mathematics

MDPI

Article

Int.2D-3D-CNN: Integrated 2D and 3D Convolutional Neural Networks for Video Violence Recognition

Wimolsree Getsopon [1], Sirawan Phiphitphatphaisit [2], Emmanuel Okafor [3] and Olarik Surinta [1,*]

[1] Multi-Agent Intelligent Simulation Laboratory (MISL) Research Unit, Department of Information Technology, Faculty of Informatics, Mahasarakham University, Mahasarakham 44150, Thailand; wimolsree.g@msu.ac.th
[2] Department of Information System, Faculty of Business Administration and Information Technology, Rajamangala University of Technology Isan Khon Kaen Campus, Khon Kaen 40000, Thailand; sirawan.ch@rmuti.ac.th
[3] SDAIA-KFUPM Joint Research Center for Artificial Intelligence, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia; emmanuel.okafor@kfupm.edu.sa
* Correspondence: olarik.s@msu.ac.th

Abstract

Intelligent video analysis tools have advanced significantly, with numerous cameras installed in various locations to enhance security and monitor unusual events. However, the effective detection and monitoring of violent incidents often depend on manual effort and time-consuming analysis of recorded footage, which can delay timely interventions. Deep learning has emerged as a powerful approach for extracting critical features essential to identifying and classifying violent behavior, enabling the development of accurate and scalable models across diverse domains. This study presents the Int.2D-3D-CNN architecture, which integrates a two-dimensional convolutional neural network (2D-CNN) and 3D-CNNs for video-based violence recognition. Compared to traditional 2D-CNN and 3D-CNN models, the proposed Int.2D-3D-CNN model presents improved performance on the Hockey Fight, Movie, and Violent Flows datasets. The architecture captures both static and dynamic characteristics of violent scenes by integrating spatial and temporal information. Specifically, the 2D-CNN component employs lightweight MobileNetV1 and MobileNetV2 to extract spatial features from individual frames, while a simplified 3D-CNN module with a single 3D convolution layer captures motion and temporal dependencies across sequences. Evaluation results highlight the robustness of the proposed model in accurately distinguishing violent from non-violent videos under diverse conditions. The Int.2D-3D-CNN model achieved accuracies of 98%, 100%, and 98% on the Hockey Fight, Movie, and Violent Flows datasets, respectively, indicating strong potential for violence recognition applications.

Keywords: 2D convolutional neural network; 3D convolutional neural network; deep feature extraction; frame-level deep features; video violence recognition

MSC: 68T07

Academic Editors: Paolo Crippa and Ioannis Tsoulos

Received: 21 June 2025
Revised: 30 July 2025
Accepted: 17 August 2025
Published: 19 August 2025

Citation: Getsopon, W.; Phiphitphatphaisit, S.; Okafor, E.; Surinta, O. Int.2D-3D-CNN: Integrated 2D and 3D Convolutional Neural Networks for Video Violence Recognition. *Mathematics* **2025**, *13*, 2665. https://doi.org/10.3390/math13162665

1. Introduction

The technology involved in video surveillance systems has significantly advanced. Many private and public areas have cameras installed to secure, monitor, and prevent unusual events. The recognition of violent or abnormal behavior in videos is categorized under human action recognition, which is a crucial process for improving the efficiency of

the detection system for various tasks, such as monitoring theft in a mall and attacks in a park, patient behavior tracking in hospitals, and the detection of falls by the elderly [1–3]. However, monitoring unusual events in videos still relies on manual methods for analyzing and detecting visual information, which is labor-intensive and time-consuming [4]. Furthermore, human operations are inaccurate and cannot be acted upon immediately when anomalous events occur [5]. Ensuring the safety of human beings and preventing violence are of utmost importance. Therefore, developing an automatic system with efficiency and accuracy is especially crucial for analyzing and detecting the risk of violent human behavior and providing timely warnings.

There are two types of violent behaviors: global and local abnormal actions [6]. A local abnormal action refers to the actions of one or two individuals that deviate from the norm and result in harm, such as physical violence, fighting, altercations, theft, and accidents. In comparison, a global action involves many people engaging in violent behavior, such as fighting, sniping, and assaults. The challenge lies in distinguishing between violent behaviors and common gestures when categorizing violent videos. Instances of violent and simple behaviors share close similarities; for example, fighting may appear similar to hugging, and hitting may resemble raising hands in greeting.

Video recognition differs from image recognition in that training the model requires multiple frames for each video, whereas image recognition relies on a single image. Videos with high frame rates are particularly time-consuming to process. Additionally, factors such as different viewpoints, scales, and video resolutions; the number of people in the area; crowd scenes; and dynamic environments can significantly impact recognition performance, rendering action recognition more challenging in capturing practical and discriminative features.

Numerous researchers have proposed methods to enhance the effectiveness of video violence recognition [7–9]. In the literature, violence recognition involves feature extraction and classification. Several years ago, local and global feature extraction methods were employed as handcrafted approaches to recognize violence in surveillance videos. For instance, Souza et al. [7] proposed a violence detector based on local spatiotemporal features. Meanwhile, Das et al. [8] utilized the histogram of oriented gradients method to extract gradient edges and the orientation in localized areas of an image. Additionally, various studies have suggested approaches for global feature extraction. For example, Gao et al. [9] enhanced the violent flow feature descriptor by incorporating orientation information from optical flow, specifically oriented violent flow. This approach considers both magnitude and orientation information, encoding the obtained features into a bag of words. Subsequently, a classifier, such as a support vector machine, is employed to recognize violence in the video.

Deep learning is a core methodology widely used across various areas of machine learning due to its models' strong ability to learn from given data [10]. A convolutional neural network (CNN) represents one of the most effective neural networks for deep learning. Many researchers utilize a CNN as a robust deep feature extraction method. Khan et al. [11] proposed a lightweight deep learning method that employs the softmax function to classify frames based on spatial features. Carneiro et al. [12] utilized a pre-trained model of the VGG16 architecture to generate spatial features, temporal features, rhythm features, and depth information from videos for violent detection. Their approach significantly improved recognition efficiency.

Additionally, Soliman et al. [13] employed the pre-trained VGG16 and long short-term memory (LSTM) models to extract both spatial and temporal features from videos. Meanwhile, Ullah et al. [14] and Li et al. [15] presented the effectiveness of the 3D-CNN architecture for spatiotemporal feature extraction. Ullah et al. [14] achieved good recognition performance using 3D-CNNs to learn complex sequential patterns in surveillance video

streams for violence prediction. Similarly, Li et al. [15] obtained an effective recognition model by utilizing the 3D-CNN model to extract spatiotemporal features of multiplayer violence, further emphasizing the potential of these methods.

