Journal of Ambient Intelligence and Humanized Computing (2023) 14:15983–15996
https://doi.org/10.1007/s12652-023-04679-6

ORIGINAL RESEARCH

Two-stage complex action recognition framework for real-time surveillance automatic violence detection

Dylan Josh Domingo Lopez[1,2,3,4] · **Cheng-Chang Lien**[1]

Received: 17 September 2021 / Accepted: 21 August 2023 / Published online: 19 September 2023

Abstract
Violent action classification in community-based surveillance is a particularly challenging concept in itself. The ambiguity of violence as a complex action can lead to the misclassification of violence-related crimes in detection models and the increased complexity of intelligent surveillance systems leading to greater costs in operations or cost of lives. This paper demonstrates a novel approach to performing automatic violence detection by considering violence as complex actions mitigating oversimplification or overgeneralization of detection models. The proposed work supports the notion that violence is a complex action and is classifiable through decomposition into more identifiable actions that could be easily recognized by human action recognition algorithms. A two-stage framework was designed to detect simple actions which are sub-concepts of violence in a two-stream action recognition architecture. Using a basic logistic regression layer, simple actions were further classified as complex actions for violence detection. Varying configurations of the work were tested, such as applying action silhouettes, varying activation caching sizes, and different pooling methods for post-classification smoothing. The framework was evaluated considering accuracy, recall, and operational speed considering its implications in community deployment. The experimental results show that the developed framework reaches 21 FPS operation speeds for real-time operations and 11 FPS for non-real-time operations. Using the proposed variable caching algorithm, median pooling results in accuracy reaching 83.08% and 80.50% for non-real-time and real-time operations. In comparison, applying max pooling results to recalls reached 89.55% and 84.93% for non-real-time and real-time operations, respectively. This paper shows that complex action decomposition is deemed to be an appropriate method through the comparable performance with existing efforts that have not considered violence as complex actions implying a new perspective for automatic violence detection in intelligent surveillance systems.

Keywords Automatic violence detection · Activation caching · Complex action recognition · Real-time systems · Multi-stream networks

1 Introduction

Rapid urbanization and economic development are present in many communities, especially in developing countries, while increased diversity in the population is seen in developed countries. However, the increase in population in a community would increase the possibility of crime, while the presence of specific business establishments may attract crime (Bernasco et al. 2017), and diversity may play a role in either the increase or decrease of crime generation and eyes on the street (Kim and Hipp 2021). Human factors such as the increased presence of citizens may also decrease crime and can be augmented with surveillance systems to enhance guardianship in the area (Long et al. 2021; Jang et al. 2018).

Automatic Violence Detection (AVD) is a specific and special field of study in the Human Action Recognition (HAR) problem in computer vision that would further improve surveillance technologies in reducing crime. Like many action recognition problems, the approach to analyzing violence in videos is sequential since actions are continuous.

✉ Cheng-Chang Lien
cclien@chu.edu.tw

Dylan Josh Domingo Lopez
djlopez.ce@tip.edu.ph

[1] Chung Hua University, Hsinchu, Taiwan

[2] Chung Yuan Christian University, Taoyuan, Taiwan

[3] Borq Technologies, Inc., Manila, Philippines

[4] Technological Institute of the Philippines - Quezon City, Quezon City, Philippines

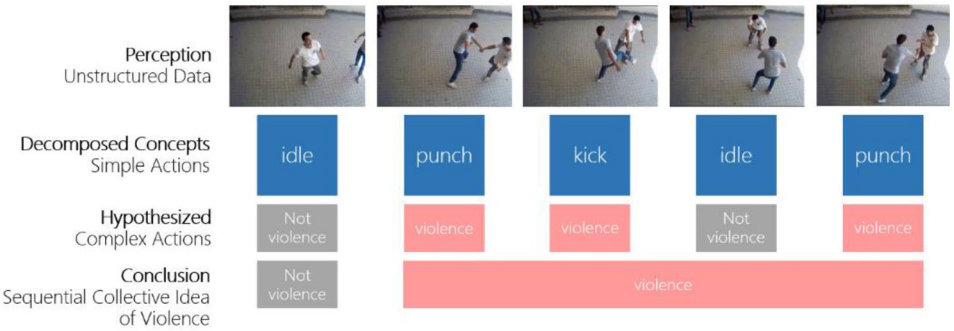

Fig. 1 Representation of violent actions in surveillance as complex actions

However, treating violent actions the same as all other actions could lead to oversimplification of the complexity of violent motion. Further in Acar et al. (2016) that a single violence classifier may overgeneralize the representation of violent scenes, whereas different scenarios under the concept of violence may exist on separate feature subspaces. AVD, specifically for surveillance videos, is more complex than it seems, whereas not only the input data is high-dimensional and sequential, and the violence class is also complex. Atomic or simple actions such as punching, kicking, swinging bats, or stabbing with a knife are collectively considered violent actions, which makes violence a complex action. This paper further proposes that violent actions in surveillance videos are considered complex actions. Referring to Fig. 1, complex action recognition for violence detection starts with the perception of unstructured data, specifically from surveillance feeds. Each frame from the video feed would include at least one instance of a simple action identified. Herein called the decomposition phase, wherein the decomposed actions are sub-concepts of violence. In the hypothesis phase, simple actions are further classified as complex actions using various methods such as signal processing or bag-of-words (Jung and Hong 2017). Lastly, the conclusion phase interpolates actions between scenes to determine the continuity or limit of the violent action sequence.

The popularity of AVDs has been increasing slowly and progressing with the improvement of deep learning. The early implementations of AVD utilized hand-crafted features for traditional machine learning methods (Bermejo et al. 2011; Hassner et al. 2012; Mahadevan et al. 2010; Zhang et al. 2017; Ehsan 2018; Moreira et al. 2017) and later used deep learning techniques (Baba et al. 2019; Accattoli et al. 2020; Traoré and Akhloufi 2020; Song et al. 2019; Samuel et al. 2019; Singh et al. 2017a; Abdali and Al-Tuma 2019; Roman and Chávez 2020; Lopez and Lien 2020). Violence detection is one of the complicated challenges in HAR due to the variety and complexity of violent actions. Due to its data source's complexity and unstructured nature, deep learning techniques are sought to solve them. There is a broad implication of AVDs in real-life spanning from censorship in media (Saad et al. 2022; Khalil et al. 2021) and entertainment to public safety and surveillance. However, real-time implementation and edge AI deployment would seem rather difficult due to the hardware requirements, complexity, and size of the models recently developed in the field.

Previous works in AVD are improving in terms of accuracy, but few would focus on their operational speed. This paper proposes a novel two-stage complex action recognition framework for automatic violence detection. The advantage of the proposed work from similar efforts in complex action recognition is the operational speed needed for industrial use and shows advantage from other automatic violence detectors is that the proposed method has less risk of over-generality and concept drift since the basis of the training data is directed to the simple actions related to violence instead of the general concept of violence itself. The system focuses on the implementability and modularity of AVDs in real-time implementations balancing speed and accuracy. Specifically, the contributions of the paper are:

1. Demonstrating real-time violence classification using a two-stage complex action recognition; and
2. Implementation of variable activation caching to interpolate actions between scenes in surveillance videos.

2 Related work

Human Action Recognition (HAR), or simply action recognition, is a task in computer vision that focuses on understanding and classifying human actions in unstructured data. However, many interpretations exist regarding how actions would be perceived or represented. This may include still images (Dehkordi et al. 2022), frame sequences, skeleton graphs (Bai et al. 2022; Liu et al. 2022a), sensor data (Zhao et al. 2020; He et al. 2017; Chen et al. 2016; Wei and Kehtarnavaz 2020; Chao et al. 2022; Dawar and Kehtarnavaz 2018), and spatiotemporal features. Action recognition is commonly fed with high-dimensional sequential data. The dimensionality of the input data would then give the

direction in the design of computer vision architecture to learn the feature space for the corresponding actions.

