Toward Interpretable Multimodal Violence Detection With Knowledge Distillation and Modality-Aligned Preprocessing

Wen-Dong Jiang, *Graduate Student Member, IEEE*, Chih-Yung Chang, *Member, IEEE*, Ming-Yang Su, Yue-Shi Lee, and Diptendu Sinha Roy, *Senior Member, IEEE*

***Abstract*—Social violence presents a compelling challenge to public safety, yet existing multimodal detection systems exhibit excessive reliance on RGB image semantics and opaque decision-making processes. Despite leveraging visual and auditory data, current models demonstrate RGB bias in feature prioritization, as evidenced by explainability analyzes, thereby limiting their generalization for behavioral understanding. Additionally, modality inconsistency and inefficient fusion mechanisms impair model transparency and training stability. To bridge these gaps, this study proposes modality-aligned preprocessing (VAJ) that structurally unifies visual-auditory features through conflict resolution and input optimization, explicitly suppressing color dominance while enhancing interpretable feature representations. Complementing this, we design DTVDS, an interpretable detection framework integrating knowledge distillation to transfer distilled behavioral insights from a cumbersome teacher network to an efficient student model. This dual strategy not only addresses computational overhead but also clarifies decision logic through simplified inference pathways. Evaluations on XD-Violence and UCF-Crime benchmarks demonstrate superior performance, with AP (89.64%) and AUC (88.35%) outperforming existing methods. Qualitative evaluations further validate interpretability, revealing modality-coherent attention maps and human-aligned rationale visualization. The proposed method advances violence detection by addressing persistent shortcomings in multimodal alignment and model explainability.**

***Index Terms*—Deep learning, interpretable machine learning, multimodal learning, Violence detection.**

Received 20 August 2024; revised 15 March 2025; accepted 2 June 2025. Date of publication 24 June 2025; date of current version 20 August 2025. This work was supported by the National Science and Technology Council of Taiwan through the Republic of China under Grant NSTC 112-2622-E-032-005. This article was recommended by Associate Editor O. Erdinc. *(Corresponding author: Chih-Yung Chang.)*

Wen-Dong Jiang and Chih-Yung Chang are with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei 25137, Taiwan (e-mail: wendongjiang@ieee.org; cychang@mail.tku.edu.tw).

Ming-Yang Su and Yue-Shi Lee are with the Department of Computer Science and Information Engineering, Ming Chuan University, Taoyuan City 333, Taiwan (e-mail: minysu@mail.mcu.edu.tw; leeys@mail.mcu.edu.tw).

Diptendu Sinha Roy is with the Department of Computer Science and Engineering, National Institute of Technology, Shillong 793003, India (e-mail: diptendu.sr@nitm.ac.in).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TSMC.2025.3576390.

Digital Object Identifier 10.1109/TSMC.2025.3576390

I. Introduction

Social violence is a key factor affecting public safety and stability. Violence detection is crucial for maintaining public order and reducing violent acts. In the early days, due to technological limitations, manual inspection of a large volume of video data was both time-consuming and inefficient. With the development of AI technology, many studies have begun to use AI to detect violent behavior in videos. From the perspective of behavior identification [1], violence detection is like behavior recognition [2], [3]. Violence detection technology has a wide range of applications in the real world, particularly in the fields of personal safety, home security, and public safety. For example, it can be integrated into wearable devices, such as smart glasses, smartphones, smartwatches, or dash cams to help users detect violent behavior in real time in dangerous environments and automatically send alerts. In home security monitoring, it can be used to detect domestic violence; even if the perpetrator attempts to lower their voice or conceal their actions, the system can still recognize abnormal behavior. In public places, especially in privacy-sensitive areas, such as fitting rooms, restrooms, and counseling rooms, traditional camera surveillance may not be suitable. However, violence detection systems can analyze audio to identify violent behavior while protecting individual privacy. However, the actual situation is much more complex, as the tone of abuse and threats is also part of violence detection. Therefore, the current violence detection tends to use multimodal models that integrate image, audio, and text to detect violent behavior.

In literature, many studies have proposed the multimodal framework, aiming to extract a variety of features from heterogeneous data. However, the multimodal [4], [5], [6], [7] may still encounter the following three issues. First, the imbalance in input data across different models may lead to problems of difficulty in convergence during training and incorrect prediction results [8], [9]. Fig. 1 gives the flaws that occurred due to data imbalance. As shown in Fig. 1(a), two girls are mocking another girl with insulting language, but the model mistakenly predicts it as nonviolent. This occurs because the video and audio data are imbalanced. That is, the video data contains more information than audio, resulting in the limited effectiveness of the audio modality.

The second issue of the multimodal is that the prediction results of two modalities may conflict with each other, leading

TABLE I
GLOSSAR

CAM	Class Activation Mapping
HD-Net	Hybrid Deep Network
OCSVM	One-Class Support Vector Machine
SOTA metrics	State-of-the-Art Metrics
C3D	Convolutional 3D Network
C3D + Self-Attention	An enhanced version of the C3D model that integrates self-attention mechanisms
MCL	Multimodal Contrastive Learning
MIL	Multiple Instance Learning
AR-MIL	Abnormal Ratio-based Multiple Instance Learning
AutoEncoder	A Type of Neural Network Used for Unsupervised Learning
τ	A Hyperparameter in the Loss Function
γ	A Hyperparameter Related to the Activation Function.
π	A hyperparameter Related to the Attention Mechanism.
A^l	Output or Activation of the **l-th** Layer of the Neural Network.

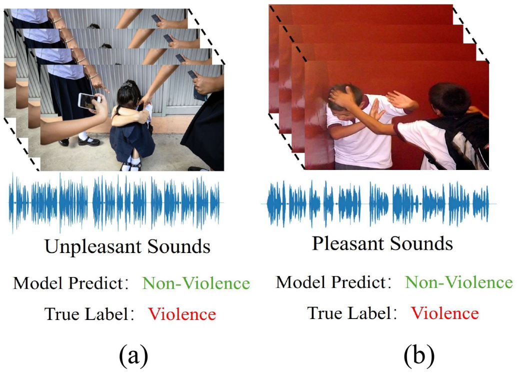


Fig. 1. Defects in multimodal models for violence detection. (a) Violent scene with unpleasant sounds incorrectly predicted as non-violence by the model. (b) Violent scene with pleasant sounds incorrectly predicted as non-violence by the model.

to errors in the predicted outcomes [10], [11]. As shown in Fig. 1(b), the two boys are fighting, but with cheerful music playing in the background, leading the model to also mistakenly predict it as nonviolent. This occurs because noise from different modalities can interfere with each other. That is, the cheerful music acts as the noise for the video, disrupting the accurate detection of violent behavior.

The third issue with multimodal models is interpretability [12], [13], [14], [15], [16], [17]. Although many recent studies have focused on improving the performance of models in violence detection tasks, there remains a lack of understanding and interpretability analysis of the models' decision-making processes. In the relevant literature, researchers have proposed various visualization-based methods to attempt to uncover the behavior logic of models under different input conditions. One commonly used technique is class activation mapping (CAM) [12], which generates heatmaps to highlight the regions of the input image that the model focuses on.

Fig. 2 presents visualization results of a deep learning model [18] designed for violence detection tasks. The figure contains four rows of content: from top to bottom, they are the original RGB input, interpretability analysis of the RGB image, grayscale image input, and interpretability analysis of the grayscale image. As shown in Fig. 2, the heatmaps generated using CAM technology indicate that when processing RGB images, the model's most intense red areas (considered to be violent parts) are primarily concentrated in regions of bodily movement, suggesting that the model may be correctly identifying violent behavior. However, when the same image is converted to grayscale and re-entered into the model, the model's focus shifts significantly. The red areas on the heatmap are no longer concentrated in the action areas but instead focus on darker pixel regions, such as black clothing or background. This phenomenon suggests that the model may rely more on RGB color information rather than actual motion features during its prediction process.

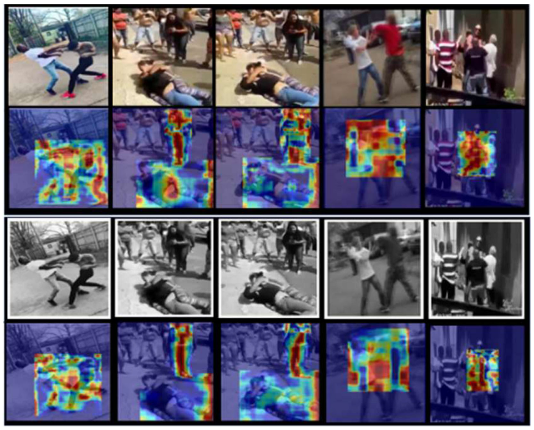


Fig. 2. Interpretable analysis by using CAM-based [12].

This observation reveals limitations in the interpretability of existing models designed for violence detection tasks: their decision-making logic does not always align with human understanding of behavior recognition. Although the model has undergone 217 h of video training and can achieve excellent results on specific datasets, its prediction process is not always based on reasonable behavioral features. This discrepancy may lead to misjudgments in real-world applications, particularly when dealing with varying lighting conditions or scenes lacking color information.

This article proposes the design of a unimodal network architecture that retains the advantages of multimodal systems, capable of capturing data features from different modalities while avoiding the three common problems in multimodal models. Specifically, this article introduces an innovative and interpretable data preprocessing method called VAJ (Video and Audio Joint). VAJ integrates high-quality representations of video and audio into a single image during the data preprocessing stage, allowing this image to be operated within a unimodal network architecture.