1.1. Contribution

This study aims to enhance the performance of violent video recognition while maintaining low computational complexity. The main contributions are as follows:

1. A lightweight architecture is constructed for violent video recognition by integrating MobileNet-based 2D-CNNs with a simplified 3D-CNN module. The proposed framework is designed to achieve an optimal balance between classification accuracy and computational efficiency. By combining spatial features extracted from individual frames with temporal patterns captured across sequences, the model effectively represents the essential characteristics of violent actions in video data.
2. The proposed method employs spatial features extracted by the 2D-CNN component as input to the 3D-CNN module, instead of processing raw video data directly. This approach significantly reduces the computational cost while preserving the capacity of the model to learn spatiotemporal patterns for accurate violence detection.

1.2. Paper Outline

Section 2 provides a comprehensive review of related studies on violent video recognition. Section 3 introduces our proposed method, which integrates 2D-CNNs and 3D-CNNs for violence recognition. The violent video datasets used in the experiments are described in Section 4, followed by the evaluation metrics presented in Section 5. Sections 6 and 7 report the experimental results and provide an in-depth discussion. Finally, Section 8 summarizes the main contributions and outlines directions for future research.

2. Related Work

In this section, we provide a concise overview of violence recognition in videos. Additionally, we discuss relevant research on deep learning techniques, including CNNs, spatial and temporal feature extraction, and 3D-CNNs, that have been proposed for violence recognition in video data.

2.1. Violence Recognition in Videos

In previous studies, violence recognition in videos had relied on a handcrafted feature extraction approach to distinguish violent from nonviolent actions. First, the robust features are encoded and aggregated using encoding strategies, and then machine learning is applied as a classifier [16]. For instance, Dalal and Triggs [17] introduced the histogram of oriented gradients (HOG) method, which quantifies the occurrences of gradients in an image. These gradients are extracted from localized areas, resulting in robust features. Subsequently, the support vector machine (SVM) method is employed to create a model that classifies the localized areas as either containing people or not. Their approach has achieved superior results and has led many researchers [8,18] to adopt their methodology for feature extraction, recognition, and detection tasks. Das et al. [8] proposed a system for detecting violent situations in videos. They employed the HOG method as a feature descriptor to extract robust features from the images. These features were then trained using various classifiers. Subsequently, a majority voting technique was utilized as the final decision mechanism to determine whether a video clip contained violence.

Numerous research studies have investigated and analyzed motion characteristics in videos with the aim of detecting violence. Souza et al. [7] introduced a framework for detecting video violence. In their method, the videos were initially divided into various im-

ages, and interest points were subsequently detected. Local descriptor methods, specifically the scale-invariant feature transform (SIFT) and space–time interest points (STIPs), were then employed to extract robust features from the identified interest points. Subsequently, a visual codebook was constructed using the bag of visual words (BoW) approach. Finally, the SVM method was utilized to create a model that classified instances as non-violent or violent.

Nievas et al. [19] employed the BoW method to construct a visual codebook based on the SIFT and motion SIFT (MoSIFT) features. Subsequently, these features were classified using the SVM classifier. To assess the efficacy of their proposed method, they gathered the Hockey Fight dataset, comprising 1000 video clips categorized into fights and non-fights from hockey games. In their experimental evaluation, the proposed method achieved an accuracy exceeding 90% on the Hockey Fight dataset. Xu et al. [20] proposed a violence detection framework based on MoSIFT and sparse coding. Initially, local spatiotemporal features, referred to as low-level features, were extracted using the MoSIFT algorithm. Then, the kernel density estimation (KDE) method was employed to select features from the MoSIFT descriptor to enhance feature discriminability. Subsequently, sparse coding was applied to transform these low-level features into mid-level representations that capture highly discriminative information. The max-pooling method was then utilized to create a feature vector (video-level features). Finally, the SVM classifier was trained on these video-level features. Their proposed method achieved accuracies of 89.05% and 94.3% on the Crowd Violence and Hockey Fight datasets, respectively.

Hassner et al. [21] introduced the violent flow (ViF) descriptor for real-time detection of violent crowd behavior. The ViF descriptor uses the optical flow method to compute a sequence of frames, which calculates the magnitude of changes between consecutive frames. The features extracted using the ViF descriptor are also fed into a linear SVM classifier to construct a robust model. The results indicated that their method achieved an accuracy of 81.31% on the Crowd Violence dataset. However, their method yielded results with an accuracy of 82.90% on the Hockey Fight dataset. Furthermore, Gao et al. [9] enhanced the ViF descriptor by incorporating orientation information from optical flow, namely the oriented violent flow (OViF) descriptor. The OViF descriptor considers both the motion magnitude and motion orientation information derived from optical flow. The results showed that the OViF descriptor achieved superior performance when compared to the ViF descriptor. Additionally, when integrated with the SVM classifier, the OViF descriptor surpassed the ViF descriptor, achieving an accuracy of 84.20%. The combination of the ViF and OViF descriptors with the Adaboost and SVM classifiers increased violence detection accuracy and achieved an accuracy of 87.50% on the Hockey Fight dataset.

Table 1 presents a summary of the methods used for violence recognition in videos.

Table 1. Comparison of methods for violence recognition in videos.

Reference	Year	Method	Dataset (Acc.%)
Nievas et al. [19]	2011	(SIFT-BoW, MoSIFT) + SVM	Hockey Fight (90.00%)
Hassner et al. [21]	2012	ViF + SVM	Crowd Violence (81.31%) Hockey Fight (82.90%)
Xu et al. [20]	2014	(MoSIFT, KDE, Sparse Coding) + SVM	Crowd Violence (89.05%) Hockey Fight (94.30%)
Gao et al. [9]	2016	(ViF, OViF) + Adaboost/SVM	Hockey Fight (87.50%)
Patil et al. [18]	2017	HOG + SVM	UT-Interaction (N/A)
Das et al. [8]	2019	HOG + Various Classifiers + Majority Voting	Hockey Fight (86.00%)
Li et al. [16]	2019	3D-CNN	Hockey Fight (98.30%) Movie (100%) Violent Flows (97/17%)

2.2. Convolutional Neural Networks

Recently, deep learning methods, specifically CNN architectures, have gained widespread adoption as robust feature extractors for recognizing violent activities in videos [11,12,22,23]. Khan et al. [11] employed the lightweight MobileNet model to train salient frames selected from videos. The process involved three main steps: Firstly, the videos were segmented into frames. Subsequently, a histogram-based method was applied to each frame to identify salient frames. Secondly, they employed a kernel-density-based saliency estimation method, selecting key frames based on maximum information. Thirdly, they fine-tuned the MobileNet model using a transfer learning approach to classify videos as violent or non-violent. Their method achieved accuracies of 87.0%, 97.0%, and 99.5% on the Hockey Fight, Violent Scene Detection, and Violent in Movie datasets, respectively.