Deep learning has been widely used for learning the feature space of actions, such as the use of two-dimensional convolutional neural networks (2D CNN) (Dehkordi et al. 2022; Carreira and Zisserman 2017) for spatial features, optical flow processing (Kurban et al. 2022) for temporal features, three-dimensional CNNs (Accattoli et al. 2020; Liu et al. 2018) and two-stream networks (Ali and Taylor 2018; Saha et al. 2016; Singh et al. 2017b; Han et al. 2019; Feichtenhofer et al. 2016) for both spatial and temporal features. Recurrent neural networks (RNN) coupled with CNNs are also used to capture the spatial features of the input data in a sequential time-series manner (Traoré and Akhloufi 2020; Singh et al. 2017a; Abdali and Al-Tuma 2019). Mixed methods in implementing HARs can be extended to other methods such as interval type-3 fuzzy systems. Interval type-3 fuzzy systems are often used to predict time-series data (Castillo et al. 2022; Cao et al. 2021). HARs in deep learning oftentimes use gradient-based optimizers such as Adam and gradient descent, an alternative implementation of HARs can be employed considering non-gradient-based nature-inspired optimization algorithms (Yousefi and Loo 1509; Xu et al. 2016; Vanchinathan and Selvaganesan 2021; Vanchinathan and Valluvan 2018; Vanchinathan et al. 2021). However, given the time-series nature of actions, there can be future directions for human action classification prediction, especially with dealing with the other descriptors for action recognition, such as motion energy and spatial features.

2.1 Complex action recognition

Current action recognition models are significant in space and have high-order complexities. Such complexity is due to the high dimensionality of the input data and the effort to distinguish actions' spatial and temporal features in unison. This action recognition problem can be rooted in the same problem in feature engineering complex concepts. Some actions can be considered complex in concept wherein it can be considered super-classes of other more atomic actions (Hussein et al. 2019; Wang et al. 2016; Yi et al. 2017; Yeung et al. 2018; Liu et al. 2016a; Bacharidis and Argyros 2020). Given that the universal action feature space is represented by the union of the simple and complex action feature spaces seen in (1):

$$S = \{s|s \in \mathbb{R}^k\}; \quad C = \{c|s \in \mathbb{R}^k\} \tag{1}$$

whereas S is the feature subspace of simple action given s is a single feature vector representing an action while C is the feature subspace of complex actions containing feature vectors c for a single complex action class. Complex action recognition addresses the ambiguity of action super-classes (Dehkordi et al. 2022) where the actions with broader concepts are identified by their underlying simple actions (Hussein et al. 2019; Liu et al. 2021). Liu et al. (2022b) introduced common and difference dictionaries which represent the feature space of simple and complex actions.

$$D = \{d|d \in S \cap C\}; \quad D^c = \{d^c|d^c \in C \wedge d^c \notin S\} \tag{2}$$

whereas d and d^c represented individual feature vectors of dictionaries D and D^c. It can then be interpreted that the common dictionary D contains features from simple actions and complex actions while the difference dictionary D^c contains feature spaces exclusive only to complex actions that cannot be represented with simple action features. This supports the idea that complex actions serve as a super-class that includes the concepts of simpler actions.

Complex actions can also be described with other descriptors aside from sub-classes of simple actions. These descriptors can include temporal and spatial features (Jung and Hong 2017; Yi et al. 2017; Liu et al. 2016a) that are also generally inherent in two-stream architectures for HAR (Kurban et al. 2022; Singh et al. 2017b).

2.2 Automatic violence detection

There is a multitude of ways to tackle violence detection problems. These include the use of hand-engineered features such as Space–Time Interest Points (STIP), Histogram of Gradients (HOG), and Histogram of Optical Flows (HOF) algorithms using spatial-level matching and bag-of-words methods (Bermejo et al. 2011; Hassner et al. 2012; Mahadevan et al. 2010; Zhang et al. 2017; Ehsan 2018; Moreira et al. 2017). The work of Bermejo et al. (Bermejo et al. 2011) is one of the most notable violence detection systems using engineered features through Motion SIFT (MoSIFT) features. Another notable work in violence detection is by Hassner et al. (2012), which introduces violent flow descriptors for crowd violence classification using Support Vector Machines. Temporal analysis can also be used as a strategy to find violence in videos, such as optical flows and motion energy histograms (Mahadevan et al. 2010). Previous research achieved only fair accuracies utilizing the Histogram of Optical Flows (Ehsan 2018; Garje et al. 2018). In recent progress, deep neural networks have proven great accuracy and efficiency in violence detections such as (Bermejo et al. 2011). The use of deep learning has also improved the performance of AVD models through 2DCNNs (Baba et al. 2019) for spatial analysis and combinations of RNNs and CNNs (Traoré and Akhloufi 2020; Singh et al. 2017a; Abdali and Al-Tuma 2019; Roman and Chávez 2020; Liu et al. 2021)

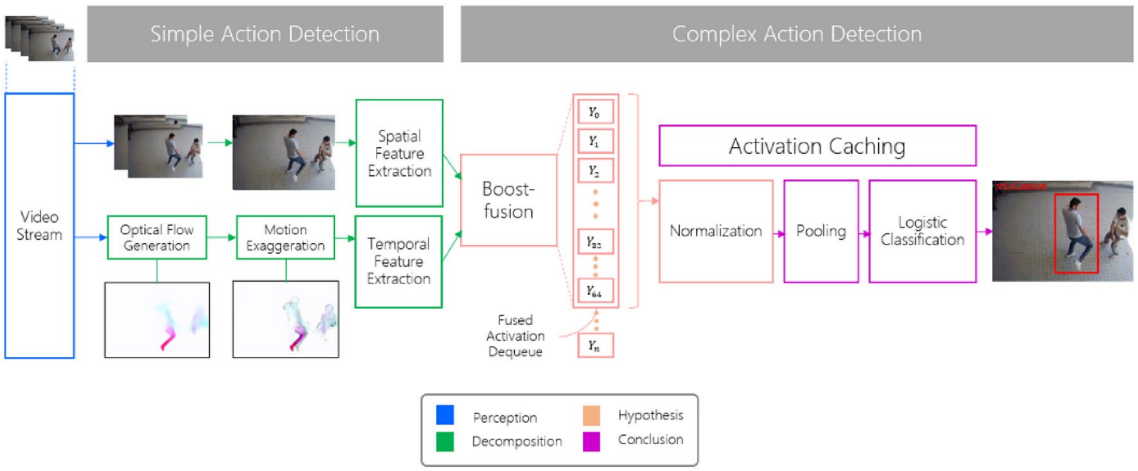

Fig. 2 Two-stage complex action recognition pipeline

for sequential spatial analysis. Spatiotemporal analysis for AVDs was also achieved using two-stream architectures (Lopez and Lien 2020), 3DCNNs (Accattoli et al. 2020; Samuel et al. 2019), and transformer networks (Mazzia et al. 2022).

In the societal context, automatic violence detection increases safety and lowers the risk of harm to the public. In contrast, all the previously mentioned works focused on public safety using surveillance cameras. Another fascinating field of application is behavioral science, wherein violence and aggression are exciting topics to be married with technology. The juncture of computer vision and behavioral science leads to a mix of understanding and predicting violence and aggression using behavioral theories based on action cues such as the implementations in Khan et al. (2018); Saif et al. 2019).

