wang, L. Modeling temporal dynamics and spatial configurations of actions using two-stream recurrent neural networks. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 499–508.
22. Ding, Z.; Wang, P.; Ogunbona, P.O.; Li, W. Investigation of different skeleton features for CNN-based 3D action recognition. In Proceedings of the IEEE International Conference on Multimedia & Expo Workshops, Hong Kong, China, 10–14 July 2017; pp. 617–622.
23. Li, M.; Chen, S.; Chen, X.; Zhang, Y.; Wang, Y.; Tian, Q. Actional-structural graph convolutional networks for skeleton-based action recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 3595–3603.
24. Cho, S.; Maqbool, M.; Liu, F.; Foroosh, H. Self-attention network for skeleton-based human action recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Snowmass Village, CO, USA, 1–5 March 2020; pp. 635–644.
25. Zhao, R.; Ali, H.; Van der Smagt, P. Two-stream RNN/CNN for action recognition in 3D videos. In Proceedings of the IEEE/RISJ International Conference on Intelligent Robots and Systems, Vancouver, BC, Canada, 24–28 September 2017; pp. 4260–4267.
26. Li, W.; Wen, L.; Chang, M.-C.; Lim, S.N.; Lyu, S. Adaptive RNN tree for large-scale human action recognition. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 1444–1452.
27. Xu, Y.; Cheng, J.; Wang, L.; Xia, H.; Liu, F.; Tao, D. Ensemble one-dimensional convolution neural networks for skeleton-based action recognition. *IEEE Signal Process. Lett.* **2018**, *25*, 1044–1048. [[CrossRef](#)]
28. Wang, P.; Li, Z.; Hou, Y.; Li, W. Action recognition based on joint trajectory maps using convolutional neural networks. In Proceedings of the ACM International Conference on Multimedia, Amsterdam, The Netherlands, 15–19 October 2016; pp. 102–106.
29. Yan, S.; Xiong, Y.; Lin, D. Spatial temporal graph convolutional networks for skeleton-based action recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; pp. 7444–7452.
30. Lee, J.; Lee, M.; Lee, D.; Lee, S. Hierarchically decomposed graph convolutional networks for skeleton-based action recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Paris, France, 1–6 October 2023; pp. 10444–10453.
31. Hao, X.; Li, J.; Guo, Y.; Jiang, T.; Yu, M. Hypergraph neural network for skeleton-based action recognition. *IEEE Trans. Image Process.* **2021**, *30*, 2263–2275. [[CrossRef](#)] [[PubMed](#)]
32. Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2017; Volume 30, pp. 6000–6010.
33. Dosovitskiy, A.; Beyer, L.; Kolesnikov, A.; Weissenborn, D.; Zhai, X.; Unterthiner, T.; Dehghani, M.; Minderer, M.; Heigold, G.; Gelly, S.; et al. An image is worth 16×16 words: Transformers for image recognition at scale. *arXiv* **2020**, arXiv:2010.11929.
34. Lee, Y.; Kim, J.; Willette, J.; Hwang, S.J. MPViT: Multi-path vision transformer for dense prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 7287–7296.
35. Xia, Z.; Pan, X.; Song, S.; Li, L.E.; Huang, G. Vision transformer with deformable attention. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 4794–4803.
36. Liu, Z.; Hu, H.; Lin, Y.; Yao, Z.; Xie, Z.; Wei, Y.; Ning, J.; Cao, Y.; Zhang, Z.; Dong, L.; et al. Swin Transformer V2: Scaling up capacity and resolution. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, New Orleans, LA, USA, 18–24 June 2022; pp. 12009–12019.
37. Ahn, D.; Kim, S.; Hong, H.; Ko, B.C. Star-transformer: A spatio-temporal cross attention transformer for human action recognition. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, Waikoloa, HI, USA, 2–7 January 2023; pp. 3330–3339.
38. Li, Y.; Geller, T.; Kim, Y.; Panda, P. SEENN: Towards temporal spiking early exit neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2023; Volume 36, pp. 63327–63342.
39. Hagenaars, J.; Paredes-Vallés, F.; De Croon, G. Self-supervised learning of event-based optical flow with spiking neural networks. In *Advances in Neural Information Processing Systems*; Curran Associates Inc.: Red Hook, NY, USA, 2021; Volume 34, pp. 7167–7179.
40. Kosta, A.K.; Roy, K. Adaptive-SpikeNet: Event-based optical flow estimation using spiking neural networks with learnable neuronal dynamics. In Proceedings of the IEEE International Conference on Robotics and Automation, Laguna Hills, CA, USA, 11–13 December 2023; pp. 6021–6027.
41. Lee, C.; Sarwar, S.S.; Panda, P.; Srinivasan, G.; Roy, K. Enabling spike-based backpropagation for training deep neural network architectures. *Front. Neurosci.* **2020**, *14*, 497482. [[CrossRef](#)] [[PubMed](#)]
42. Yang, Z.; Wu, Y.; Wang, G.; Yang, Y.; Li, G.; Deng, L.; Zhu, J.; Shi, L. DashNet: A hybrid artificial and spiking neural network for high-speed object tracking. *arXiv* **2019**, arXiv:1909.12942.
43. Zhao, R.; Yang, Z.; Zheng, H.; Wu, Y.; Liu, F.; Wu, Z.; Li, L.; Chen, F.; Song, S.; Zhu, J.; et al. A framework for the general design and computation of hybrid neural networks. *Nat. Commun.* **2022**, *13*, 3427–3438. [[CrossRef](#)] [[PubMed](#)]

44. Aydin, A.; Gehrig, M.; Gehrig, D.; Scaramuzza, D. A hybrid ANN-SNN architecture for low-power and low-latency visual perception. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seattle, WA, USA, 17–18 June 2024; pp. 5701–5711.
45. Hu, Y.; Deng, L.; Wu, Y.; Yao, M.; Li, G. Advancing spiking neural networks toward deep residual learning. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *36*, 2353–2367. [[CrossRef](#)] [[PubMed](#)]
46. Liu, Z.; Lin, Y.; Cao, Y.; Hu, H.; Wei, Y.; Zhang, Z.; Lin, S.; Guo, B. Swin Transformer: Hierarchical vision transformer using shifted windows. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, QC, Canada, 10–17 October 2021; pp. 10012–10022.
47. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
48. Tran, D.; Bourdev, L.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision, Santiago, Chile, 7–13 December 2015; pp. 4489–4497.
49. Tran, D.; Ray, J.; Shou, Z.; Chang, S.F.; Paluri, M. ConvNet architecture search for spatiotemporal feature learning. *arXiv* **2015**, arXiv:1511.06290.
50. Xiao, S.; Li, Y.; Kim, Y.; Lee, D.; Panda, P. ReSpike: Residual frames-based hybrid spiking neural networks for efficient action recognition. *Neuromorphic Comput. Eng.* **2025**, *5*, 014009. [[CrossRef](#)]
51. Hassner, T.; Itcher, Y.; Kliper-Gross, O. Violent flows: Real-time detection of violent crowd behavior. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, Providence, RI, USA, 16–21 June 2012; pp. 1–6.
52. Gao, Y.; Liu, H.; Sun, X.; Wang, C.; Liu, Y. Violence detection using oriented violent flows. *Image Vis. Comput.* **2016**, *48*, 37–41. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.