Zhou et al. [22] introduced the FightNet architecture, which was adapted from the inception of batch normalization (BN-Inception). Within the FightNet architecture, two inputs are employed. The first input comprises RGB images, while the optical flow algorithm is applied to these RGB images and used as the second input. Additionally, Carneiro et al. [12] proposed a multi-stream fight detection framework that utilizes four input modalities: RGB, optical flow, depth estimation, and visual rhythms. Initially, each input image undergoes processing in a modified VGG16 model to calculate weight vectors. Subsequently, these images are fed into separate VGG16 models (called individual stream learners) pre-trained on the ImageNet and UCF101 datasets. The pre-training ensures that the model can effectively distinguish motion patterns in videos. Finally, the outputs from each stream learner are combined and used for classification via the SVM method. The FightNet model [22] achieved remarkable accuracy rates on both the Movie and Hockey Fight datasets. Specifically, it attained 100% accuracy on the Movie dataset and 97% accuracy on the Hockey Fight dataset. Furthermore, the multi-stream fight detection approach [12] also achieved strong performance. The FightNet model reached 100% accuracy on the Movie dataset and 89.10% accuracy on the Hockey Fight dataset.

The summary of the methods based on CNNs is presented in Table 2.

Table 2. Comparison of methods for convolutional neural networks.

Reference	Year	Method	Dataset (Acc.%)
Zhou et al. [22]	2017	FightNet with RGB, Optical Flow	Movie (100%) Hockey Fight (97.00%)
Khan et al. [11]	2019	MobileNet	Hockey Fight (87.00%) Violent Scene Detection (97.00%) Violent in Movie (99.50%)
Carneiro et al. [12]	2019	Multi-Stream (SVM)	Movie (100%)

2.3. Spatial and Temporal Feature Extraction

Extracting robust features is an essential process in reducing computational costs and dimensionality [24]. In traditional video recognition, researchers commonly extract robust features based on various techniques such as interest points, regions of interest, and geometry [7,17,19,20]. However, these methods can be challenging to apply to complex scenarios and only sometimes guarantee extracted robustness and high-accuracy recognition.

Deep learning is highly effective in extracting deep features from images, a capability that surpasses traditional feature extraction methods [25]. However, the process of extracting deep features from a single image in an image recognition task differs significantly from extracting deep features from a sequence of images in a video recognition task. In video recognition, the feature extraction method is responsible for computing spatial features and extracting relevant information from consecutive images. Consequently, extensive

research has focused on extracting both spatial and temporal features using a combination of 2D-CNNs and LSTM architectures [13,26–28].

Sudhakaran and Lanz [26] introduced a new CNN architecture that combines CNNs with convolutional LSTM (convLSTM) to detect the characteristics of violent scenes in videos. The CNN and convLSTM focus on extracting discriminant features, while the convLSTM component encodes spatial and temporal frame-level changes from consecutive frames. Sumon et al. [27] initially proposed a CNN architecture comprising only two convolutional layers. Subsequently, dropout and batch normalization layers were attached to each convolutional layer. After that, a fully connected layer with the softmax function was employed. Additionally, they utilized the LSTM architecture for training and classifying violent and non-violent categories. Finally, they combined both the CNN and LSTM in their approach. However, their experimental results showed that combining the CNN and LSTM yielded lower precision performance than using only the CNN model.

Hanson et al. [29] introduced a new spatiotemporal encoder architecture, which represents a unique fusion of spatial encoding, a temporal encoder, and classifier components. In their proposed architecture, the spatial encoding module is specifically designed to extract spatial features using a VGG13 architecture. Meanwhile, the bidirectional convLSTM (BiConvLSTM) model captures and encodes the spatiotemporal information from each video frame. The resulting spatiotemporal encoding was then subjected to an elementwise max-pooling operation and sent to the classifier, a fully connected layer. Their architecture achieved impressive accuracy rates of 96.32%, 98.1%, and 100% on the Violent Flows, Hockey Fight, and Movie datasets, respectively.

Furthermore, considerable research efforts have been directed toward combining CNN and LSTM models for video violence recognition. Researchers have utilized several advanced CNN architectures, such as VGGNet, ResNet, and MobileNet, to extract robust spatial features [11,30–32]. Moreover, they have explored the fusion of multiple CNN architectures rather than relying solely on a single CNN model [5].

Table 3 summarizes the methods based on spatial and temporal feature extraction.

Table 3. Comparison of methods for spatial and temporal feature extraction.

Reference	Year	Method	Dataset (Acc.%)
de Souza et al. [7]	2010	(SIFT, STIP, and BoVW) + SVM	Violent Dataset on Social Networks (85.35%)
Sudhakaran & Lanz [26]	2017	CNNs + ConvLSTM	Hockey Fight (97.10%) Movie (100%) Violent Flows (94.57%)
Sumon et al. [27]	2019	VGG13 + BiConvLSTM	Violent Collected from YouTube (89.79%)
Soliman et al. [13]	2019	VGG16 + LSTM	Hockey Fight (95.10%) Movie (99.00%) Violent Flows (90.01%)
Hanson et al. [29]	2019	VGG13 + Bi-ConvLSTM	Violent Flow (96.32%) Hockey Fight (98.10%) Movie (100%)
Naik & Gopalakrishna [28]	2021	Mask-RCNN + LSTM	KTH (93.40%) Weizmann (73.10%)
Jahlan & Elrefaei [5]	2022	(AlexNet, SqueezeNet) + ConvLSTM	Movie (100%) Violent Flows (96.00%)
Vosta &Yow [30]	2022	ResNet50 + ConvLSTM	UCF Crime (62.50%)
Getsopon & Surinta [31]	2022	Fusion-CNNs + BiLSTM	Hockey Fight (97.20%) Movie (100%) Violent Flows (96.77%)

2.4. Three-Dimensional Convolutional Neural Networks

Three-dimensional CNNs are now utilized to analyze videos, detecting anomalies in crowded scenes [33–35]. Hu et al. [33] proposed a novel method for detecting spatial–temporal cuboids by utilizing varied cell-size structures. They employed an optical flow algorithm to identify moving objects based on cell size. Subsequently, the 3D-CNN was applied to detect abnormalities within the cuboids. Maqsood et al. [35] employed a 3D-CNN architecture with five convolutional layers to extract anomalous spatiotemporal features from videos. They evaluated their approach using the UCF Crime dataset, which contains 14 different classes of anomalies. The results indicated that their proposed 3D-CNN outperformed existing state-of-the-art methods for anomaly detection on the UCF Crime dataset.