3 Proposed method

The proposed framework for automatic violence detection is seen in Fig. 2, wherein it also highlights the complex action recognition phases previously discussed. The video stream is considered the input to the system as part of the *perception* phase. The first stage of the AVD is the simple action detection which is also the *decomposition* phase of the framework. The simple action detection used in this framework is a two-stream architecture that performs spatiotemporal analysis on the incoming input data. The second stage of the pipeline is the complex action detection which comprises the hypothesis and conclusion phases. The second stage implements feature fusion of the activations produced from the first stage and stores the sequential detections in a dequeue. The dequeue is then subjected to the proposed activation caching algorithm to further classify the prediction sequences as violence or non-violence.

3.1 Simple action detection

The first stage of the AVD focuses on the decomposition of actions into simple actions. The simple actions are identified using spatiotemporal analysis like many action recognition frameworks. Specifically, spatial and temporal feature extraction on frame pairs $Z = \left(F_{n-1}, F_n\right)$ from the incoming video stream is carried out. The main architecture for the *decomposition* phase implements a two-stream architecture for spatiotemporal analysis. Each stream uses a YOLO backbone as the object/action detector framework. Each image Z is fed into a YOLO detector to determine the spatial features of the simple action in the frame. Consequently, action silhouettes I_ζ first obtained and then fed into a separate YOLO detector to obtain the temporal features of the simple action.

3.1.1 Spatial feature extraction

The proposed architecture integrates a single-stage CNNs as (Ali and Taylor 2018; Saha et al. 2016; Singh et al. 2017b) for action localization and classification of simple actions (i.e., run, walk, punch, or kick). Specifically, through an object detector backbone (Liu et al. 2016b; Redmon et al. 2016; Ren et al. 2017; Bochkovskiy et al. 2020). The frame F_n from Z is fed into a Spatial YOLO to extract the spatial features to create the spatial activation tensor A_s which

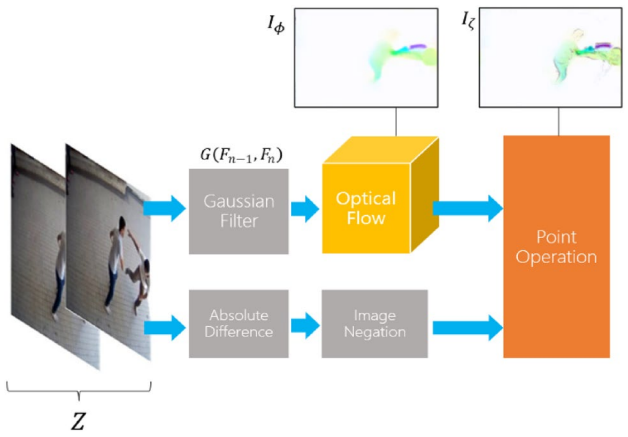

Fig. 3 Action silhouette generation

has a dimension of $(13, 13, b \times C)$ wherein b is the number of bounding boxes after detection and C are the number of classes.

3.1.2 Temporal feature extraction

Optical flow images are considered temporal features for most two-steam action recognition architectures (Ali and Taylor 2018; Saha et al. 2016; Singh et al. 2017b). Fast Dense Inverse Search (Brox et al. 2014; Kroeger et al. 2016) is used in Ali and Taylor (2018) to achieve real-time analysis but compensates for the model's accuracy. For better accuracy, (Saha et al. 2016) used the more accurate FlowNet2 (Ilg et al. 2017). Action intensity and the variation of the direction of motion are considered primary features for determining violence. The proposed implementation has accurate and real-time modes similar to Singh et al. 2017b. The implementation's accurate mode utilizes LiteFlowNet (Hui et al. 2018), while the real-time mode also utilizes Fast Dense Inverse Search. LiteFlowNet with the sintel model was used to generate the optical flow images I_ϕ obtained as:

$$I_\phi = \Phi_{LFN}\big(G\big(F_{n-1}, F_n\big)\big) \tag{3}$$

whereas $G()$ is a smoothing function with a 3×3 Gaussian filter. F_n is an image frame, whereas n denotes the temporal order of the frame. Φ_{LFN} is the LightFlowNet function to get an optical flow image using an image pair input. I_ϕ is the image produced by running the for smoothing then to the optical flow estimation neural network as seen in Fig. 3. Optical flow images I_ϕ are then transformed as action silhouette images I_ζ. The inputs for the temporal YOLO are the I_ζ to produce the temporal activation tensor A_T.

3.2 Action silhouettes

Key features to be considered in violence recognition are action intensity and the variation of the direction of motion. However, optical flow estimation alone would not expose or exaggerate these intensities and variations. With this, action silhouettes generation is proposed as an image feature for violence recognition.

As shown in Fig. 3, motion exaggeration is achieved by first accepting the frame pair Z and running through preprocessing such as smoothing and getting the negated image of the absolute difference of Z before feeding it into the optical flow estimator and the silhouette overlay algorithm. The action silhouette optical flow overlay is obtained by:

$$I_{\zeta(x,y)} = I_{\phi(x,y)} \cdot \left|(L-1) - Z_{(x,y)}\right| \tag{4}$$

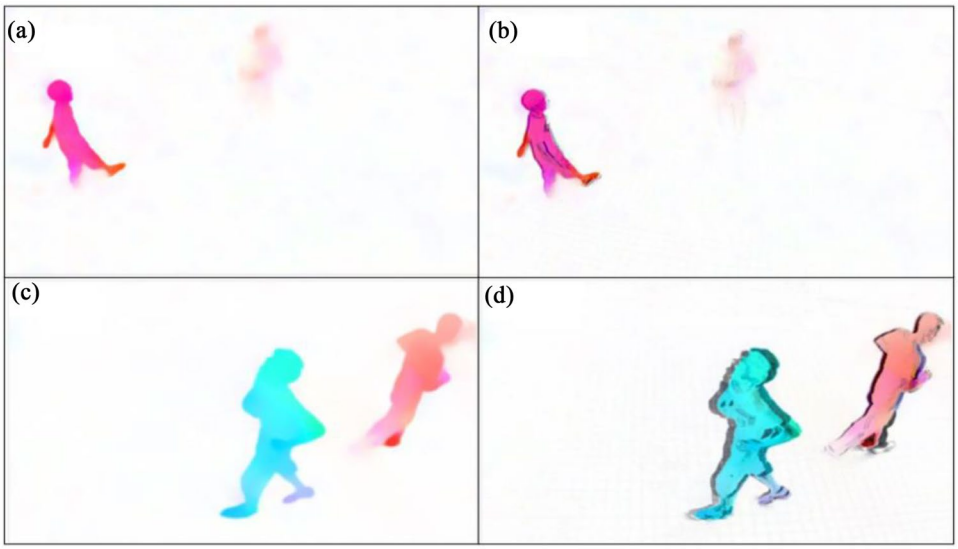

Fig. 4 Demonstrating motion exaggeration with action silhouettes

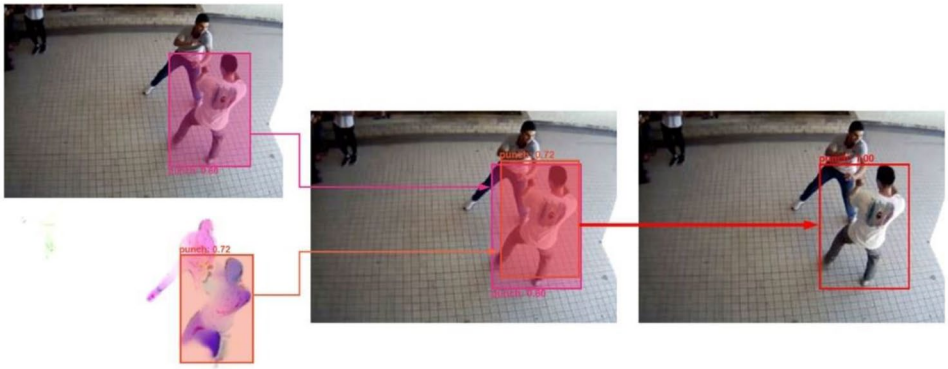

Fig. 5 Feature fusion of the spatial and temporal activations using the boost-fusion algorithm