To realize this concept, this article proposes an innovative violence detection system called DTVDS. In the preprocessing stage, DTVDS synthesizes speech signals and images into a single picture through VAJ, and accurately captures and identifies temporal dynamics and spatial features in video data through DenseNet and Transformer models. Subsequently, knowledge distillation techniques are used to effectively transfer the temporal sequence knowledge acquired by the Transformer model to the CNN model. In this architecture, CNN acts as a learner, acquiring deep-temporal knowledge from the Transformer to achieve fast and accurate violence recognition with a more streamlined structure. This approach not only simplifies the model and improves interpretability but also enhances processing speed and real-time detection capabilities. The VAJ data preprocessing method and DTVDS system designed in this article can be deployed in scenarios involving street violence detection and campus bullying. When street fights or campus violence incidents occur, the cameras can quickly and accurately identify violent behavior, enabling the system to issue alerts as soon as violent events begin. This allows law enforcement to intervene promptly, minimizing potential harm to the greatest extent possible.

This article aims to address the following three research questions.

First, how can multimodal data, specifically speech and image signals, be effectively integrated into a unified representation to improve the accuracy and efficiency of violence detection systems?

Second, what are the challenges and limitations of current convolutional and self-attention-based models in violence detection, and how can these be addressed through knowledge distillation techniques?

Third, in what ways can a lightweight yet robust violence detection system be designed to enhance real-time detection capabilities while ensuring data privacy and interpretability?

The main contributions of this article are summarized in three aspects.

1) A novel data preprocessing method VAJ is proposed to address the issues with RGB images and to integrate essential information extracted from the multiple modalities. This powerful representation more accurately conveys scene semantics and action information.
2) Compared to the convolutional models, such as C3D or ResNet and the self-attention mechanisms commonly used in the current field of violence detection, the proposed DTVDS demonstrates superior semantic extraction and model generalization capabilities. This is because, during its application, the model inherits knowledge from the pretrained model through a knowledge distillation method. Experimental results achieved an AP value of 89.64% on the XD-Violence dataset and an AUC of 88.35% on the UCF-Crime dataset, surpassing existing methods.
3) The proposed VAJ and DTVDS methods are more lightweight in practical surveillance scenario deployments, with stronger interpretability, while also offering advantages in data and privacy protection. The proposed method is well-suited for real-life applications, such as campus violence detection and street brawls.

The remainder of this article is organized as follows. Section II discusses and compares previous relevant studies. Section III describes the Assumptions and problem formulation in detail. Section IV details the method and model proposed in this article. Section V provides the experiments and performance evaluation. Section VI provides lation and fature work The conclusion is discussed in Section VII.

II. Related Work

In this chapter, a review of some relevant studies in violence detection is presented. These studies are categorized into three groups: machine learning, deep learning, and multimodal deep learning methods. Definitions of symbols and terms are shown in Table I.

A. Violence Detection in Machine Learning

In literature, many studies adopted machine learning technologies for violence detection. First, the feature selection comes from manually designed methods, including histogram of oriented gradients (HOG), histogram of optical flow (HOF), scale-invariant feature transform (SIFT), and violent flows (ViF). These methods can extract features which can be taken as inputs of machine learning technologies. A variety of machine learning methods have been developed in the past years. For example, Bermejo et al. [19] explored the combined use of SIFT and STIP with the bag-of-words model in combat detection. Hessner et al. [20] used ViF to represent videos and then employed a linear SVM to classify videos into violent and nonviolent categories. Schölkopf et al. [21] employed an OCSVM to detect violent behavior in videos. The design of these manual features is based on human understanding of image characteristics. However, designing and selecting effective manual features often requires a significant amount of expertise and trial and error, which is not only time-consuming but also highly subjective. With the increase in dataset size and complexity, the performance of manually designed features often reaches a bottleneck.

B. Violence Detection in Deep Learning

A variety of deep neural networks (DNNs) have been widely adopted in violence detection tasks due to their outstanding performance. For instance, Sudhakaran and Lanz [22] utilized Convolutional long short-term memory (LSTM) networks to identify violent videos, achieving significant performance improvements compared to methods based on handcrafted features. Hanson et al. [23] applied Bidirectional Convolutional LSTM for violence detection. Peixoto et al. [24] designed two DNNs to learn spatial information from videos as a means to understand the definition of violence. Singh et al. [25] proposed a method combining sparse networks with deep learning networks for detecting violent behaviors in drone surveillance videos. Saltani et al. [26] contributed a public data set called UCF-Crime and implemented it using MIL. While these deep-learning models surpass the performance of traditional machine learning models, their effectiveness is often limited to single modalities. However, in real life, the occurrence of violent behavior often involves multiple modalities, including images, voice, text, and more. Similarly, compared to machine learning methods, the interpretability of deep learning models has been questioned.

TABLE II
COMPARISON WITH OTHER WORK

Model	Input	Dualmodal	Interpretable	privacy protection	Faster computation
Linear SVM	RGB	×	×	×	✓
OCSVM	RGB	×	×	×	✓
AutoEncoder	RGB	×	×	×	×
C3D	RGB	×	×	×	×
HD-Net	RGB+Audio	✓	×	×	×
C3D+ Self-Attention	RGB+Audio	✓	×	×	×
MCL	RGB+Audio	✓	×	×	×
AR-MIL	RGB+Audio	✓	×	×	×
RTFM	RGB+Text	✓	×	×	×
MSL	RGB+Text	✓	×	×	×
MGFN	RGB+Text	✓	×	×	×
CLIP-TSA	RGB+Text	✓	×	×	×
TPWNG	RGB+Text	✓	×	×	×
DTVDS (Ours)	VAJ	✓	✓	✓	✓

C. Violence Detection in Multimodal

With the advancement of deep learning technology, numerous deep multimodal and cross-modal learning tasks and methods have been developed. For example, Tian et al. [27] and [28] combined images and sound for audio–visual event localization and audio–visual film analysis. Wu et al. [18] employed a multimodal that combines images, sound, and skeletal data for audio–visual action recognition. They also contributed the largest public dataset in the field of violence detection, called XD-Violence [8] which integrated images, videos, and sound for detecting violent actions. Wu et al. [4] considered the relationship between video and audio information to enhance the performance of multimodal fusion. Yang et al. [29] applied contrastive learning to combine images and sound for violence detection. These multimodal models combine various data sources to enhance the detection efficiency of abnormal behaviors. However, the differences between modalities may interfere with each other during model training, potentially affecting model performance due to noise from different modalities. Shi et al. [6] applied using contrastive learning with a MIL loss based on abnormal ratios for anomaly detection. Moreover, compared to unimodal approaches, it is more complex and thus presents more challenges in interpretability. Tian et al.'s study demonstrated that robust temporal feature magnitude learning (RTFM) [28] enhanced weakly supervised video anomaly detection by improving temporal feature magnitude learning, particularly excelling in identifying rare and subtle anomalies. Li et al.'s evidence showed that Self-Training multisequence learning (MSL) [30] refined anomaly scores using MSL and self-training strategies, thereby reducing the likelihood of selection errors. Chen et al.'s research indicated that magnitude-contrastive glance-and-focus network (MGFN) [31] introduced a glance-and-focus network, integrating feature magnification mechanisms and magnitude-contrastive loss to effectively address scene variations. Zhou et al.'s study revealed that Dual Memory Units with Uncertainty Regulation (UR-DMU) [32] employed dual memory units and uncertainty regulation to learn representations of normal and anomalous data, enabling more accurate anomaly differentiation. Joo et al.'s evidence suggested that CLIP-assisted temporal self-attention (CLIP-TSA) [33] leveraged CLIP's ViT features and temporal self-attention mechanisms, outperforming existing methods on benchmark datasets. Yang et al.'s research demonstrated that text prompt with normality guidance (TPWNG) [34] generated precise pseudo-labels using text prompts and normality guidance, improving the performance of weakly supervised video anomaly detection.

Table II summarizes the aforementioned methods and compares them with the proposed model in terms of Input type, Dualmodal, interpretability, privacy protection and Faster computation.

III. ASSUMPTIONS AND PROBLEM FORMULATION

This section introduces the assumptions and problem statement of this study. Given a video V with a duration of t, this article aims to identify whether or not there is violent behavior in V. In recent years, many stuies [19], [20], [21], [22], [28] have conducted research similar to this article. The label of violent behavior is represented in the following format.

Let video $V = \{\Phi_1, \Phi_2, \ldots, \Phi_n\}$ be divided into n equal-length segments, each containing nonviolent, violent behaviors, or their combination. That is, each segment Φ_i might comprise both $\widehat{V}_i$ and V_i, where $\widehat{V}_i$ contains no violent behavior and V_i contains violent behavior, for $0 < i \leq n$. The video V can be represented by $V = \{\widehat{V}_1, V_1, \widehat{V}_2, V_2 \ldots.., \widehat{V}_n, V_n\}$.

Let T_i=(t_i^{Start}, t_i^{End})denote the period of V_i, where t_i^{Start} and t_i^{End} denote the starting and end time points of V_i, respectively. Similarly, let $\widehat{T}_i = (\widehat{t}_i^{\text{Start}}, \widehat{t}_i^{\text{End}})$ denote the time period of $\widehat{V}_i$, where $\widehat{t}_i^{\text{Start}}$ and $\widehat{t}_i^{\text{End}}$ denote the starting and end time points of $\widehat{V}_i$, respectively. For each V_i, this article needs to identify both V_i^b and $\widehat{V}_i^b$ separately.