Furthermore, Kokila et al. [34] and Pratama et al. [36] employed two-stream 3D-CNN architectures. In their approach, the first stream utilized a 3D-CNN to extract contextual and spatial features, while the second stream focused on extracting temporal deep motion features using a 3D-CNN optical flow. Kokila et al. [34] combined the output of these two streams through a concatenation operation, followed by an attentive bidirectional LSTM (BiLSTM) architecture, to classify videos as normal or abnormal. In contrast, Pratama et al. [36] attached a softmax function to each stream and then aggregated the output from both streams to obtain the final result of their proposed method.

Keceli and Kaya [37] addressed the classification of violent activity in videos using transfer deep features and a 3D-CNN. Their approach comprised several essential steps. Initially, a person detection algorithm combined the HOG with a linear SVM classifier. Subsequently, the AlexNet architecture was employed to extract robust spatial features from the region of interest corresponding to the detected individual. These spatial features were then reshaped and computed using trilinear interpolation, resulting in a 3D prism structure. Finally, these 3D prism features were fed into the 3D-CNN model. Their proposed method was evaluated on three benchmark datasets: Violent Flows, Hockey Fight, and Movie. Impressively, their approach achieved an accuracy of 88% on the Violent Flows dataset, 92.90% on the Hockey Fight dataset, and 98.7% on the Movie dataset.

Table 4 summarizes the methods based on 3D-CNNs.

Table 4. Comparison of methods for 3D-CNNs.

Reference	Year	Method	Dataset (Acc.%)
Hu et al. [33]	2020	Parallel Spatial-Temporal CNN	UCSD (96.73%) UMN (96.37%)
Maqsood et al. [35]	2021	3D ConvNets	UCF Crime (45.00%)
Kokila et al. [34]	2023	2MPD-3DFCN -AttBiDLSTM	UCSD (96.00%) UCF Crime (87.20%) LV (81.00%)
Pratama et al. [36]	2023	Two-Stream 3D-CNN	RWF-2000 (90.50%)
Keceli & Kaya [37]	2023	3D Prism + 3D-CNN	ViolentFlows (88.00%) Hockey FIght (92.90%) Movie (98.70%)

2.5. Transformer Models

Transformer-based models are increasingly recognized as a powerful alternative to traditional CNNs for video-understanding tasks, including action and violence recognition. The transformer architectures frequently encounter challenges in modeling long-range temporal dependencies, ensuring scalability, and capturing global contextual information. To address these limitations, transformer-based models, such as ViViT, TimeSformer, and the video Swin transformer, have been proposed, offering a novel framework for both

video classification and understanding [38–40]. By employing self-attention mechanisms, transformers enable each frame to dynamically interact with all others, thereby facilitating the comprehensive modeling of global temporal relationships.

Particularly, the TimeSformer architecture [38] is distinguished by its separation of spatial and temporal attention, which facilitates more efficient training and improved generalization, a methodology commonly referred to as joint spatiotemporal feature learning. ViViT addresses the complexity of video input by factorizing attention into distinct spatial and temporal components. This decomposition significantly reduces the computational overhead while preserving the ability of the model to capture essential spatiotemporal features required for accurate video classification [39]. The video Swin transformer factorizes attention into spatial and temporal components. Hence, it also captures global self-attention by shifting windows across layers so that information can propagate across the entire video sequence over layers, allowing global dependencies to be learned indirectly [40].

Singh et al. [41] introduced an end-to-end framework for automatic violence detection in videos based on the video vision transformer (ViViT) architecture. In their study, various image augmentation techniques, such as Gaussian blur, random rotation, uniform perturbations, and horizontal flipping, were applied during the training process. However, the ViViT model requires 56 input video frames, which increases the computational demand. Although this increases the computational demand, the method achieved impressive accuracy rates of 97.14% on the Hockey Fight dataset and 98.46% on the Violent Flows dataset.

Despite the transformative impact of transformers on video classification, particularly their capacity to capture long-range spatiotemporal dependencies, these models present several critical limitations. One significant drawback is their high computational complexity. The self-attention mechanism scales quadratically with the input sequence length, leading to substantial memory usage and prolonged training and inference times. Additionally, transformers are highly data-dependent, often requiring large-scale annotated datasets to achieve effective generalization. Inherent inductive biases, such as spatial locality, are also absent in transformer architectures, reducing efficiency compared to CNNs in visual processing tasks [42,43].

To address the limitations of existing approaches, this study proposes a new framework that integrates deep spatial features from two lightweight CNNs (MobileNetV1 and MobileNetV2) with a 3D-CNN model to effectively capture both spatial and temporal information. In contrast to the deep features fusion technique [5], which uses AlexNet and SqueezeNet combined with ConvLSTM modules for temporal modeling, the proposed method eliminates the complexity and high computational cost of recurrent layers by applying a 3D-CNN for temporal aggregation, thereby improving efficiency and scalability. Compared to the multi-stream (SVM) approach [12], which relies on handcrafted high-level features (e.g., depth and visual rhythm) and traditional classifiers (SVM), the proposed framework employs an end-to-end deep learning model to learn discriminative representations directly from raw video data, avoiding manual feature engineering. Furthermore, while the two-stream 3D-CNN architecture [36] independently processes RGB and optical flow streams and combines them at a later stage, the proposed model learns from fused spatial features extracted by two CNNs and feeds them directly into a single 3D-CNN stream, thus reducing redundancy and enhancing the modeling of temporal information.

Details of the proposed integrated 2D- and 3D-CNN architecture are described in Section 3.

3. Proposed Integrated 2D and 3D Convolutional Neural Networks

The effectiveness of violence recognition in videos is essential for a range of applications, including crowd events. A primary contributors to this effectiveness is the extraction of robust deep features, which are fundamental to the accurate identification and classification of violent behaviors within video data [7,19,20,25,26]. The primary objective of this research is to design a deep learning architecture capable of efficiently classifying violent actions in videos. To achieve this, we propose a new approach that integrates robust deep features with a 3D convolutional neural network (3D-CNN). Specifically, we extracted spatial features using state-of-the-art lightweight CNN architectures and subsequently create temporal features by learning from the spatial representations using the 3D-CNN. The proposed model is named Int.2D-3D-CNN.

The architecture of the proposed Int.2D-3D-CNN model, as presented in this study, is illustrated in Figure 1. A detailed explanation of the proposed architecture, which integrates deep features extracted by a 2D-CNN with a 3D-CNN, is provided in the following subsections.