whereas I_{ζ} is an action silhouette image formed by overlaying each pixel of the optical flow image $I_{\phi(x,y)}$ by executing a point operator equivalent to a logical AND with the absolute negative transform of $Z_{(x,y)}$ with respect to the max value L. In Fig. 4a and c, the cyan gradient indicates the leftward direction while the magenta gradient indicates the rightward movement. Both objects have solid opacity, which limits the amount of information indicating the intensity of the motion. Violent activities could be identified by exaggerated values such as the spike of motion or the variability of the movement. To produce such exaggeration, a silhouette overlay algorithm on the optical flow images was applied to produce action silhouettes adapted from Lopez and Lien (2020). The exaggeration effect is seen as a silhouette in the optical flow induced by the speed difference between running and walking actions can be seen in Fig. 4d and b respectively.

3.3 Feature fusion

The Simple Action Detection and Complex Action Detection stages are the two main parts of the proposed framework. Simple Action Detection consists of the object detector pipelines that are processed in parallel, thus making the two-stream architecture. The activations of each pipeline are merged through an early feature fusion. In summary, the Simple Action Detection stage is used to fuse and interpret high-level features to simple actions while the Complex Action Detection stage interprets the abstraction of simple actions as low-level features of complex actions (Fig. 5).

3.3.1 Boost-fusion algorithm

Early fusion was applied for the spatial and temporal activations using an approach similar to the fusion technique done by Singh et al. 2017b to obtain a combined simple action. The spatial bounding box retains the original fusion method proposed in Saha et al. (2016) and uses a boost-fusion:

$$s_c^*\left(b_i^s\right) = \begin{cases} s_c\left(b_i^s\right) + s_c\left(b_{\max}^t\right) \times \mathrm{IoU}\left(b_i^s, b_{\max}^t\right) & \mathrm{IoU} \geq \tau \\ s_c\left(b_i^s\right) + s_c\left(b_i^t\right) \times \mathrm{IoU}\left(b_i^s, b_i^t\right) & \text{otherwise} \end{cases} \tag{5}$$

whereas s_c is the softmax augmentation of a bounding box, $s_c^*(b_i^s)$ is the boost-fusion softmax score of the spatial bounding box. While b_i^s is the bounding box from the spatial object detector, b_i^t is the bounding box from the temporal object detector, and $b_{\max}^t$ is the bounding box from the temporal object detector with maximum overlap withb_i^s. The softmax of b_i^s remains while the softmax b_i^t is weighted by the Intersection over Union of b_i^s and $b_{\max}^t$ given that IoU is greater than the set threshold$\tau = 0.7$. If the IoU is less thanτ, the weighted score of the temporal detection is weighted

Fig. 6 Action degree mapping scale

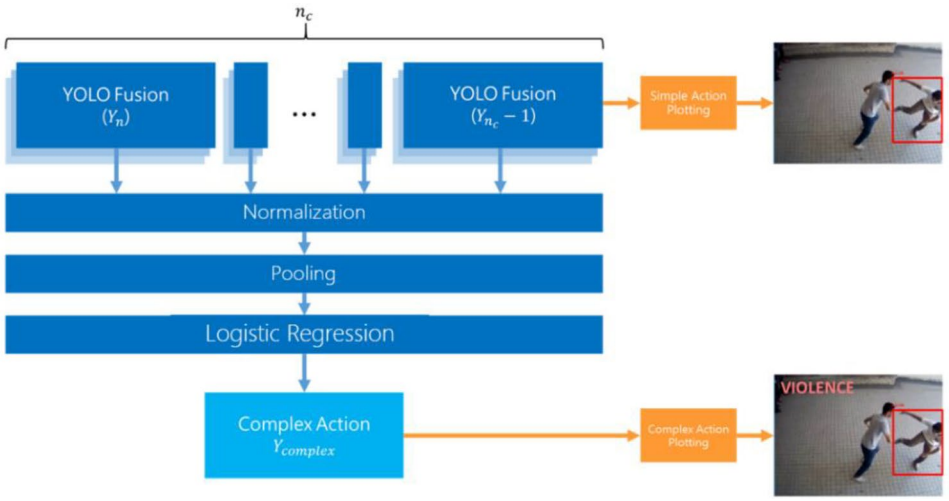

Fig. 7 A detailed view of the complex action detection section of the architecture

Fig. 8 Decomposed actions plotted with the localized simple action and the violence class

with the IoU of b_i^s and b_i^t.This helps to maximize the fusion between detected bounding boxes.

3.3.2 Action degrees

Action detections are ranked and mapped according to their degree of violence, as seen in Fig. 6, after obtaining the fused scores and bounding boxes. The corresponding action label values for this research are pre-determined to test the effectiveness of simple action components in determining violence. Determining the level of violence in simple actions are beyond the scope of this research but is a possible development of the research.

4 Complex action detection

The thought process of human comprehension of complex topics involves breaking the context into more understandable parts to further analyze the actions individually and recombine their essences to deduce a key concept. Such a thought process is analogous to the proposed system, which once simple actions are obtained acting as low-level features. Further analysis of each of the activations could also be done to determine the complex actions in which each of the predicted labels underlies.

The Complex Action Detection stage in Fig. 2 is responsible for simplifying the activations further by getting their essence and performing the generalization of the concept by predicting their respective complex action class. The *decomposition* phase in Fig. 1 suggests the concept of action decomposition by breaking down a video into simple actions and further classifying simple actions into complex actions. Comparable with (Hussein et al. 2019) and (Wang et al. 2016), simple actions are treated as motion atoms or phrases considering the actions as a temporal label and a fundamental feature for determining complex actions.

4.1 Prediction/activation variable caching

High-level features of abstraction are further analyzed to produce more specific information. Activation cache consisting of n_c fused activations from the Complex Action Detection stage were then utilized. Referring to Fig. 7, the bounding boxes of fused activations are first plotted to the current frame before being cached for complex action detection then mapped to their corresponding action degrees.

The activation cache is a sliding window for the predictions to smoothen the complex prediction. A visual approximation is shown at Fig. 8 in the results section, in which a cache with size $n_c = 32$ is simulated with a hypothetical signal. The ground truth degree of violence per frame is plotted against the signals wherein it marks the area that corresponds to violence in orange.

Several pooling options were tested to determine suitable pooling strategies for violence detection in real-time applications. The pooling techniques in this paper are average pooling, max pooling, and median pooling.