Consider a violence detection mechanism M which aims to detect the occurrence of violent behavior. Let R^M denote the detected results of V by applying mechanism M. The R^M can be represented as $R^M = \{\widehat{R}_1^M, R_1^M, \widehat{R}_2^M, R_2^M \ldots.., \widehat{R}_n^M, R_m^M\}$.

Let δ_i denote whether or not a given video segment $\widetilde{V}_i$ contains violence behavior. That is, $\delta_i = \left\{\begin{matrix} 1, & \widetilde{V}_i = V_i \\ 0, & \widetilde{V}_i = \widehat{V}_i \end{matrix}\right\}$.

Let $\theta = \{\theta_1, \theta_2, \ldots, \theta_n\}$ be the set of prediction thresholds consisting of n threshold values. Let $\delta_i^M(\theta_j)$ denote whether or not the result of video segment $\widetilde{V}_i$ predicted by M contains violence, with P_i denoting prediction score of $\widetilde{V}_i$. That is, $\delta_i^M(\theta_j) = \{\begin{matrix} 1, \text{ if } P_i \geq \theta_j \\ 0, \text{ if } P_i < \theta_j \end{matrix}.$.

Let TP_i, TN_i, FP_i and FN_i represent True Positive, True Negative, False Positive, and False Negative, respectively, of the prediction result by applying mechanism M to identify the individual video segment $\widetilde{V}$. These can be calculated as follows: TP_i is given by $\mathrm{TP}_i = \delta_i \times \delta_i^M(\theta_j)$, TN_i is calculated as $(1-\delta_i) \times (1-\delta_i^M(\theta_j))$, FP_i is determined by $(1-\delta_i) \times \delta_i^M(\theta_j)$, and FN_i is calculated as $\delta_i \times (1 - \delta_i^M(\theta_j))$.

Let TP, TN, FP, and FN represent the cumulative True Positive, True Negative, False Positive, and False Negative, respectively, of the prediction results by applying mechanism M to all segments $\widetilde{V}_i \in V$, for $1 \leq i \leq n$. The values of these cumulative metrics can be calculated as follows: TP is the sum of all individual true positives, calculated as $\sum_{i=1}^{n} \mathrm{TP}_i$; TN is the sum of all individual true negatives, calculated as $\sum_{i=1}^{n} \mathrm{TN}_i$; FP is the sum of all individual false positives, calculated as $\sum_{i=1}^{n} \mathrm{FP}_i$; and FN is the sum of all individual false negatives, calculated as $\sum_{i=1}^{n} \mathrm{FN}_i$.

Let $\wp^M$ and $\mathcal{R}^M$ denote the Precision and Recall of the predictions by applying mechanism M to identify a given video V. The values of $\wp^M$ can be calculated as the ratio of true positives to the sum of true positives and false positives, given by $\wp^M = (\mathrm{TP}/\mathrm{TP} + \mathrm{FP})$. The value of $\mathcal{R}^M$ can be calculated as the ratio of true positives to the sum of true positives and false negatives, given by $\mathcal{R}^M = (\mathrm{TP}/\mathrm{TP} + \mathrm{FP})$.

A. Average Precision (AP)

Let ${}_{M}$ denote the *AP* of mechanism *M,* which can be calculated by,

$$ {}_{M} = \sum_{j=1}^{n-1} \left(\mathcal{R}\left(\theta_{j+1}\right)^M - \mathcal{R}\left(\theta_j\right)^M \right) \times \wp\left(\theta_j\right)^M \tag{1}$$

where $\mathcal{R}(\theta_j)^M$ and $\wp(\theta_j)^M$ denote the Recall and Precision values at θ_j, respectively.

For a given video V in XD-Violence, the first objective of this article is to develop a violence detection mechanism M aiming to maximize ${}_{M}$. Let $\mathcal{M}$ denote the set of all possible models M and M^{best} denote the best mechanism that achieves the maximal value of ${}_{M}$. Similar to the study [22], [28], The first objective of this article is to develop mechanism M^{best} that satisfies (2).

First Objective in XD-Violence:

$$M^{\text{best}} = \underset{M \in \mathcal{M}}{\arg \text{Max}} \left(\varepsilon_{\tau_M} \right). \tag{2}$$

B. Area Under the Curve (AUC)

Let $\mathcal{A}_M$ denote the *AUC* of mechanism *M*. The value of $\mathcal{A}_M$ can be calculated by,

$$\mathcal{A}_M = \int_0^1 \mathrm{TP}\left(\theta_j\right) d\left(FP\left(\theta_j\right)\right) \tag{3}$$

where $FP(\theta_j)$ and $\mathrm{TP}(\theta_j)$ denote the FP and *TP* values at threshold θ_j. For a given video V in UCF-Crime, the second objective of this article is to develop a violence detection mechanism M aiming to maximize $\mathcal{A}_M$. Equation (4) reflects the objective of UCF-Crime. Let M^{best} denote the best mechanism that achieves the maximal value of $\mathcal{A}_M$. Similar to the study [22], [28], The second objective of this article is to develop mechanism M^{best} that satisfies (4).

Objective in UCF-Crime:

$$M^{\text{best}} = \underset{M \in \mathcal{M}}{\arg \text{Max}} \left(\mathcal{A}_M \right). \tag{4}$$

This section introduced the assumptions and problem formulation of this article. The next section will introduce the method proposed in this article.

IV. Proposed VAJ and DTVDS Mechanisms

This section introduces the details of the proposed VAJ and DTVDS mechanisms. The VAJ is a data preprocessing technique that allows the fusion of image and audio data into a single image. This process converts Dual-modal data formats into an unimodal format for processing. Following this format conversion, the DTVDS model processes the input message treated by VAJ, learning to extract violent features in the image, classify whether it is violent, and visualize the violent features, thereby achieving the interpretability of DTVDS.

4.1 VAJ Processing Phase

The VAJ is a data preprocessing technology that aims to combine the image and voice signal into a single image. In related multimodal studies [18], [22], audio and video features are extracted using different models and then combined at the backend to form a multimodal network. However, this approach has the drawbacks of inconsistency between models and high difficulty in feature fusion. Unlike previous research, this article proposes the innovative idea of VAJ. The VAJ guarantees that the fusion of source data from images and voice signals, which originally served as input data for dual-modal models, can be integrated into a single image while retaining all features. The VAJ preprocessing phase also aims to synchronize the image and voice signal using MFCC to avoid the imbalance issue that exists in multimodal networks. The VAJ consists of two tasks: 1) the transformation of the image data in the original video frame and 2) the integration of the audio information into the transformed image data.

A. Image Transformation Task of VAJ

Fig. 3 gives an example to illustrate the concepts of the first task of the VAJ method. The main function of this task is to extract the key features from the video frame and further transfer these features to an image. This task includes four steps: First, the original color image I is converted into a grayscale image I_{gray}. The purpose of this step is to reduce computational complexity because a grayscale image contains only brightness information and no color information, which is very useful for subsequent feature extraction. In the second step, a shift operation is first applied to each

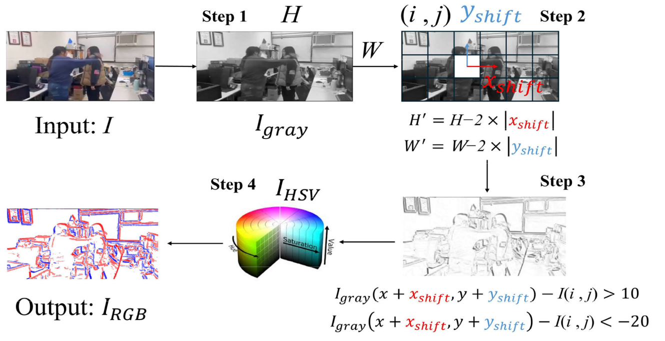

Step 1
H
W
(i, j) y_{shift}
Step 2
x_{shift}
Input: I
I_{gray}
$H' = H-2\times|x_{shift}|$
$W' = W-2\times|y_{shift}|$
Step 4
I_{HSV}
Step 3
Saturation
Value
Output: I_{RGB}
$I_{gray}(x + x_{shift}, y + y_{shift}) - I(i, j) > 10$
$I_{gray}(x + x_{shift}, y + y_{shift}) - I(i, j) < -20$

Fig. 3. First task of VAJ: convert the image data in the original video frame.

pixel in the image to create a shifted image. The aim of shifting is to facilitate subsequent edge detection calculations by highlighting changes through the comparison of pixel values before and after the shift. In the same step, the VAJ further calculates the difference in pixel values after the shift compared to the original pixel values to obtain the pixel differences. This step is crucial for feature extraction, as it highlights edges that change significantly between frames.

To visualize these edge features in the grayscale image, in the third step, the RGB color space is utilized. The Red color represents the edges where intensity is calculated to decrease, blue represents edges where intensity is increasing, and white represents edges with less noticeable intensity changes. This visualization helps analyze which edges are visually prominent and which are secondary. In the fourth step, the image colored in RGB is then transformed into the HSV space, to adjust the display brightness based on the magnitude of pixel differences. In the HSV space, it is more intuitive to adjust the visualization of edges based on brightness values, making the features stand out more. Finally, the image in the HSV space is mapped back to the RGB space to obtain I_{RGB}.This completes the VAJ task, including the feature extraction and conversion process from video frames to images.

The following presents the details of the first task of the VAJ.

Step 1 (Grayscale): Let I denote one video frame. This stage converts I into a grayscale image I_{gray}, with dimensions defined by height H and width W.