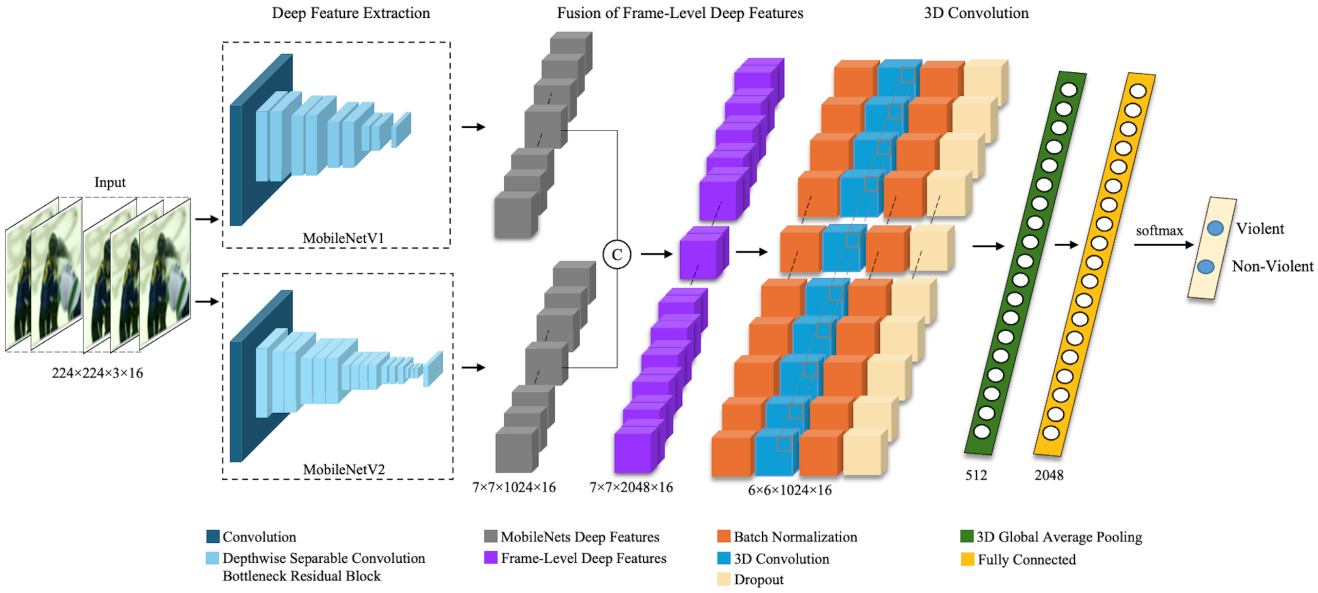


Figure 1. The proposed Int.2D-3D-CNN architecture for video violence recognition.

3.1. Deep Feature Extraction

Deep feature extraction, the primary operation in identifying robust deep features, was carried out using the 2D-CNN model, which is widely recognized for its effectiveness in image and video recognition tasks and was our chosen architecture. In the context of violent video analysis, we focused on deep feature extraction at the frame level. Each frame was processed by two pre-trained lightweight 2D-CNN models, namely MobileNetV1 and MobileNetV2. Brief details of the two lightweight models are explained below.

3.1.1. MobileNetV1

MobileNetV1 [44] is a lightweight architecture designed by using depthwise separable convolutions to construct deep networks. It separates the standard convolution process into two distinct layers: depthwise and pointwise convolution operations, collectively known as depthwise separable convolutions. The application of depthwise separable convolutions significantly reduces the number of network parameters, making the architecture more efficient. The MobileNetV1 architecture begins with a standard convolution layer followed by 13 depthwise separable convolution blocks, a global average pooling layer, and a

fully connected layer for classification. A brief overview of the standard convolution and depthwise convolution operations is provided below.

In the standard convolution operation, a kernel of size $D_K \times D_K$, where D_K is the spatial dimension of the square kernel, performs multiplications across the entire image to generate a feature map (G) of size $D_G \times D_G$. Here, D_G represents the spatial width and height of the square output feature map. The standard convolution operation is illustrated in Figure 2a.

The depthwise separable convolution operation, a specialized form of convolutional operation, is partitioned into two processes: depthwise convolution and pointwise convolution [44]. In the first process, depthwise convolution, a single convolutional kernel is utilized for each input channel, contrasting standard convolutions that execute computations across the entire input channels. The second process, called pointwise convolution, creates a linear combination of the output by applying a 1×1 convolution that integrates the outputs derived from the depthwise convolution [38]. The depthwise convolution and pointwise convolution operations are illustrated in Figure 2b and 2c, respectively.

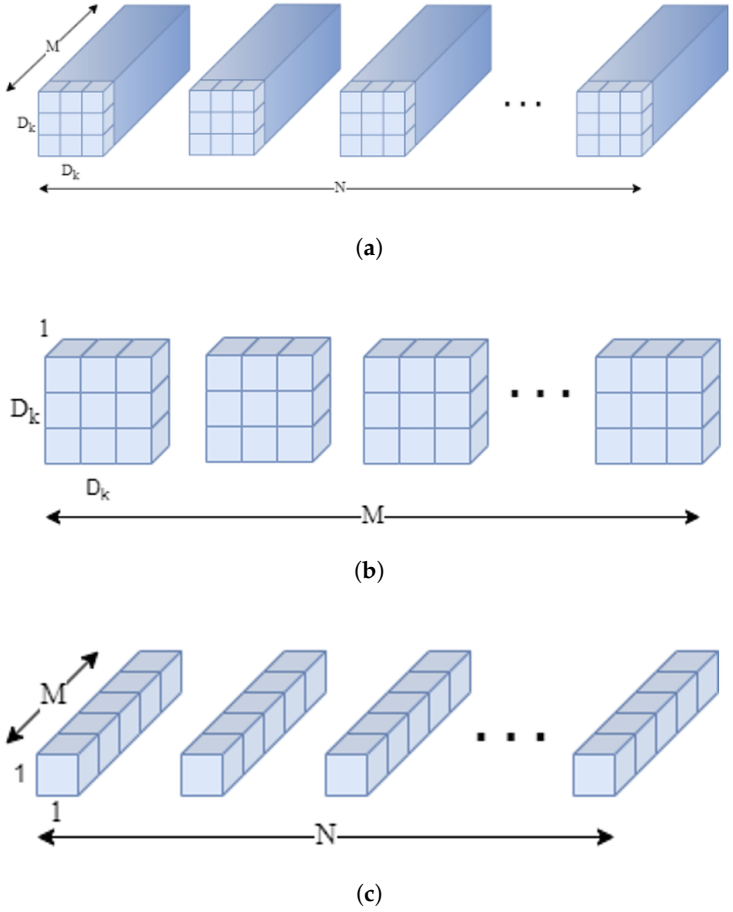


Figure 2. Architecture of the convolution operations: (**a**) the standard convolution kernel, (**b**) the depthwise convolution, and (**c**) the pointwise convolution [44].