Algorithm 1. Complex Action Classification

```
procedure COMPLEX(Y^T_simple)
    varcache := Y^T_simple                                  Y_simple at time T
    if length(varcache) ≥ n_c then                          dequeue activations
        Y^T_simple := map(Y_simple, action_degree)
        Y^T_norm := normalize(Y^T_simple)                                  (6)
        e := pooling(Y^T_norm)                                             (7)
        Y_complex := σ(e)                                                  (8)
        varcache.pop()
    end if
    return Y_complex
end procedure
```

4.2 Complex action classification

The algorithm for the complex action classification with activation caching is presented in Algorithm 1. Before getting the essence of the activation cache, the normalize the activations are obtained by getting the normalized vector using (6).

$$Y_{norm} = 2 \cdot \frac{Y_{fused} - \min(Y_{pred})}{\max\left(Y_{pred} - min\left(Y_{pred}\right)\right)} \tag{6}$$

whereas Y_{norm} is the normalized essence tensor with values ranging between +1.0 and −1.0. Y_{fused} is the activation cache consisting of fused predictions. $\min(Y_{pred})$ and $\max(Y_{pred})$ denotes the minimum and maximum of the range of the class indices. The pooling operation for detecting violent actions is further defined using (7).

$$e = pool_{cache}(Y_{norm}) \tag{7}$$

whereas e is the scalar essence of the activation. The normalized activations are subjected to a 1-D pooling layer. In this implementation, average pooling layer and a max-pooling layer were tested.

Finally, a sigmoid function onto the complex action as a logistic classifier using (8). The fundamental machine learning algorithm is chosen due to the ease of separability of the complex action features contributed by the degrees of violence and to reduce the total complexity of the network.

$$Y_{complex} = \sigma(e) \tag{8}$$

5 CHU surveillance violence dataset

The existing datasets at the time this research was developed for Violence Detection (Perez et al. 2019; Blunsden and Fisher 2010; Patino et al. 2016) are incompatible with violence detection using modern surveillance cameras. Most of the videos in the existing datasets are in non-RGB, moving, or recorded using mobile phones. The abrupt movement in recording would make the optical flow inaccurate or fail to capture the intensity of the motion vectors. Datasets such as ARENA (Perez et al. 2019), BEHAVE (Blunsden and Fisher 2010), and UCSD (Patino et al. 2016) would be most suitable for violence detection systems since the data are extracted or collected through actual surveillance cameras.

There is also an imbalance between violent and non-violent actions (Carreira and Zisserman 2017) in all existing datasets. To the researchers' knowledge, no known datasets for violence detection consider violence a complex action. A custom dataset is then used for static surveillance of violent actions as the CHU Surveillance Violence Dataset (CSVD).[1] The CSVD includes RGB frames for spatial analysis and optical flow images for temporal analysis. The dataset consists of 12 simple action classes: idle, run, walk, high-five, wave, fight-pose, shove, grapple, punch, kick and melee weapon attack.

[1] The dataset is available at http://ieee-dataport.org/2662.

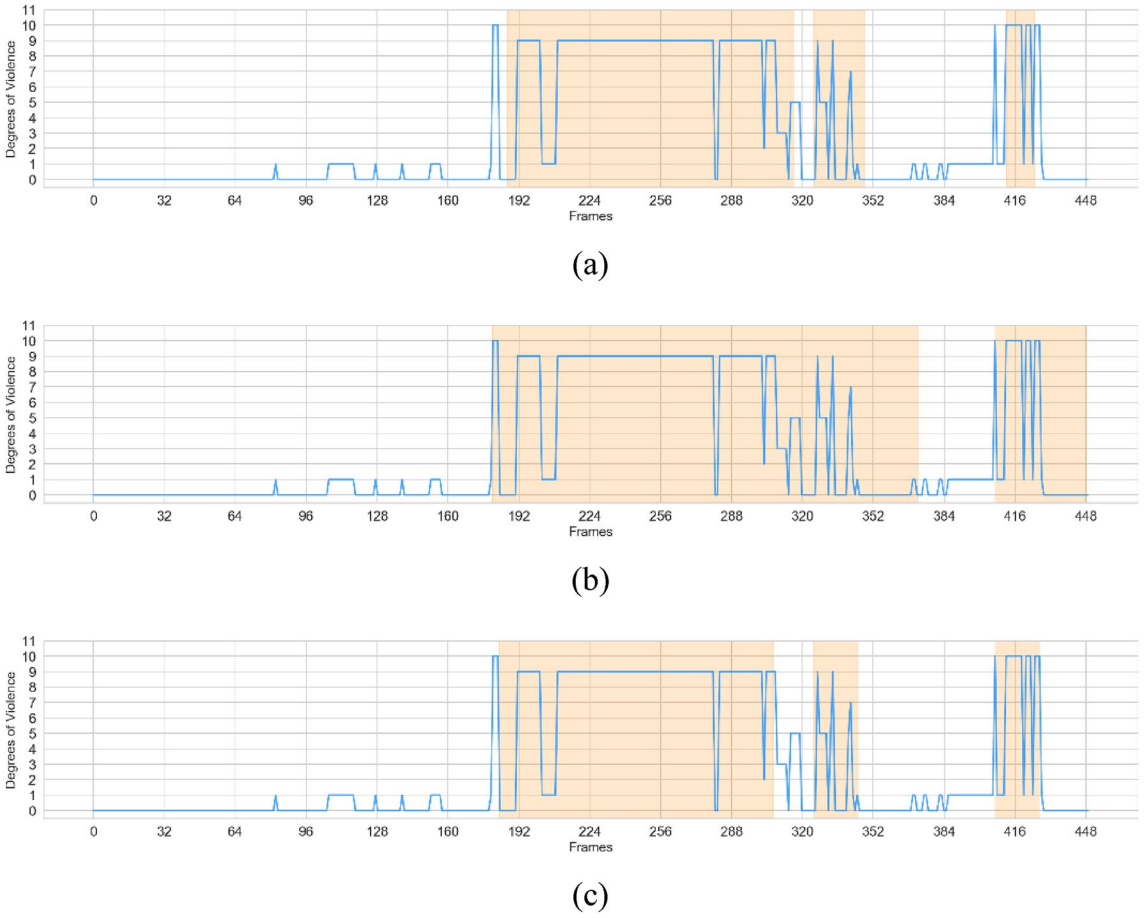

Fig. 9 Comparison of activation caching pooling techniques

6 Experimental results

To measure the performance of the system, the experimental results for simple and complex action recognition are separated. For the complex action detector, accuracy and recall were used to measure its performance with a focus on the recall metric to determine its effectiveness for safety–critical tasks. The developed systems are compared using the BEHAVE dataset with other violence recognition efforts. The overall performance speed of the system is also measured for its operability in practical implementation by determining its inference speed in frames per second (FPS). Several variants of the framework with accurate OF only (OF), action silhouettes (ASilh), and real-time implementation (RT) were also tested. No direct comparison for the accuracy of the developed model since previous efforts do not consider violence detection as a complex action recognition specifically for static surveillance systems and no other publicly available dataset is suitable for such an approach.

6.1 Activation caching

A constant cache size of 32 to identify different detection results, as seen in Fig. 9 for all pooling techniques. In Fig. 9a, average pooling was simulated in which it could be predicted that it will slightly decrease in accuracy since it will only fully determine the violence if most of the frames in the cache have high degrees of violence. This is in contrast with Fig. 9b which has the same n_c but uses max pooling wherein it determines the violence as soon as it enters the cache. The median pooling in Fig. 9c is also illustrated where it performs slightly better than average pooling. The assumed difference among the pooling techniques is that max pooling will be better at determining true positives but will have a significant number of false positives and negatives. Average pooling will have fewer false negatives and positives but has fewer true positives compared to max pooling. Median pooling can be assumed to be the better version of average pooling. However, it