Step 2 (Grayscale Image Translation Calculation): Let $f_{\text{shift}}(.)$ denote the shift function. This function is applied to each pixel in an image to detect object edges and retain these important features. The function $f_{\text{shift}}(.)$ can be implemented by using two parameters, x_{shift} and y_{shift}, representing the horizontal and vertical translation amounts, respectively. These parameters are set by predefined values. After applying the shift function, the final shifted image frame can be obtained by the operation $I_{\text{gray}}(i + x_{\text{shift}},\ j + y_{\text{shift}}) = f_{\text{shift}}(I_{\text{gray}}(i,\ j))$.

The height and width of the final shifted image can be calculated by subtracting twice the absolute value of the shifts from the original dimensions, given as $H' = H - 2 \times |x_{\text{shift}}|$ and $W' = W - 2 \times |y_{\text{shift}}|$.

Next, the changes of the image frame I_{gray} are calculated to extract the object edges. Let φ denote the differences between the original and the shifted images. The value φ calculated as $\varphi = I_{\text{gray}}\ (i + y_{\text{shift}}, j + x_{\text{shift}})$ - $I_{\text{gray}}\ (i, j)$.

Step 3 (Transformation From Gray Image to RGB Image): This step aims to develop color mapping rules, evaluate the differences between before and after mapping, and intuitively mark these changes on the image. By calculating the differences between the displaced and the original images, it is possible to detect the changes in the edges of a person's movements. In the RGB color space, the blue (B) channel is expected to represent the enhancement of edges, which can record more apparent motion characteristics, while the red (R) channel is expected to indicate the weakening of edges, meaning that the original motion characteristics become less obvious. The following presents how to carry out these expectations.

Based on the value of φ, a heuristic color mapping rule is presented, aiming to transform the gray color image to the RGB image. The transformation rule is presented as shown in (5).

The positive value of φ indicates the object edges is formed. Since the Blue channel is designed for represent the enhancement of edges, the condition for storing positive φ to Blue channel. On the contrary, the Red channel stores the negative value, representing the weakening of edges. The threshold values of color mapping are 10 and -20, respectively.

$$\text{Color Mapping}\begin{cases} \text{if } \varphi > 10\,\text{Blue} \\ \text{if } \varphi < -20, \text{Red} \\ \text{if } -20 \leq \varphi \leq 10, \text{White} \end{cases} \tag{5}$$

Step 4 (Enhancement of Transformation From RGB Image to HSV Space): The fourth step of VAJ's first task aims to enhance the expressiveness of color mapping by converting RGB images into HSV space. By adjusting saturation and brightness based on the differences obtained in the second step, it intensifies the contrast before and after mapping, and visually displays these changes in intensity within the image. After the HSV conversion, the image is converted back to RGB format to facilitate neural network training.

Following the completion of heuristic color mapping to reflect the intensity of changes, the saturation and brightness are further adjusted. The image is converted to the HSV color space, keeping the hue (H) constant. The adjustment of saturation (S) and brightness (V) is based on the magnitude of the pixel difference φ, which is calculated using the following equations: the new saturation S_{new} is given by $S_0{+}\alpha_s{\cdot}(\tanh(\beta_s \cdot \varphi) - S_0)$, and the new brintess V_{new} is given by $V_0 + \alpha_v \cdot (\tanh(\beta_v \cdot \varphi) - V_0)$.

Herein, α_s and β_s are the parameters that control the magnitude and rate of adjustment for saturation, respectively. Similarly, the parameters α_v and β_v control the magnitude and rate of adjustment for brightness, respectively. The hyperbolic tangent function (tanh) ensures that the adjustments are contained within a certain saturation and brightness range, avoiding excessive adjustments. This parametric form allows for precise control over the changes in saturation and brightness, based on the pixel difference φ.

After the adjustment, the transformed I_{HSV} is converted back into the RGB space I_{RGB} through the mapping function,

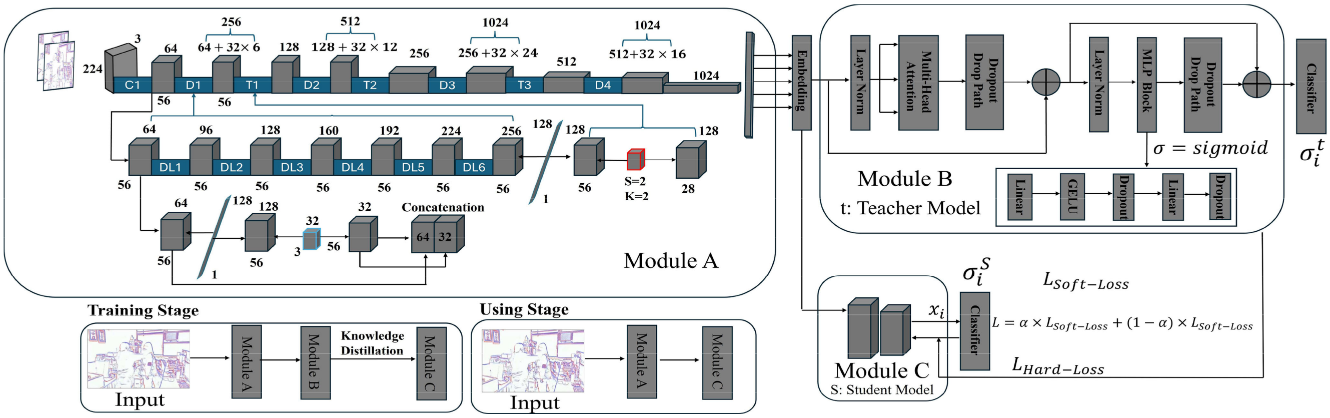


Fig. 4. Second stage of VAJ: convert the Audio data in the original video.

where the red (R) component is calculated as $R = V_{\text{new}} \times (1 - S_{\text{new}} \times (1 - \cos(H)))$ and the blue (B) component as $B = V_{\text{new}} \times (1 - S_{\text{new}} \times (1 + \cos(H)))$. Herein,$H$, S_{new} and V_{new} represent the hue, adjusted saturation, and brightness, respectively. This mapping function illustrates the specific method for converting the adjusted HSV image back to the RGB color space, where the calculations for the red (R) and blue (B) components consider the influence of the hue H along with the adjusted values of saturation and brightness. This process effectively transforms the I_{HSV} to I_{RGB}, facilitating further processing and display of the image in the RGB color system.

Till now, this section presents the entire process of the first stage of the VAJ method. The second stage of the VAJ will be introduced next.

B. Voice Transformation Task of VAJ

Fig. 4 showcases the second task of the VAJ method, which involves extracting audio from the original video and integrating it with the I_{RGB} obtained from the first task to produce the updated I'_{RGB}’ image. As shown in Fig. 4, this task is divided into two steps. First, for audio–visual synchronization, the audio is converted into MFCCs and segmented, then synchronized with I_{RGB} to ensure that the MFCCs are averaged across each video segment. Following this, to incorporate the audio features into I_{RGB}, the image is converted to the HSV space. Based on this, the original image's white hues are adjusted to green, and the brightness and contrast are modified according to the average MFCC values, before finally converting the processed image back to the RGB space. The specific process is as follows.

Step 1 (Synchronize Voice and Video): Initially, the audio A from the original video is segmented to obtain $A = \{A_1, A_2, \ldots, A_n\}$, where n denotes the number of segments. Then, each audio segment A_i, where $1 \leq i \leq n$, is processed with MFCC to yield $\text{AMF} = \{\text{AMF}_1, \text{AMF}_2, \ldots, \text{AMF}_n\}$. Then the average of each segmented AMF_i is calculated over T frames.

The purpose of averaging is to synchronize the audio data with the video frames I_{RGB}, ensuring that the temporal features of the audio align with the visual data. This synchronization helps maintain temporal consistency between the audio and video in subsequent processing. The formula for calculating the average is given by $\overline{\text{AMF}}_j = (1/T) \sum_{i=1}^{T} \text{AMF}_{(i,j)}$. Where T represents the total number of frames within the segmented time, and $\overline{\text{AMF}_j}$ is the average value of the jth MFCC coefficient over T frames.

Step 2 (Synchronize Voice and Video): Subsequently, the synchronized I_{RGB} is processed through the HSV color space, specifically adjusting the original white parts to green, based on the corresponding $\overline{\text{AMF}_j}$ by adjusting the V value. The calculation of the new V value is as given by: $V'_{\text{new}} = (([\overline{\text{AMF}_j}]/\max\ (AMF))) \times 255$, where max (AMF) is the maximum value amongst all elements in the AMF set. The calculated V'_{new} value will be used for the brightness component in the HSV color space. During this process, the hue (H) and saturation (S) are usually unchanged. The purpose is to allow audio information to influence the color representation in the video, so that the brightness changes in the green areas can reflect the features of the audio, thus enhancing the synergy of audio–visual effects. For the replacement of green pixels, if a pixel is detected as green, then it is replaced with the new HSV value. The purpose of this replacement is to map the variations in the audio signal to changes in the brightness of the image, visualizing the intensity changes of the audio signal. Finally, the adjusted HSV color space I'_{HSV} is converted back to the RGB color space I'_{RGB}. he final converted information in the green (G) channel is calculated as follows $G = V'_{\text{new}} \times (1 - S \times (1 - \cos(H - 120^{\circ})))$.

This concludes the detailed operations designed in the VAJ method. The VAJ method merges image and voice information into a single image through two stages and integrates the features of both data sources. Additionally, the transformed fusion image remains in the RGB image format, making it suitable for further processing and analysis. In the next section, the details of DTVDS will be presented.