The number of parameters in a standard convolution operation is given by $D_K \times D_K \times M \times N$, while for a depthwise separable convolution, it is reduced to $M \times D_K \times D_K + M \times N$. In terms of the computational cost, the standard convolution requires approximately $D_K \times D_K \times M \times N \times D_F \times D_F$ operations, whereas the depthwise separable convolution reduces this to $D_K \times D_K \times M \times D_F \times D_F + M \times N \times D_F \times D_F$. Here, D_K is the kernel size (e.g., three for a 3×3 filter); M is the number of input channels; N is the number of output channels; and D_F is the spatial dimension (width or height) of the input feature map.

Furthermore, in the standard convolution operation, the feature map (O), characterized by dimensions of $D_K \times D_K \times N$, is derived from a kernel (K) with dimensions of $D_K \times D_K$. The kernel is applied to the entire input image (F) with dimensions of $D_K \times D_K \times M$. The standard convolution operation is computed using Equation (1), where

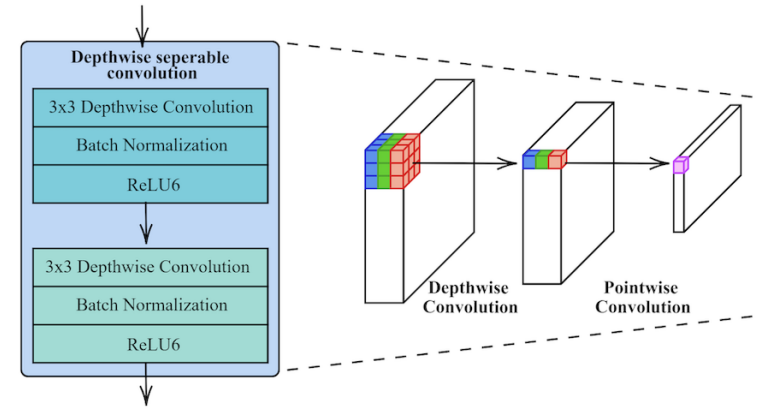
$$O_{k,l,n} = \sum_{i,j,m} K_{i,j,m,n} \cdot F_{k+i-1,l+j-1,m} \quad (1)$$

In the depthwise separable convolution, firstly, a depthwise convolution $(\hat{K})$ with dimensions of 3×3 is applied to each image channel. Subsequently, a kernel with dimensions of 1×1 (pointwise convolution) is employed to combine the output derived from the depthwise convolution operation [45]. The computation of the depthwise convolution and pointwise convolution is represented as Equations (2) and (3), where

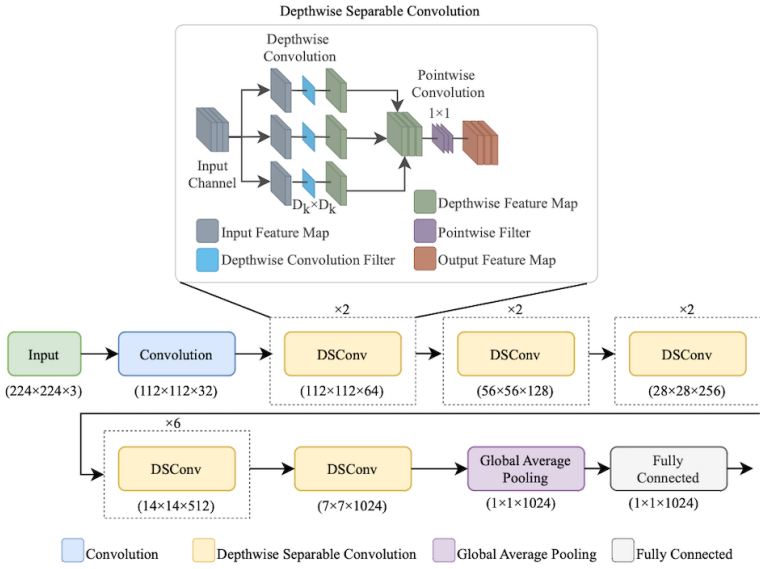
$$\hat{O}_{k,l,m} = \sum_{i,j} \hat{K}_{i,j,m} \cdot F_{k+i-1,\ l+j-1,m} \quad (2)$$

$$O_{k,l,m} = \sum_{m} \tilde{K}_{m,n} \cdot \hat{O}_{k-1,l-1,m} \quad (3)$$

The depthwise separable convolution block and the architecture of MobileNetV1 are illustrated in Figure 3a,b.



(**a**)



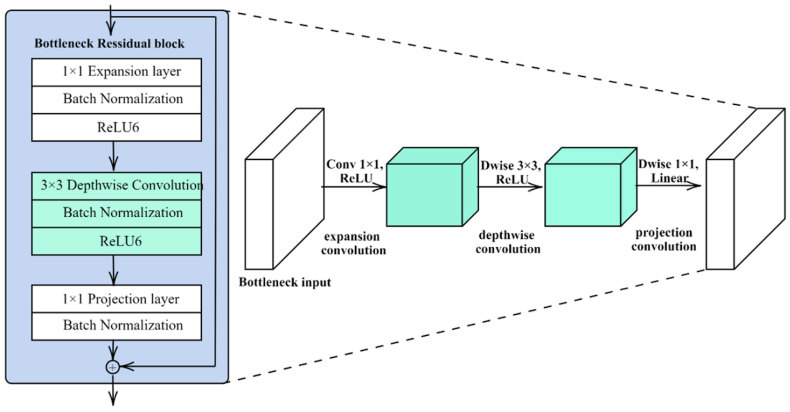
(**b**)

Figure 3. Architecture of MobileNetV1: (**a**) depthwise separable convolution block and (**b**) overall architecture of MobileNetV1 [44].