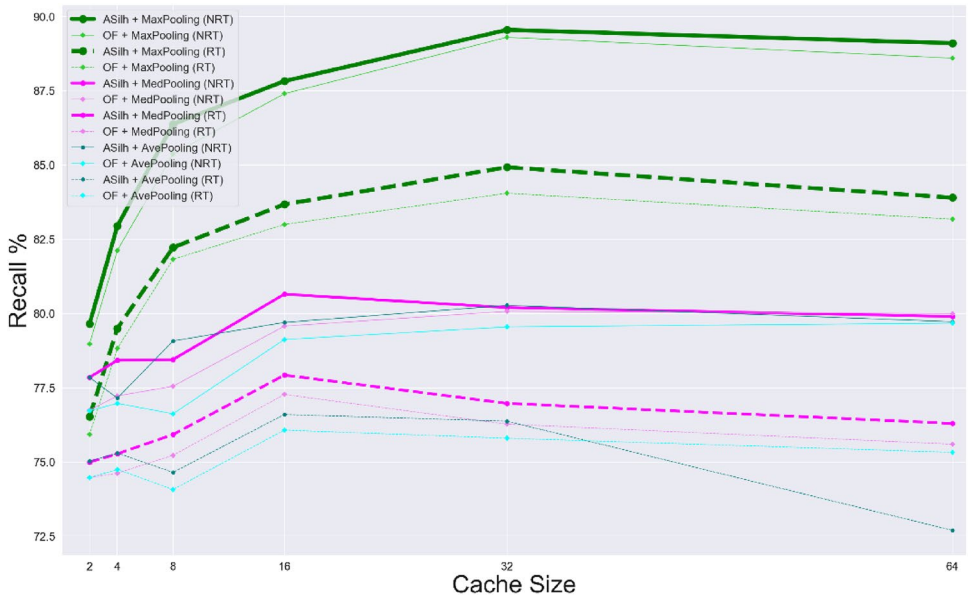

Fig. 10 AVD Recall corresponding to the activation cache size n_c

Table 1 Performance Comparison on the BEHAVE Dataset

Implementation	Accuracy (%)	Recall	Operational speed (FPS)	Dataset action complexity
Bermejo et al. (2011)†	62.0	n/a	n/a	Mixed
Hassner et al. (2012)†	82.02	n/a	n/a	Mixed
Zhang et al. (2017)†	87.17	n/a	n/a	Mixed
Baba et al. (2019)*	77.90	**100.00%**	**25**	Mixed
Lopez and Lien (2020)*	80.15	81.40%	**21**	Simple-complex
Ours-OF (AvePooling64)*	81.63	79.68%	11	Simple-complex
Ours-OF (MaxPooling32)*	78.55	**89.30%**	11	Simple-complex
Ours-OF (MedPooling32)*	82.90	80.08%	11	Simple-complex
Ours-ASilh (AvePooling16)*	82.20	80.28%	11	Simple-complex
Ours-ASilh (MaxPooling32)*	77.90	**89.55%**	11	Simple-complex
Ours-ASilh (MedPooling16)*	**83.08**	80.65%	11	Simple-complex
Ours-ASilh + RT (AvePooling16)*	80.10	76.60%	21	Simple-complex
Ours-ASilh + RT (MaxPooling32)*	76.68	**84.93%**	21	Simple-complex
Ours-ASilh + RT (MedPooling16)*	**80.50**	77.93%	**21**	Simple-complex

†These methods are implemented using traditional computer vision and machine learning

*Implements very deep CNN architectures

will still have less recall compared to max pooling. A constant cache size of 32 to identify different detection results, as seen in Fig. 9 for all pooling techniques. In Fig. 9a, average pooling is simulated in which it can be predicted that it slightly decreases in accuracy since it only fully determine the violence if the majority of the frames in the cache have high degrees of violence.

In further detail, the activation cache is encoded as a dequeue data structure in which activations are appended from the fusion method until it reaches a size of n_c. Complex action can be done once the dequeue reaches the specified n_c.

6.2 Complex action recognition

The framework was tested with the BEHAVE dataset with variants of the developed framework. Action silhouettes (ASilh) and optical flow (OF) only in real-time (RT) and non-real-time (NRT) mode, as well as varying the 1D pooling layer between AvePooling and MaxPooling were also tested. These various configurations were tested against increasing values of cache sizes.

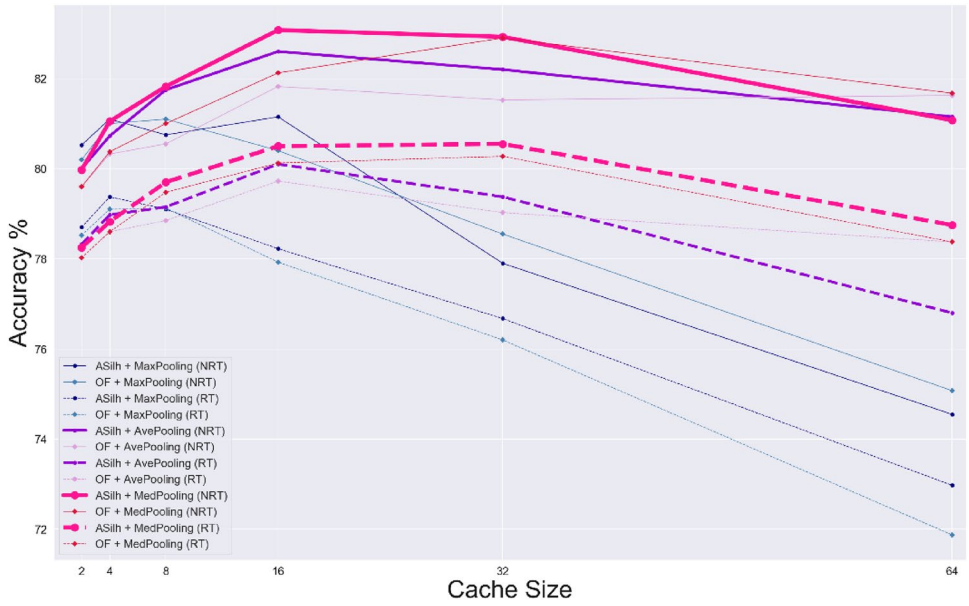

Fig. 11 AVD Recall corresponding to the activation cache size n_c

6.2.1 Recall performance

As seen in Fig. 10, the variant of the developed framework that produces the best recall uses MaxPooling for both real-time and not real-time modes. Accurate and not real-time framework variants have a recall score of 89.55% using action silhouettes and 89.33% using optical flow only, both at a cache size of 32. In comparison, real-time frameworks have a recall score of 84.93% and 84.05% for action silhouettes and optical flow only, respectively, both at cache size 32. The second-best option for a recall-based metric is Median which has a recall of 80.65% using action silhouettes and 80.08% using optical flow only, both at a cache size of 32. In contrast, real-time frameworks have a recall of 77.93% and 76.28% for action silhouettes and optical flow only, respectively, both at cache size 16. For all max pooling implementations, it can be seen that using action silhouettes brings improvement from using optical flows only. Increasing the cache sizes also contributes to the recall while increasing towards a cache size of 32 and decays further than 32. However, for median and average pooling implementations decay starts after $n_c = 16$. Best performances in terms of recall are in boldface in the recall column of Table 1.

6.2.2 Accuracy performance

In contrast with the recall, using Median pooling at $n_c = 16$ is seen to be more effective for accuracy for both real-time and not real-time mode as shown in Fig. 11. Accurate not real-time framework variants have an accuracy of 83.08% using action silhouettes and 82.90% using optical flow only. While real-time frameworks have an accuracy of 80.50% and 80.13% for action silhouettes and optical flow only, respectively. Such performances are noted in bold in the Accuracy column of Table 1. The second-best pooling technique for an accuracy-based metric is Average pooling at $n_c = 32$ which has an accuracy of 80.28% using action silhouettes and 79.68% using optical flow at $n_c = 64$. While real-time frameworks have an accuracy of 76.60% and 76.08% for action silhouettes and optical flow only $n_c = 16$ at respectively. For all configurations, it can be seen that using action silhouettes brings improvement from using optical flows only. Increasing the cache sizes also contributes to the recall while increasing towards a cache size of 16 and decays further than 16 while frameworks using max pooling decay faster compared to frameworks using average and median pooling.