C. DTVDS

This section will detail the proposed DTVDS system. Fig. 5 shows the specific architecture of DTVDS, which is composed of three submodels: Module *A*, Module *B*, and Module *C*. The system includes two phases: 1) training and 2) usage. In the training phase, RGB images processed by VAJ are

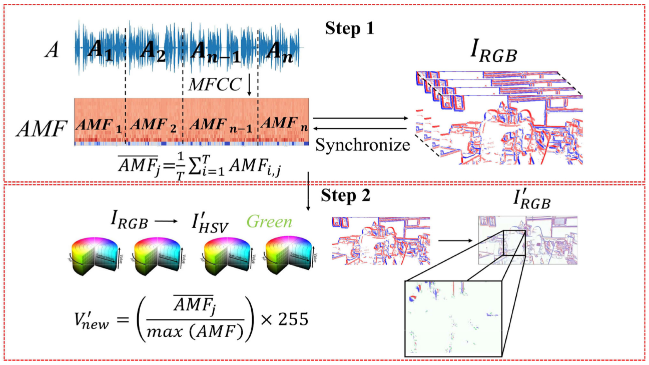

Step 1
A
A_1
A_2
A_{n-1}
A_n
MFCC
AMF
AMF_1
AMF_2
AMF_{n-1}
AMF_n
Synchronize
I_{RGB}
$\overline{AMF}_j = \frac{1}{T}\sum_{i=1}^{T} AMF_{i,j}$
Step 2
$I_{RGB} \longrightarrow I'_{HSV}$
Green
I'_{RGB}
$V'_{new} = \left(\frac{\overline{AMF_j}}{max\ (AMF)}\right) \times 255$

Fig. 5. Proposed DTVDS system.

first input into Module A for feature extraction, and then Module B is used to explore and learn from time series data. This is also known as the offline phase. After the training of Module B is completed, the knowledge distillation technique allows the teacher Module B, to guide the training of the student model, Module C. In the usage phase, after Module A processes the image with VAJ for feature extraction, Module C quickly identifies violent scenes with its two convolutional layers, also known as the online phase. Below is the specific process.

The input image $I \in \mathbb{R}^{224\times224\times3}$, has a size of 224 × 224 with three channels for RGB. First, it is input into Module A, where it passes through the convolutional layer $C1$, which uses 64 filters, each with a size of 7 × 7, and a stride of 2. The convolution operation can be represented as $\text{Conv}(I,\ K,\ S)$, where I is the input image, $K = 64$ is the number of filters, and $S = 2$ is the stride. This operation is depicted in $F_{C1} = \text{Conv}(I, 64, 2)$.

Subsequently, the image passes through four dense layers D and three transition layers T, each dense layer D consisting of several dense blocks DL_i. Each dense block DL_i includes L_i layers and each layer performs the operation sequence of Batch Normalization (BN), ReLU and then Convolution operations. Let H denote the output of the current layer, and H_0 denote the output of the previous layer, the output of each layer in the dense layer can be represented $H = \text{Conv}(\text{BN}(\text{ReLU}(H_0)),\ K)$, where K is the growth rate, BN represents batch normalization.

In each transition layer, a convolution operation followed by a 2×2 average pooling operation is performed. Let F be the output of the dense layer. The output of the transition layer F' can be represented by $F' = Pool(\text{Conv}(F,\ k_{\text{trans}}, 1)),\ 2)$, Where k_{trans} is the number of convolution filters in the transition layer.

Let E denote the output obtained by applying global average pooling after the last dense layer, then the representation of E can be expressed as $E = \text{GlobalAvgPool}(H)$.The global average pooling compresses the spatial dimension of each feature map to 1, thus E is a 1-D vector. Its length is determined by the number of channels in the last dense block, 1024.

The above describes the process by which images are represented as embeddings through Module A, followed by these embeddings being learned through Module B, with the specific process as follows.

First, the embeddings E obtained from Module A are used as the input for Module B. Let E_{norm} denote the Layer Normalization is applied to E, resulting in $E_{\text{norm}} = \text{LayerNorm}(E)$.

Then, E_{norm} undergoes three different linear projections to obtain the Query (Q), Key (K), and Value (V) matrices, expressed as $Q = E_{\text{norm}}W^q$, $K = E_{\text{norm}}W^k$, and $V = E_{\text{norm}}W^v$. For each attention head, the scaled dot-product attention is used to calculate the attention scores, which is given by $Attention(Q, K, V) = \text{softmax}((QK^T/\sqrt{d_k}))V$,where d_k is the dimensionality of the key vectors and is used to scale the dot product to prevent gradient vanishing problems.

In the multihead attention mechanism, this operation is executed in parallel multiple times, with each head using different W^q, W^k, W^v. The output for each head i is given by $\text{Head}_i = \text{Attention}(QW^q, KW^k, VW^v)$. The outputs of all attention heads are then concatenated and passed through linear projection to obtain the final multihead attention output, denoted as $H = \text{concat}(\text{Head}_1, \text{Head}_2, \ldots, \text{Head}_h)W^0$, where W^0 is another learnable parameter matrix, H is the number of attention heads, and the concatenation operation is represented by concat. Let MHA represent the multihead attention mechanism, which can be expressed as $MHA(E_{\text{norm}}) = \text{concat}(\text{Attention}(E_{\text{norm}}(W_q^1, \ldots.))$. After applying the multihead attention mechanism, the Dropout operation is performed to reduce overfitting, resulting in $H' = Dropout(H)$. The original input E_{norm} is added to H' through a residual connection, followed by layer normalization, leading to $H_{\text{norm}} = \text{LaynerNorm}(H' + E_{\text{norm}})$. Next, it passes through an FFN, which consists of two linear layers and an activation function GELU. Let F denote the output through the FFN, which is expressed as $F = \text{Linear}(\text{GELU}(\text{Linear}(H_{\text{norm}})))$.

The output F is then passed through Dropout and added back to H_{norm} as a residual connection, resulting in $F_{\text{norm}} = \text{LayerNorm}(F')$. Finally, a linear layer and sigmoid function are used to achieve binary classification output probabilities. Let σ_t denote the output. Which is calculated as $\sigma_t = \text{Sigmoid}(\text{Linear}(F_{\text{norm}}))$.

In Module B, the loss function $\mathcal{L}_1$ used for training is the cross-entropy loss. Given the true labels y and model outputs σ_t, the cross-entropy loss can be expressed by

$$\mathcal{L}_1 = -\frac{1}{N}\sum_{i=1}^{N}\left[y_i \log\left(\sigma_t{}^{(i)} + (1-y_i)\log\left(1-\sigma_t{}^{(i)}\right)\right)\right] \quad (6)$$

Herein, N is the number of samples, y_i is the true label of the ith sample, and $\sigma_t{}^{(i)}$ represents the probability predicted by the model for the ith sample. By minimizing this loss function, Model B can be optimized within the framework of supervised learning, improving the performance of classification.

In addition, this article designs fine-grained detection of violent behavior for Module B. The loss function is set to $\mathcal{L}_2$, as shown in

$$\mathcal{L}_2 = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{M} y_{\text{ij}} \log\left(\frac{\exp\left(z_j^{(i)}\right)}{\sum_{k=1}^{M}\exp\left(z_k^{(i)}\right)}\right), \quad (7)$$

Herein, $z_j^{(i)}$ represents the logits for the jth class of the ith example, y_{ij} is the target probability for the jth class of the ith example, N is the number of examples, and M is the number of classes.

After Module B is trained, it is used as the teacher model, and knowledge distillation is performed on the Module C student model. In the context of knowledge distillation, the student model is usually trained using two loss functions: hard target loss and soft target loss. The following is a detailed step description.

For coarse-grained and fine-grained, the expressions of the hard target loss $\mathcal{L}_{\text{hard}}$ are shown in (6) and (7), respectively. The soft target loss is the cross-entropy loss between the probability distributions of the student model output and the teacher model output. The temperature parameter T is used to help the student model learn the behavior of the teacher model. The soft target loss $\mathcal{L}_{\text{soft}}$ can be expressed by

$$\mathcal{L}_{\text{soft}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{C}\left[\operatorname{softmax}\left(\frac{\sigma_j^t}{T}\right)\log\left(\frac{\operatorname{softmax}\left(\frac{\sigma_j^s}{T}\right)}{\operatorname{softmax}\left(\frac{\sigma_j^t}{T}\right)}\right)\right] \tag{8}$$

where C is the total number of categories, σ_j^t and σ_j^s are the logits of the jth class predicted by the teacher and the student models, respectively. The total loss L is the weighted sum of the soft target loss and the hard target loss, and the student model is updated through this total loss. The expression for total loss is shown in

$$L = \alpha\mathcal{L}_{\text{soft}} + (1-\alpha)\mathcal{L}_{\text{hard}} \tag{9}$$

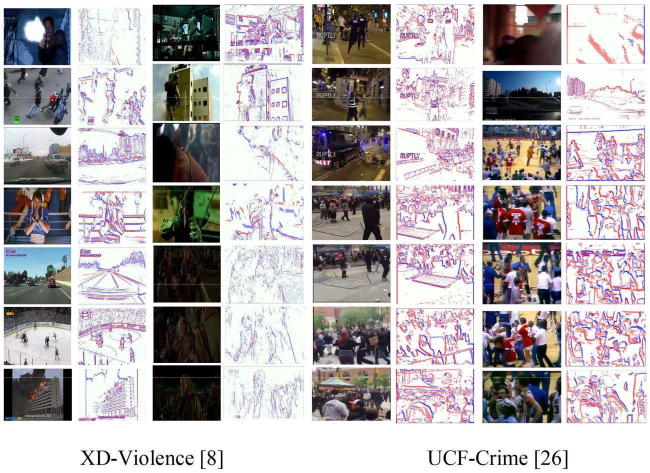
where α is a hyperparameter used to balance the contributions of the two losses. The above is the entire process of the DTVDS training period. The following describes the interpretable parts of the using stage of DTVDS.