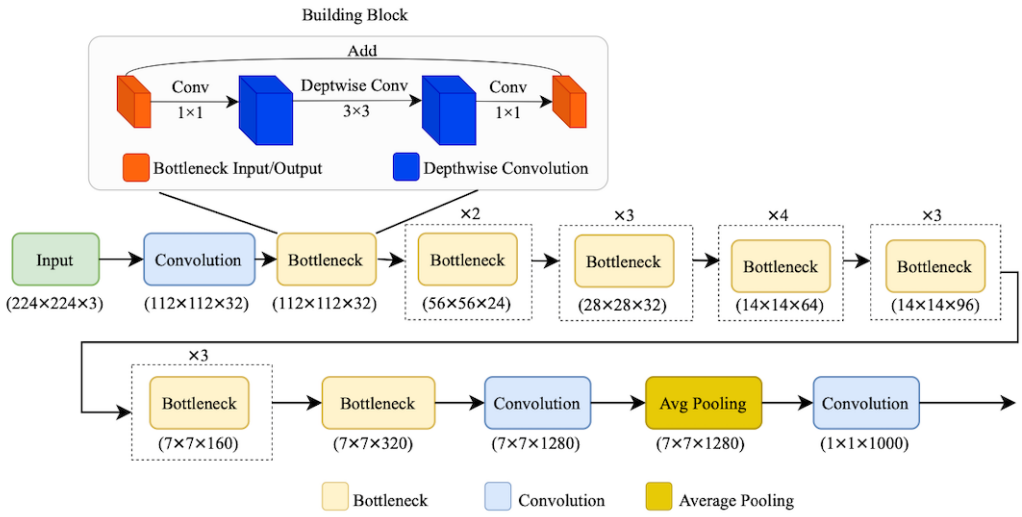
3.1.2. MobileNetV2

MobileNetV2 [46] incorporates an inverted residual module combined with a linear bottleneck. The inverted residual module first expands the channels to a higher dimensionality and then applies a depthwise separable convolution to compress the dimensionality back to match the input dimension. A skip connection is also strategically employed within the inverted residual module, linking the beginning and the end of the convolution blocks. Additionally, the network expands the layer using a 1×1 convolution, followed by a 3×3 depthwise convolution to reduce the network parameters, and then compresses the dimensionality again using another 1×1 convolution.

The MobileNetV2 architecture consists of an initial standard convolution layer and 17 inverted residual blocks, followed by a convolution layer, a global average pooling layer to aggregate the features, and a fully connected layer for classification. The bottleneck residual block and architecture of MobileNetV2 are shown in Figure 4a,b.



(**a**)



(**b**)

Figure 4. Illustration of MobileNetV2: (**a**) bottleneck residual block and (**b**) overall architecture of MobileNetV2 [46].

3.2. Fusion of Frame-Level Deep Features

In the fusion process of frame-level deep features, this study aggregates spatial representations extracted by two state-of-the-art CNNs: MobileNetV1 and MobileNetV2. These networks were employed to capture spatial characteristics from individual video frames, which are called frame-level deep features. Specifically, the extracted features from both

MobileNetV1 and MobileNetV2 have dimensions of $D_G \times D_G \times N \times F$, where $D_G \times D_G$ denotes the spatial resolution of the square output feature maps (7×7), N indicate the number of channels (set to 1024), and F represents the number of video frames (set to 16). To fuse the frame-level deep features from the two MobileNet architectures, a concatenation operation was applied. This operation is mathematically expressed in Equation (4).

$$F_{\text{concat}} = \text{Concat}(F_{\text{MobileNetV1}}, F_{\text{MobileNetV2}}) \tag{4}$$

where F_{concat} denotes the concatenated feature map obtained by combining $F_{\text{MobileNetV1}}$ and $F_{\text{MobileNetV2}}$ along the channel dimension. Both $F_{\text{MobileNetV1}}$ and $F_{\text{MobileNetV2}}$ represent the frame-level deep feature maps extracted from MobileNetV1 and MobileNetV2, respectively, each with dimensions of $D_G \times D_G \times N \times F$. After concatenation, the resulting feature map has dimensions of $D_G \times D_G \times 2N \times F$, reflecting the doubling of the channel dimension due to deep feature integration.

Consequently, the final frame-level deep feature representation is configured as $7 \times 7 \times 2048 \times 16$. These concatenated features are then passed to the subsequent 3D-CNN module for spatiotemporal learning, as detailed in the following section.

3.3. Three-Dimensional Convolution

Due to the limitations of the 2D-CNN approach, which only processes spatial information within individual frames, we employed a 3D-CNN architecture that effectively captures complex spatiotemporal features from sequence patterns [33]. The 3D-CNN enables the network to understand patterns and dynamic content across a sequence of video frames. The proposed 3D-CNN performs two primary functions. First, it refines the learning of spatial features at the individual frame level. Second, it extracts temporal features by analyzing the patterns across the sequence of frames.

The operation of 3D convolution is executed by employing a 3D kernel that slides over sequential frames, utilizing element-wise multiplication for computation. Within our 3D-CNN architecture, the input for the 3D layer is derived from the fusion of frame-level features. Consequently, the depth size (D) depends on the number of frames. The computation of the 3D convolution operation is represented by Equation (5).

$$O(i,j,k) = \sum_{p=0}^{kH-1} \sum_{q=0}^{kW-1} \sum_{r=0}^{kD-1} I(i+p, j+q, k+r) \cdot K(p,q,r) \tag{5}$$

where I represents the input volume (e.g., video frames) with dimensions of $H \times W \times D$ (height, width, and depth), while K is the 3D kernel with dimensions of $kH \times kW \times kD$. Note that, in our implementation, the depth size corresponds to 16 consecutive video frames.

To capture spatiotemporal features from sequence patterns, we proposed a 3D-CNN architecture that included a batch normalization layer, a 3D convolution layer, a dropout layer, and a global average pooling layer. The 3D convolution layer used a kernel size of $1 \times 2 \times 2$ with a stride of 1. The spatiotemporal features were reduced by transforming the feature map into 1024 feature maps. Subsequently, a global average pooling layer reduced the feature size to 512, followed by a fully connected layer with 2048 units. The final output consisted of two nodes (violent and non-violent) using a softmax function, as shown in Figure 5. In the proposed network, ReLU was applied as the activation function following the 3D convolution layer, and a dropout rate of 0.4 was employed.

We compared the proposed 3D-CNN architecture with C3D [47], a well-known 3D-CNN architecture, in terms of floating-point operations per second (FLOPS) [48] (see Equation (12)) and the number of network parameters. The proposed 3D-CNN achieved

superior performance in FLOPS, with values of 5.21×10^2 for our model and 6.17×10^2 for C3D. Additionally, the proposed 3D-CNN had 10.504 million parameters, which was significantly fewer than the 78 million parameters of C3D. The network configuration of the proposed 3D-CNN architecture is detailed in Table 5.