6.2.3 Operational speed

The operation speed was compared with (Baba et al. 2019) and (Lopez and Lien 2020) aside from the developed framework's variants, which is the only previous work that has mentioned their framework's inference time. The best operational speeds can be seen in the Operational speed column of Table 1 that all of the developed variants have competitive speeds for real-time applications specifically with the RT variants of the framework with an operational speed of 21 FPS.

7 Conclusion

Violent actions are considered complex due to numerous underlying sub-concepts and variations. The proposed approach shows that violence can still be accurately classified by first decomposing violent actions into simple actions and then re-classifying to the binary class of violence.

This paper demonstrated automatic violence detection using a two-stage complex action recognition framework. The current work has seen improvement from previous works regarding recall and accuracy by applying variable activation caching to capture violence sequences in surveillance streams while maintaining operation speeds acceptable for real-time application. Experiments show that the developed framework's best configuration for real-time and non-real-time operations is applying action silhouettes for motion exaggeration and activation caching with a median pooling operation with a cache size of 16.

Implementing complex action recognition modularly in real-time has direct uses in in-built public safety systems since it can be cost-efficient solutions for communities, especially in developing countries. Moreover, this work provides insight into a more comprehensive perspective about violent actions expanding in surveillance and censorship in general media. Improvements can be made not just in the deployment or learning architecture but also in the vector embeddings on the identity of each violent action. In engineering, this work provides a foundation for the modularity aspect of possible intelligent surveillance systems. Health and medical applications can also benefit from the work, such as for behavioral analysis of children's aggression or asylum patients.

Funding National Science and Technology Council, MOST 109-2221-E-216-00-8, Cheng-Chang Lien

Data availability The CHU surveillance violence dataset is available publicly at: https://ieee-dataport.org/documents/chusurveillance-violence-detection-dataset.

References

Abdali AMR, Al-Tuma RF (2019) Robust Real-time violence detection in video using CNN And LSTM. 2019 2nd scientific conference of computer sciences (SCCS). p 104–108

Acar E, Hopfgartner F, Albayrak S (2016) Breaking down violence detection: combining divide-et-impera and coarse-to-fine strategies. Neurocomputing 208:225–237

Accattoli S, Sermani P, Falcionelli N, Mekuria DN, Dragoni AF (2020) Violence detection in videos by combining 3D convolutional neural networks and support vector machines. Appl Artif Intell 34(4):329–344

Ali A, Taylor GW (2018) Real-time end-to-end action detection with two-stream networks. 15th conference on computer and robot vision, CRV 2018. p 31–38

Baba M, Gui V, Cernazanu C, Pescaru D (2019) A sensor network approach for violence detection in smart cities using deep learning. Sen (switzerland) 19(7):1–17

Bacharidis K, Argyros A (2021) Extracting action hierarchies from action labels and their use in deep action recognition. 2020 25th international conference on pattern recognition (ICPR). p 339–346

Bai Z, Ding Q, Xu H, Chi J, Zhang X, Sun T (2022) Skeleton-based similar action recognition through integrating the salient image feature into a center-connected graph convolutional network. Neurocomputing 507:40–52

Bermejo E, Deniz O, Bueno G, Sukthankar R (2011) Violence detection in video using computer vision techniques. In: International Conference on Computer Analysis of Images and Patterns. p 332–339

Bernasco W, Ruiter S, Block R (2017) Do street robbery location choices vary over time of day or day of week? A test in Chicago. J Res Crime Delinq 54(2):244–275

Blunsden SJ, Fisher RB (2010) The BEHAVE video dataset: ground truthed video for multi-person behavior classification. Annal BMVA 2010(4):1–12

Bochkovskiy A, Wang CY, Liao HYM (2020) YOLOv4: optimal speed and accuracy of object detection

Brox T, Papenberg N, Weickert J (2014) High accuracy optical flow estimation based on a theory for warping. In 8th European conference on computer vision, vol. 3024. p 25–36

Cao Y, Raise A, Mohammadzadeh A, Rathinasamy S, Band SS, Mosavi A (2021) Deep learned recurrent type-3 fuzzy system: application for renewable energy modeling/prediction. Energy Rep 7:8115–8127

Carreira J, Zisserman A (2017) Quo vadis, action recognition? a new model and the kinetics dataset. IEEE conference on computer vision and pattern recognition, CVPR 2017. vol. 2017. p 4724–4733

Castillo O, Castro JR, Melin P (2022) Interval type-3 fuzzy aggregation of neural networks for multiple time series prediction: the case of financial forecasting. Axioms 11(6):251

Chao X, Hou Z, Mo Y (2022) CZU-MHAD: a multimodal dataset for human action recognition utilizing a depth camera and 10 wearable inertial sensors. IEEE Sens J 22(7):7034–7042

Chen C, Jafari R, Kehtarnavaz N (2016) A real-time human action recognition system using depth and inertial sensor fusion. IEEE Sens J 16(3):773–781

Dawar N, Kehtarnavaz N (2018) Action detection and recognition in continuous action streams by deep learning-based sensing fusion. IEEE Sens J 18(23):9660–9668

Dehkordi HA, Nezhad AS, Kashiani H, Shokouhi SB, Ayatollahi A (2022) Multi-expert human action recognition with hierarchical super-class learning. Knowl Based Syst 250:109901

Ehsan TZ (2018) Violence detection in indoor surveillance cameras using motion trajectory and differential histogram of optical flow. 8th International Conference on Computer and Knowledge Engineering (ICCKE), no. ICCKE. p 153–158

Feichtenhofer C, Pinz A, Zisserman A (2016) Convolutional two-stream network fusion for video action recognition. Proceedings of the IEEE conference on computer vision and pattern recognition. p 1933–1941

Garje PD, Nagmode MS, Davakhar KC (2018) Optical flow based violence detection in video surveillance. 2018 International conference on advances in communication and computing technology (ICACCT). p 208–212

Han Y, Chung SL, Chen SF, Su SF (2019) Two-stream LSTM for action recognition with RGB-D-based hand-crafted features and feature combination. IEEE Int Conf Syst Man Cybern SMC 2018:3547–3552

Hassner T, Itcher Y, Kliper-Gross O (2012) Violent flows: real-time detection of violent crowd behavior. IEEE international conference on computer vision and pattern recognition workshops. p 1–6

He W, Liu B, Xiao Y (2017) Multi-View action recognition method based on regularized extreme learning machine. 2017 IEEE international conference on computational science and engineering (CSE) and IEEE international conference on embedded and ubiquitous computing (EUC). p 854–857

Hui TW, Tang X, Loy CC (2018) LiteFlowNet: a lightweight convolutional neural network for optical flow estimation. IEEE international conference on computer vision and pattern recognition. p 8981–8989

Hussein N, Gavves E, Smeulders AWM (2019) Timeception for complex action recognition. 2019 IEEE/CVF conference on computer vision and pattern recognition (CVPR). p 254–263

Ilg E, Mayer N, Saikia T, Keuper M, Dosovitskiy A, Brox T (2017) FlowNet 2.0: evolution of optical flow estimation with deep networks. IEEE conference on computer vision and pattern recognition (CVPR)

Jang Y, Kim D, Park J, Kim D (2018) Conditional effects of open-street closed-circuit television (CCTV) on crime: a case from Korea. Int J Law Crime Justice 53:9–24

Jung HJ, Hong KS (2017) Modeling temporal structure of complex actions using bag-of-sequencelets. Pattern Recogn Lett 85:21–28