Let the last convolutional layer in Module C consist of K feature maps, and the size of each feature map is $M \times N$. That is, we have $\mathrm{F}\epsilon\mathbb{R}^{M\times N\times K}$. Connecting to the GAP layer and outputting a $1 \times 1 \times K$ feature vector reduces the number of parameters in the model. After the GAP layer, a linear classification layer is connected. Since there are C output categories, this layer consists of a weight matrix $\mathrm{W}\epsilon\mathbb{R}^{K\times C}$. For each category c, calculate the weighted sum to obtain categorical logits, as follows $Z_c = \sum_{k=1}^{K} W_{kc}.\mathrm{GAP}(F_k)$. Where Z_c represent the logit for category c. Finally, visualization is generated. For category c, the visualization M_c is calculated as $M_c(x, y) = \sum_{k=1}^{K} W_{kc}.F_k(x, y)$. Herein, $M_c(x, y)$ denotes the activation strength of class c at feature map position (x, y).

V. Model Performance

In this section, relevant experimental performance and analysis are presented.

A. Dataset



XD-Violence [8]

UCF-Crime [26]

Fig. 6. Some examples from XD-Violence [8] and UCF-Crime [26].

This article employs two datasets: XD-Violence [8] and UCF-Crime [26]. The XD-Violence is the largest publicly available dataset in the violence detection domain currently. It comprises 4754 videos, totaling 217 h, and includes six types of violent events: verbal abuse, car accidents, explosions, fights, riots, and shootings. This dataset is randomly divided into a training set with 3954 videos and a test set with 800 videos. The test set is further categorized into 500 violent videos and 300 nonviolent videos. The UCF-Crime dataset consists of 1900 real-world surveillance videos, with 1610 for training and 290 for testing.

It is noteworthy that the training videos in both the XD-Violence and UCF-Crime datasets are encoded in MPEG format and possess only video-level labels. In practice, the proposed VAJ method is employed to process these MPEG-encoded videos. Specifically, the VAJ method first decodes the MPEG video stream to extract individual frames and synchronized audio signals. Subsequently, it integrates the visual and auditory information into a single image representation, a process that is critical to the violence detection system. Fig. 6 presents some examples from these two datasets, illustrating the transformation effects achieved by the proposed VAJ method.

B. Experimental Setup

This article describes the implementation of the proposed VAJ and DTVDS system.

In the VAJ processing framework, the parameter settings for the image preprocessing stage in (5) are as follows: when converting a grayscale image to an RGB mapping, the positive threshold is set to 10, and the negative threshold is set to -20. For image translation operations, both horizontal and vertical shifts are set to 1 pixel. In the HSV color space, saturation and brightness adjustments are performed using the tanh function, with the adjustment parameters α, β, γ, and δ all set to 1.

In DTVDS, In Module A, the first convolutional layer employs 64 filters of size 7×7 with a stride of 2. After global average pooling, a 1024-dimensional feature vector is obtained. During training, the Adam optimizer is used with an initial learning rate of 0.001, a batch size of 128, and a dropout rate of 0.6. Regarding the loss function, the balance hyperparameter for coarse-grained detection is set to $\lambda = 0.7$, while other weights in the fine-grained detection loss remain fixed at 1.

In the knowledge distillation module, the temperature parameter T in (8) is set to 2.5, while the hyperparameter α in (9), which balances soft and hard target losses, is set to 0.2. Additionally, in the multihead self-attention mechanism, this study employs 8 attention heads, with the Key dimension for each head set to 64.

C. Experimental Result

Fig. 7 compares the proposed DTVDS system's performance with other models on the XD-Violence and UCF-Crime datasets, using VAJ and RGB inputs. The blue line shows XD-Violence results (AP% metric), while the orange line shows UCF-Crime results (AUC% metric).

The blue section highlights that VAJ integrates multimodal data into one image, enabling the model to learn more and outperform RGB-only input. DTVDS excels due to its effective feature extraction and spatiotemporal processing. Also, DTVDS (Online), via knowledge distillation from the Offline version, updates in real-time and surpasses the Offline model.

The orange section shows one-stage processing of VAJ and RGB inputs. Lacking audio in UCF-Crime, VAJ is limited to one-stage use. VAJ outperforms RGB by capturing subtle motion changes through image shift differences, reducing color and environmental noise compared to RGB. DTVDS outperforms others, thanks to Module B's time-series analysis in the Offline version enhancing behavior recognition, and Module C's distillation in the Online version boosting detection speed and performance.

Table III compares the proposed DTVDS with other multimodal models on the XD-Violence dataset. The horizontal axis lists models, and the vertical axis shows AP value sizes. It compares VAJ input, proposed here, with RGB and Audio inputs. VAJ enhances preprocessing, addressing data imbalance and modal noise issues in traditional multimodal models. Whether using RGB+Audio or VAJ alone, DTVDS Online and Offline outperform others, thanks to a design incorporating temporal sequencing and knowledge distillation, boosting behavior recognition capabilities.

Fig. 8 compares the proposed VAJ method with RGB + Video multimodal inputs for violence detection on the XD-Violence dataset. In Fig. 8, the x-axis lists violence types (0–5: Fighting, Shooting, Riot, Abuse, Car Accident, Explosion), the y-axis shows AP values, and the z-axis represents model types. The VAJ method outperforms others, especially in detecting Category 4 (Car Accident) and Category 3 (Abuse), due to its ability to handle multimodal data imbalance and inconsistency, improving accuracy where traditional models struggle. The DTVDS model also excels, leveraging Module A's feature extraction and Module B's time-series learning to track violent behavior accurately. Through knowledge distillation, Module C learns from Module B, enhancing speed and accuracy over HD-Net, C3D+Self Attention, MCL, and AR-MIL models.

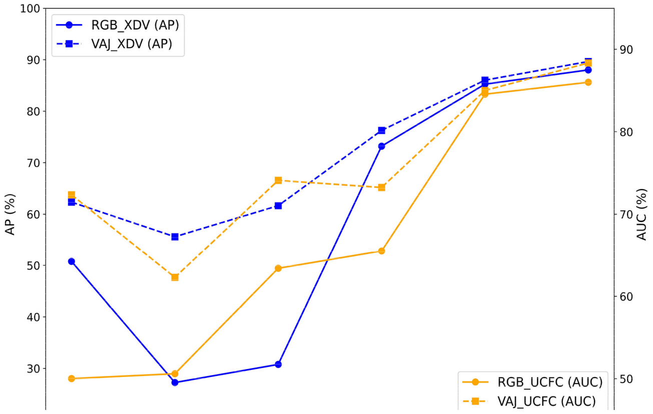


Fig. 7. Performance of the DTVDS system is compared with other models on the XD-Violence dataset (blue line) and UCF-Crime (orange line) using two different input methods: VAJ and RGB images.

TABLE III
COMPARISON OF DTVDS AND MULTIMODAL MODELS USING VAJ PROCESSING IN TERMS OF XD-VIOLENC

Method	RGB and Audio	VAJ
HD-Net	78.64%	79.29%
C3D+Self Attention	81.69%	82.71%
MCL	84.03%	84.08%
AR-MIL	85.79%	85.58%
DTVDS (Offline)	85.24%	86.02%
DTVDS (Online)	88.01%	89.64%

Table IV presents ablation experiments on DTVDS, comprising three submodules: Module A (feature extraction), Module B (time series analysis), and Module C (knowledge distillation and rapid detection). Module C, focused on speed, is excluded from ablation. The experiment assesses the synergy of Module A (CNN, ResNet, DenseNet) and Module B (LSTM, bi-directional LSTM, Transformer). Table IV shows both modules are essential; using only one reduces accuracy. While bi-directional LSTM alone yields the highest AP, combining it with Transformer outperforms it, thanks to Transformer's self-attention integrating temporal data effectively. Paired with Module A, Transformer boosts time-series comprehension, enhancing accuracy. Ablation reveals DenseNet in Module A excels, leveraging dense layer connections for efficient feature and gradient flow, improving recognition and AP. Results confirm feature extraction and time-series analysis are vital for violence detection, with DenseNet + Transformer yielding the best outcomes.

Fig. 9 presents a comparison of the explainability performance of DTVDS using two different inputs. From left to right, it shows the original image, areas perceived as violent by humans, explainability results with RGB+Audio input, and the VAJ-converted image along with its explainability results. As illustrated in Fig. 9, when using RGB+Audio as input, DTVDS fails to identify violent regions and lacks areas specifically designated as violent behavior. In contrast, when using images and audio processed through VAJ, the model demonstrates excellent explainability, with these areas clearly marked and incorporating both speech and action

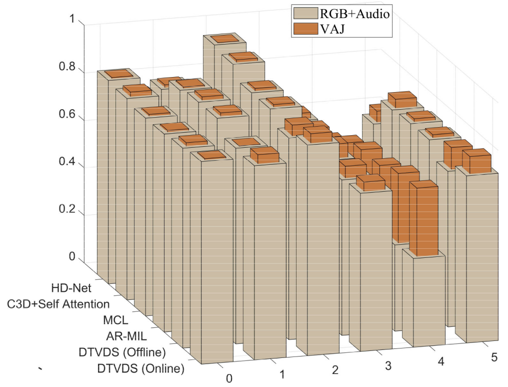


Fig. 8. Compares the performance of the proposed VAJ method with RGB + Video multimodal input in terms of violence detection.

TABLE IV
Ablation Study for DTVDS

Module A			Module B			AP (%)
$\mathfrak{A}_1$	$\mathfrak{A}_2$	$\mathfrak{A}_3$	$\mathfrak{B}_1$	$\mathfrak{B}_2$	$\mathfrak{B}_3$	
✓						48.24
	✓					51.26
		✓				53.15
			✓			56.35
				✓		62.85
					✓	53.24
✓			✓			75.27
✓				✓		76.3
✓					✓	78.88
	✓		✓			80.17
	✓			✓		85.63
	✓				✓	84.62
		✓	✓			76.48
		✓		✓		81.2
		✓			✓	**86.02**
$\mathfrak{A}_1$: CNN $\mathfrak{A}_2$: ResNet $\mathfrak{A}_3$: DenseNet			$\mathfrak{B}_1$: LSTM $\mathfrak{B}_2$: Bidirectional lstm $\mathfrak{B}_3$: Transformer			Average Precision

information. This indicates that VAJ not only transforms multimodal data into more practical unimodal data but also significantly enhances the model's explainability.

Fig. 10 compares the DTVDS model's performance in two real-time input scenarios. The upper part shows a video processed via VAJ, integrating video and audio, while the lower part uses RGB images and audio. The *x*-axis represents time, and the *y*-axis indicates predicted violence levels. The figure sequentially displays scenes of a skating rink brawl, a sinking ship, and two men fighting.

As shown in Fig. 10, using traditional RGB and audio inputs, the DTVDS model misclassifies a normal match as violent and fails to detect actual violence when a player is knocked down. In the sinking ship scene, the model does not recognize violent events, such as explosions, until the fire subsides. In the fighting scenario, it incorrectly labels the scene as nonviolent after one person is knocked down but later misidentifies a helping gesture as violence.

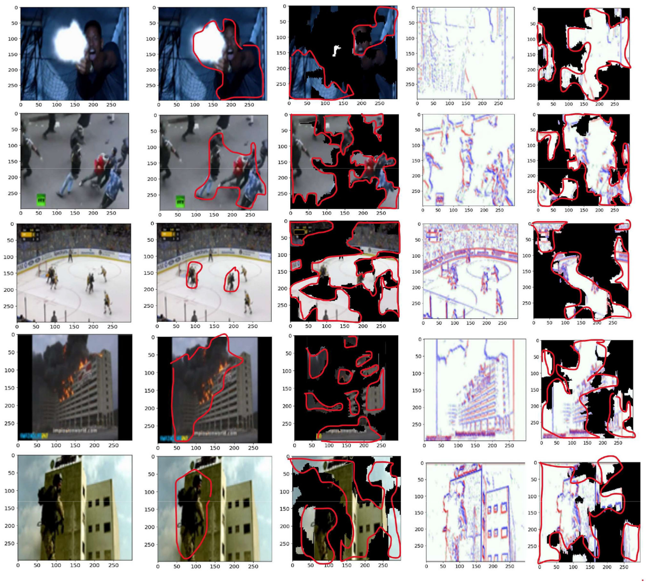


Fig. 9. Interpretable analysis in RGB and VAJ.

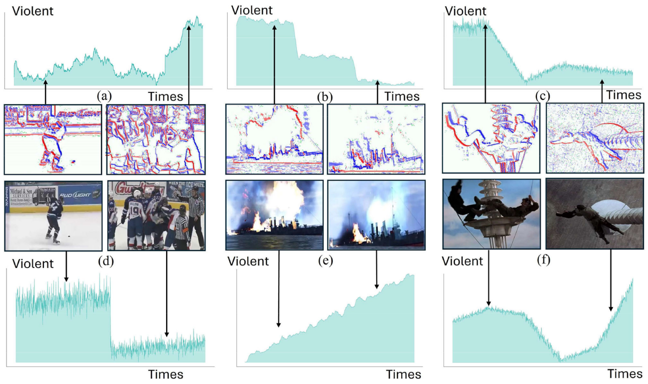


Fig. 10. Real-time detection performance comparison of DTVDS with two different inputs is as follows: Subfigures (a), (b), and (c) display the performance when VAJ is used as the input data, showing that all violent scenes are detected. Subfigures (d), (e), and (f) show the performance using RGB images and audio as inputs. In subfigure (d), a scene from a baseball game, two people standing close to each other are mistakenly identified as engaging in violent behavior. In subfigure (e), a scene of a ship being hit by artillery, the explosion is misinterpreted as violence. In subfigure (f), during a fight scene, a person falling is mistakenly judged as violence.

These errors stem from imbalanced and ambiguous dual-modal data—chaotic audio affects the first and third cases, while the absence of audio in the second forces the model to rely solely on visuals, missing fire intensity. VAJ preprocessing resolves these issues by synchronizing video and audio inputs, improving detection accuracy. Even without audio, the model identifies violence based on image edge changes, significantly enhancing its performance.

Fig. 11 aims to compare the computational complexity of the proposed DTVDS with other methods. The *x*-axis represents the sequence length, and the *y*-axis represents the complexity. As shown in Fig. 11, the proposed DTVDS online model does not increase in computational complexity as the sequence length increases, unlike other models. Moreover, the DTVDS offline model exhibits lower complexity compared to

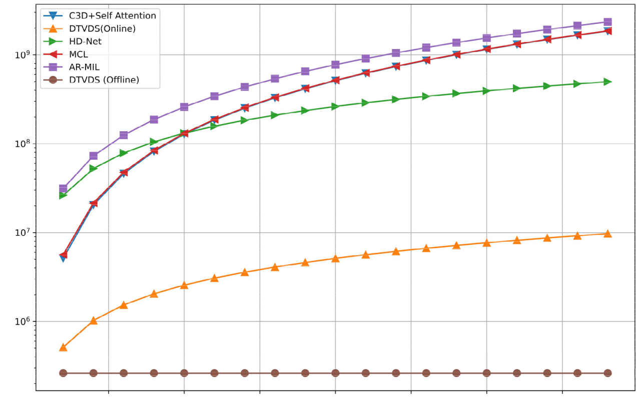


Fig. 11. Complexity statistics.

TABLE V
CORSE GRAINED COMPARISON ON XD-VIOLENC

Input	Methods	XD (AP)
Text+Video	RTFM [28]	77.81%
	MSL [30]	78.58%
	MGFN [31]	80.11%
	UR-DMU [32]	81.66%
	CLIP-TSA [33]	82.17%
	TPWNG [34]	83.68%
Audio+Video	DTVDS (Offline)	86.02%
	DTVDS (Online)	88.01%
VAJ	DTVDS (Offline)	**87.88%**
	DTVDS (Online)	**89.64%**

other models, demonstrating the feasibility of deploying the proposed DTVDS in real-world edge scenarios.

Table V presents a performance comparison of different multimodal methods on the XD-Violence dataset, with results clearly indicating that the Audio+Video fusion approach outperforms the Text+Video combination. Specifically, the DTVDS series methods show outstanding performance, particularly its online version which achieves the highest AP value of 89.64% in the VAJ configuration, nearly 6% points higher than the best Text+Video method, TPWNG (83.68%). Additionally, the online version of DTVDS consistently outperforms its offline version across various configurations, which may be attributed to online processing's superior ability to capture temporal features. These findings strongly confirm that in violence detection tasks, the integration of audio features (such as screams, explosions, etc.) with video information provides more direct and relevant identification cues, thus the combination of image and audio indeed holds a significant advantage over the integration of image and text.

Meanwhile, the Appendix chapter also discusses the computational complexity of the designed methods, their deployment on AIoT, and a simple demonstration.

VI. Limitation and Feature Work

Although the proposed VAJ and DTVDS methods have demonstrated strong performance, they still have certain limitations. The reliance on RGB color information may reduce robustness under varying lighting conditions or complex environments. The imbalance in multimodal data could affect the weight distribution between audio and video fusion, thereby impacting model performance. Moreover, the current interpretability analysis methods remain limited and fail to fully reveal the model's decision-making process. Future research should explore more robust data fusion strategies to enhance modal consistency, introduce advanced interpretability techniques to improve decision transparency, expand the application of this method to areas, such as traffic safety monitoring and medical behavior analysis.

VII. Conclusion

This article discovers that RGB images have a negative impact on the detection of violent behaviors, a finding that is supported both theoretically and empirically. This article presents an innovative data preprocessing method named VAJ, along with its accompanying interpretability system, DTVDS. The proposed VAJ first uses edge detection techniques to analyze image information in videos, retaining critical information. The diminishing and enhancing of edges are represented through red and blue color intensities, respectively, which also exploit the key features of violent behavior. Additionally, in processing audio from videos, VAJ uses MFCC preprocessing to obtain voice data synchronized with the video on a per-second basis and employs an averaging method to ensure that each video frame is synchronized with the corresponding audio. The frequency variations in the MFCC of the voice are indicated with green color intensity in VAJ. The VAJ integration method simplifies multimodal data into an unimodal format while preserving key information from the original images. It not only reduces the complexity of multimodal training and conflicts arising from different modal data types but also significantly enhances the model's interpretability. Moreover, the proposed DTVDS system combines the feature extraction capabilities of DenseNet with the temporal analysis prowess of Transformers, simplifying the model through distillation learning to facilitate real-time computation and further enhance interpretability. Experimental results show that these methods not only effectively address the interpretability issues of Dual-modal models but also significantly improve upon previous research.

References

[1] Q. Wu, P. Xiong, Z.-R. Tang, G.-J. Li, A. Song, and L.-M. Zhu, "Detecting dynamic behavior of brain fatigue through 3-D-CNN-LSTM," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 52, no. 1, pp. 90–100, Jan. 2022.