Khalil T, Bangash JI, Khan AW, Lashari SA, Khan A, Ramli DA (2021) Detection of violence in cartoon videos using visual features. Procedia Comput Sci 192:4962–4971

Khan SS, Ye B, Taati B, Mihailidis A (2018) Detecting agitation and aggression in people with dementia using sensors—a systematic review. Alzheimers Dement 14(6):824–832

Kim YA, Hipp JR (2021) Density, diversity, and design: three measures of the built environment and the spatial patterns of crime in street segments. J Crim Just 77:101864

Kroeger T, Timofte R, Dai D, Van Gool L (2016) Fast optical flow using dense inverse search. European conference on computer vision

Kurban OC, Calik N, Yildirim T (2022) Human and action recognition using adaptive energy images. Pattern Recogn 127:108621

Liu F, Xu X, Qing C (2016a) Temporal order information for complex action recognition. 2016a IEEE international conference on consumer electronics-China (ICCE-China). p 1–4

Liu W, Anguelov S, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016b) SSD: single shot multibox detector. Eur Conf Comput Vis 9905:21–37

Liu K, Liu W, Gan C, Tan M, Ma H (2018) T-C3D: temporal convolutional 3d network for real-time action recognition. 32nd AAAI conference on artificial intelligence, AAAI 2018. p 7138–7145

Liu Z, Yin Z, Wu Y (2021) MLRMV: multi-layer representation for multi-view action recognition. Image Vis Comput 116:104333

Liu J, Akhtar N, Mian A (2022a) Adversarial attack on skeleton-based human action recognition. IEEE Trans Neural Netw Learn Syst 33(4):1609–1622

Liu F, Xu X, Xing X, Guo K, Wang L (2022b) Simple-action-guided dictionary learning for complex action recognition. Neurocomputing 501:387–396

Long D, Liu L, Xu M, Feng J, Chen J, He Li (2021) Ambient population and surveillance cameras: The guardianship role in street robbers' crime location choice. Cities 115:103223

Lopez DJD, Lien CC (2020) Real-time human violent activity recognition using complex action decomposition. International computer symposium (ICS). p 360–364

Mahadevan V, Li WX, Bhalodia V, Vasconcelos N (2010) Anomaly Detection in Crowded Scenes. IEEE International Conference on Computer Vision and Pattern Recognition. p 1975–1981

Mazzia V, Angarano S, Salvetti F, Angelini F, Chiaberge M (2022) Action transformer: a self-attention model for short-time pose-based human action recognition. Pattern Recogn 124:108487

Moreira D, Avila S, Perez M, Moraes D, Testoni V, Valle E, Goldenstein S, Rocha A (2017) Temporal robust features for violence detection", IEEE Winter Conference on Applications of Computer Vision, WACV 2017. p 391–399

Patino L, Cane T, Vallee A, Ferryman J (2016) PETS 2016: dataset and challenge. IEEE international conference on computer vision and pattern recognition workshops. p 1240–1247

Mauricio Perez, Alex C. Kot, Anderson Rocha (2019) Detection of Real-world Fights in Surveillance Videos. IEEE international conference on acoustics, speech, and signal processing (ICASSP)

Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: unified, real-time object detection. IEEE international conference on computer vision and pattern recognition. vol. 2016-Decem. p 779–788

Ren S, He K, Girshick R, Sun J (2017) Faster R-CNN: towards real-time object detection with region proposal networks. IEEE Trans Pattern Anal Mach Intell 39(6):1137–1149

Roman DGC, Chávez GC (2020) Violence detection and localization in surveillance video. 2020 33rd SIBGRAPI conference on graphics, patterns and images (SIBGRAPI). p 248–255

Saad K, El-Ghandour M, Raafat A, Ahmed R, Amer E (2022) A markov model-based approach for predicting violence scenes from movies. 2022 2nd international mobile, intelligent, and ubiquitous computing conference (MIUCC). p 21–26

Saha S, Singh G, Sapienza M, Torr PHS, Cuzzolin F (2016) Deep learning for detecting multiple space-time action tubes in videos. In British Machine Vision Conference

Saif AFMS, Khan MAS, Hadi AM, Karmoker RP, Gomes JJ (2019) Aggressive action estimation: a comprehensive review on neural network based human segmentation and action recognition. Int J Educ Manag Eng 9(1):9–19. https://doi.org/10.5815/ijeme.2019.01.02

Samuel RDJ, Fenil E, Gunasekaran M, Vivekananda GN, Thanjaivadivel T, Jeeva S, Ahilan A (2019) Real time violence detection framework for football stadium comprising of big data analysis and deep learning through bidirectional LSTM. Comput Netw 151:191–200

Singh D, Merdivan E, Hanke S, Kropf J, Geist M, Holzinger A (2017a) Convolutional and recurrent neural networks for activity recognition in smart environment. In: Holzinger A, Goebel R, Ferri M, Palade V (eds) Towards integrative machine learning and knowledge extraction. Lecture notes in computer science, vol 10344. Springer, Cham

Singh G, Saha S, Sapienza M, Torr P (2017b) Online real-time multiple spatiotemporal action localisation and prediction. International conference on computer vision. p 3657–3666

Song W, Zhang D, Zhao X, Yu J, Zheng R, Wang A (2019) A novel violent video detection scheme based on modified 3d convolutional neural networks. IEEE Access 7:39172–39179

Traoré A, Akhloufi MA (2020) Violence Detection in Videos using Deep Recurrent and Convolutional Neural Networks. 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC). p 154–159

Vanchinathan K, Selvaganesan N (2021) Adaptive fractional order PID controller tuning for brushless DC motor using artificial bee colony algorithm. Results Control Optim 4:100032

Vanchinathan K, Valluvan KR (2018) A metaheuristic optimization approach for tuning of fractional-order PID controller for speed control of sensorless BLDC motor. J Circuits Syst Comput 27(08):1850123

Vanchinathan K, Valluvan KR, Gnanavel C, Gokul C, Albert JR (2021) An improved incipient whale optimization algorithm based robust fault detection and diagnosis for sensorless brushless DC motor drive under external disturbances. Int Trans Electr Energy Syst. https://doi.org/10.1002/2050-7038.13251

Wang L, Qiao Y, Tang X (2016) MoFAP: a multi-level representation for action recognition. Int J Comput Vision 119:254–271

Wei H, Kehtarnavaz N (2020) Simultaneous utilization of inertial and video sensing for action detection and recognition in continuous action streams. IEEE Sens J 20(11):6055–6063

Xu D, Xiao X, Wang X, Wang J (2016) Human action recognition based on Kinect and PSO-SVM by representing 3D skeletons as points in lie group. international conference on audio, language and image processing. p 568–573

Yeung S, Russakovsky O, Jin N, Andriluka M, Mori G, Fei-Fei L (2018) Every moment counts: dense detailed labeling of actions in complex videos. Int J Comput Vision 126(2–4):375–389

Yi Y, Cheng Y, Xu C (2017) Mining human movement evolution for complex action recognition. Expert Syst Appl 78:259–272

Yousefi B, Loo CK (2015) Bio-inspired human action recognition using hybrid max-product neuro-fuzzy classifier and quantum-behaved PSO. arXiv:1509.03789 [cs.AI]

Zhang T, Jia W, Yang B, Yang J, He X, Zheng Z (2017) MoWLD: a robust motion image descriptor for violence detection. Multimedia Tools Appl 76(1):1419–1438

Zhao Y, Xu D, Wang T, Ren Y (2020) Dynamic action recognition under simulated prosthetic Vision. 2020 International conference on networking and network applications (NaNA). p 417–421