[2] B. Nikpour and N. Armanfard, "Spatial hard attention modeling via deep reinforcement learning for skeleton-based human activity recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 7, pp. 4291–4301, Jul. 2023.

[3] M. Ding, Y. Ding, L. Wei, Y. Xu, and Y. Cao, "Individual surveillance around parked aircraft at nighttime: Thermal infrared vision-based human action recognition," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 2, pp. 1084–1094, Feb. 2023.

[4] P. Wu et al., "Not only look, but also listen: Learning multimodal violence detection under weak supervision," in *Proc. IEEE Eur. Conf. Comput. Vis.*, 2020, pp. 322–339.

[5] W.-F. Pang, Q.-H. He, Y.-J. Hu, and Y.-X. Li, "Violence detection in videos based on fusing visual and audio information," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, 2021, pp. 2260–2264.

[6] H. Shi, L. Wang, S. Zhou, G. Hua, and W. Tang, "Abnormal ratios guided multi-phase self-training for weakly-supervised video anomaly detection," *IEEE Trans. Multimedia*, vol. 26, pp. 5575–5587, 2024.

[7] G. Huang et al., "Glance and focus networks for dynamic visual recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 4, pp. 4605–4621, Apr. 2023.

[8] W. Wang, D. Tran, and M. Feiszli, "What makes training multi-modal classification networks hard?" in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12695–12705.

[9] C. Du et al., "Improving multi-modal learning with uni-modal teachers," 2021, *arXiv:2106.11059*.

[10] Y. Sun, S. Mai, and H. Hu, "Learning to balance the learning rates between various modalities via adaptive tracking factor," *IEEE Signal Process. Lett.*, vol. 28, pp. 1650–1654, 2021.

[11] T. Winterbottom, S. Xiao, A. McLean, and N. A. Moubayed, "On modality bias in the TVQA dataset," 2020, *arXiv:2012.10210*.

[12] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE/WACV Appl. Comput. Vis. (WACV)*, 2018, pp. 839–847.

[13] S. Iqbal, A. N. Qureshi, M. Alhussein, K. Aurangzeb, and M. S. Anwar, "AD-CAM: Enhancing interpretability of convolutional neural networks with a lightweight framework-from black box to glass box," *IEEE J. Biomed. Health Inf.*, vol. 28, no. 1, pp. 514–525, Jan. 2024.

[14] K. Klein, O. De Candido, and W. Utschick, "Interpretable classifiers based on time-series motifs for lane change prediction," *IEEE Trans. Intell. Veh.*, vol. 8, no. 7, pp. 3954–3961, Jul. 2023.

[15] W. Liu and Y. Li, "Optimal strategy model checking in possibilistic decision processes," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 53, no. 10, pp. 6620–6632, Oct. 2023.

[16] W. Zheng, L. Yan, and F.-Y. Wang, "So many heads, so many wits: Multimodal graph reasoning for text-based visual question answering," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 54, no. 2, pp. 854–865, Feb. 2024.

[17] H. Dui, S. Zhang, M. Liu, X. Dong, and G. Bai, "IoT-enabled real-time traffic monitoring and control management for intelligent transportation systems," *IEEE Internet Things J.*, vol. 11, no. 9, pp. 15842–15854, May 2024.

[18] P. Wu, X. Liu, and J. Liu, "Weakly supervised audio-visual violence detection," *IEEE Trans. Multimedia*, vol. 25, pp. 1674–1685, 2023.

[19] E. Bermejo, O. Deniz, G. Bueno, and R. Sukthankar, "Violence detection in video using computer vision techniques," in *Proc. Int. Conf. Comput. Anal. Images Patterns*, 2010, pp. 332–339.

[20] T. Hassner, Y. Itcher, and O. Kliper-Gross, "Violent flows: Real-time detection of violent crowd behavior," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 1–6.

[21] B. Schölkopf, R. C. Williamson, A. J. Smola, J. Shawe-Taylor, and J. C. Platt, "Support Vector Method for Novelty Detection," in *Proc. Adv. Neural Inf. Process. Syst.*, 2000, pp. 582–588.

[22] S. Sudhakaran and O. Lanz, "Learning to detect violent videos using convolutional long short-term memory," in *Proc. IEEE Int. Conf. Adv. Video Signal Based Surveillance*, Aug. 2017, pp. 1–6.

[23] A. Hanson, K. Pnvr, S. Krishnagopal, and L. Davis, "Bidirectional convolutional LSTM for the detection of violence in videos," in *Proc. Eur. Conf. Comput. Vis. Workshop*, 2018, pp. 280–295.

[24] B. Peixoto et al., "Toward subjective violence detection in videos," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2019, pp. 8276–8280.

[25] A. Singh, D. Patil, and S. N. Omkar, "Eye in the sky: Real-time drone surveillance system (DSS) for violent individuals identification using scatternet hybrid deep learning network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, 2018, pp. 1629–1637.

[26] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 6479–6488.

[27] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 776–794.

[28] Y. Tian, G. Pang, Y. Chen, R. Singh, J. Verjans, and G. Carneiro, "Weakly-supervised video anomaly detection with robust temporal feature magnitude learning," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 4975–4986.

[29] L. Yang, Z. Wu, J. Hong, and J. Long, "MCL: A contrastive learning method for multimodal data fusion in violence detection," *IEEE Signal Process. Lett.*, vol. 30, pp. 408–412, 2023.

[30] S. Li, F. Liu, and L. Jiao, "Self-training multisequence learning with transformer for weakly supervised video anomaly detection," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 1395–1403.

[31] Y. Chen, Z. Liu, B. Zhang, W. Fok, X. Qi, and Y.-C. Wu, "MGFN: Magnitude-contrastive glance-and-focus network for weakly-supervised video anomaly detection," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 387–395.

[32] H. Zhou, J. Yu, and W. Yang, "Dual memory units with uncertainty regulation for weakly supervised video anomaly detection," in *Proc. AAAI Conf. Artif. Intell.*, 2023, pp. 3769–3777.

[33] H. K. Joo, K. Vo, K. Yamazaki, and N. Le, "CLIP-TSA: CLIP-assisted temporal self-attention for weakly-supervised video anomaly detection," in *Proc. IEEE Int. Conf. Image Process.*, 2023, pp. 3230–3234.

[34] Z. Yang, J. Liu, and P. Wu, "Text prompt with normality guidance for weakly supervised video anomaly detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2024, pp. 18899–18908.

Wen-Dong Jiang (Graduate student member, IEEE) received the B.S. degree from the Department of Multimedia and Game Science Development, Lunghwa University of Science and Technology, Taoyuan, Taiwan, in 2021, and the M.S. degree from the Department of Computer Science and Information Engineering, Ming Chuan University, Taoyuan, in 2023. He is currently pursuing the Ph.D. degree in computer science and information engineering with Tamkang University, New Taipei City, Taiwan.

His research interests include explainable machine learning and its applications in smart cities.

Dr. Jiang is also the winner of the Best Student Paper Award at WASN 2024 and DLT 2024, and currently serves as a reviewer for ETASR.

Chih-Yung Chang (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University, Taoyuan City, Taiwan, in 1995.

He is currently a Distinguished Professor with the Department of Computer Science and Information Engineering and the Department of Artificial Intelligence, Tamkang University, New Taipei, Taiwan. His current research interests include artificial intelligence, deep learning and machine learning, the Internet of Things, and wireless sensor networks.

Dr. Chang is the Co-Chair of ACM SIGMOBILE, Taiwan. He has been serving as an associate guest editor for several SCIindexed journals, including *International Journal of Ad Hoc* and *Ubiquitous Computing* since 2011, *Journal of Applied Science and Engineering* since 2018, *International Journal of Distributed Sensor Networks* from 2012 to 2014, *IET Communications* in 2011, *Telecommunication Systems* in 2010, *Journal of Information Science and Engineering* in 2008, and *Journal of Internet Technology* from 2004 to 2008.

Ming-Yang Su received the B.S. degree from the Department of Computer Science and Information Engineering, Tunghai University, Taichung City, Taiwan, in 1989, the M.S. degree from the Department of Computer Science and Information Engineering, National Central University, Taoyuan City, Taiwan, in 1991, and the Ph.D. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1997.

He is currently an Associate Professor with the Department of Computer Science and Information Engineering, Ming Chuan University, Taoyuan, Taiwan. His research interests include network security, intrusion detection/prevention, malware detection, wireless ad hoc network, and wireless sensor networks.

Yue-Shi Lee received the Ph.D. degree in computer science and information engineering from National Taiwan University, Taipei, Taiwan, in 1997.

He is currently a Professor with the Department of Computer Science and Information Engineering, Ming Chuan University, Taoyuan, Taiwan. His initial research interests were computational linguistics and Chinese language processing, and over time he evolved toward data warehousing, data mining, information retrieval and extraction, and Internet technology.

Diptendu Sinha Roy (Senior Member, IEEE) received the Ph.D. degree in engineering from the Birla Institute of Technology, Ranchi, India, in 2010.

In 2016, he joined as an Associate Professor with the Department of Computer Science and Engineering, National Institute of Technology (NIT) Meghalaya, Shillong, India, where he has been working as the Chair since January 2017. Prior to his stint at NIT Meghalaya, he worked with the Department of Computer Science and Engineering, National Institute of Science and Technology, Berhampur, India. His current research interests include software reliability, distributed and cloud computing, and the Internet of Things, specifically applications of artificial intelligence/machine learning for smart integrated systems.

Dr. Roy is a member of the IEEE Computer Society.