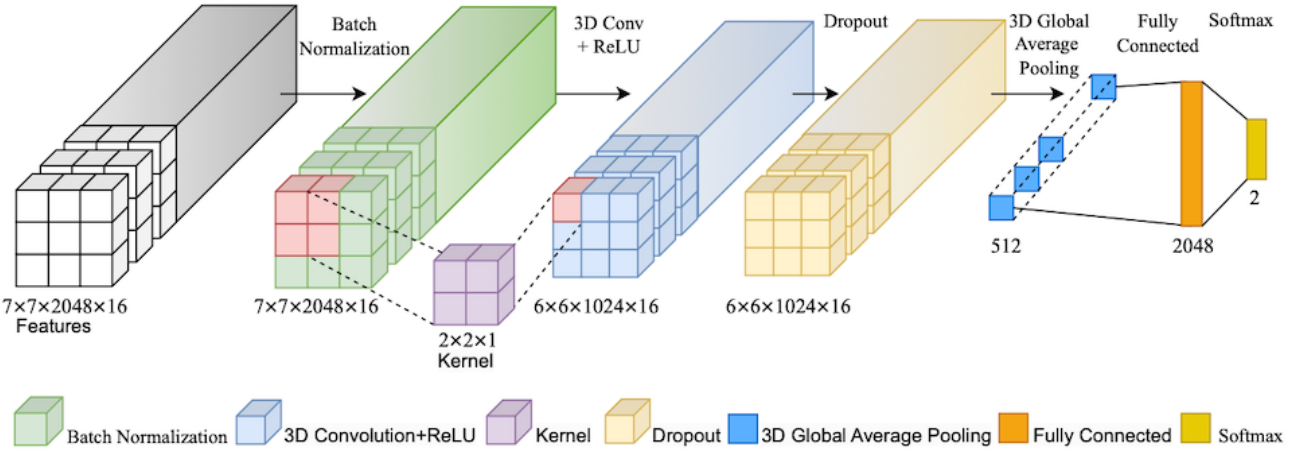


Figure 5. Architecture of the proposed 3D-CNN.

Table 5. Network architecture and configuration of the proposed 3D-CNN.

Layer	**Kernel Size** $(W \times H \times D)$	**Input Size** $(W \times H \times D \times F)$	**Output Size** $(W \times H \times D \times F)$	**Parameter** (M)
BN	-	$7 \times 7 \times 2048 \times 16$	$7 \times 7 \times 2048 \times 16$	0.008
3D Conv	$2 \times 2 \times 1$	$7 \times 7 \times 2048 \times 16$	$7 \times 7 \times 2048 \times 16$	8.389
BN	-	$6 \times 6 \times 1024 \times 16$	$6 \times 6 \times 1024 \times 16$	0.004
Dropout	-	$6 \times 6 \times 1024 \times 16$	$6 \times 6 \times 1024 \times 16$	-
3D GAP	-	$6 \times 6 \times 1024 \times 16$	512	-
FC	-	512		2.099
softmax	-	2048	2	0.004
Total parameters				10.504
FLOPS $(\times 10^2)$				5.21

BN is the batch normalization layer. *GAP* is the global average pooling layer. *FC* is the fully connected layer. *W* is the width. *H* is the height. *D* is the dimensional. *F* is the feature map, and *M* is a million.

4. Violent Video Datasets

Three violent video datasets, namely Hockey Fight, Movie, and Violent Flows, were used as benchmarks to evaluate the proposed architecture. Each dataset comprises two classes: violent and non-violent. Additionally, the combined violence dataset was constructed for training purposes. The details of all datasets are described in the following subsections.

4.1. Hockey Fight Dataset

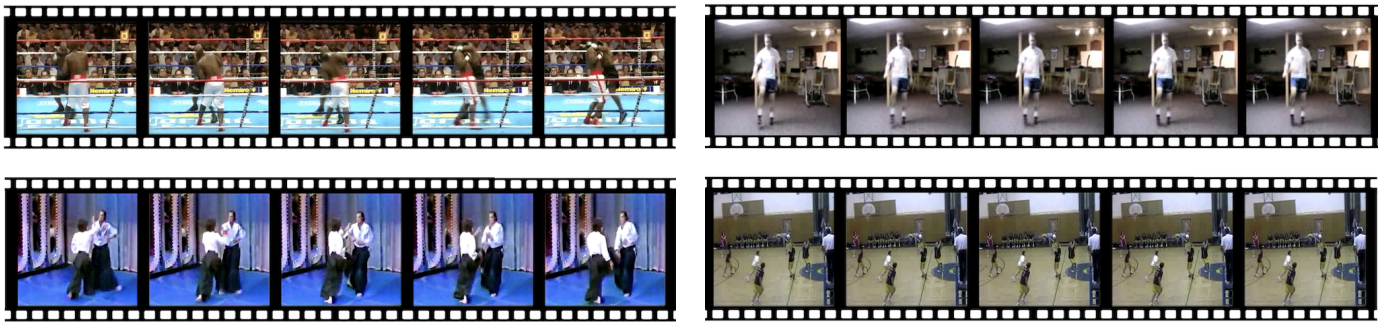
The Hockey Fight dataset was collected by Nievas et al. [19]. It consists of 1000 short video clips extracted from hockey games, with each video containing 50 frames at a resolution of 720×576 pixels. The videos are categorized into two classes: fight (representing violent actions) and non-fight (indicating non-violent actions). The dataset is balanced, comprising 500 videos in each class. Figure 6 illustrates examples of violent and non-violent videos from the Hockey Fight dataset by showing five video frames.

(**a**) (**b**)

Figure 6. Examples of (**a**) violent and (**b**) non-violent videos from the Hockey Fight dataset [19].

4.2. Movie Dataset

In 2011, Nievas et al. [19] collected the Movie dataset, which consists of 200 video clips sourced from action movies. The dataset is evenly divided into 100 non-fight and 100 fight scenes. Each video clip has a duration of approximately 2 s. The video resolution for the non-fight category is 720 × 480, while the resolution for the fight category is 720 × 576 pixels. Examples of violent and non-violent videos from the Movie dataset are illustrated in Figure 7.



(**a**) (**b**)

Figure 7. Examples of (**a**) violent and (**b**) non-violent videos from the Movie dataset [19].

4.3. Violent Flows Dataset

The Violent Flows dataset, which was collected by Hassner et al. [21], is a collection of 246 videos sourced from YouTube. This dataset is split into two categories, namely violent and non-violent, with each category containing 123 videos. The average duration of each video is approximately 3.60 s. The videos were processed using the DivX codec software, yielding a resolution of 320 × 240 pixels. The Violent Flows dataset comprises videos captured from a distance during violent occurrences in crowded environments, as shown in Figure 8.

4.4. Combined Violence Dataset

The motivation for combining violence datasets is due to the fact that each dataset exhibits diverse characteristics in terms of violent scenarios, camera angles, environments, and motion patterns. Additionally, combining datasets allows the model to learn from a broader range of non-violent patterns. The primary objective of this combination is to construct a more diverse and representative training set.

To construct the combined training dataset, each of the three violence datasets was initially partitioned using an 80:20 ratio for training and testing. Only the training sets were then used for combination. Specifically, the combined violence dataset includes 800 videos

from the Hockey Fight dataset, 160 videos from the Movie dataset, and 197 videos from the Violent Flows dataset, resulting in a total of 1157 training videos.

(**a**) (**b**)

Figure 8. Examples of (**a**) violent and (**b**) non-violent video from the Violent Flows dataset [21].

5. Evaluation Metrics

To evaluate the performance of the recognition method, we used various metrics, including accuracy, a receiver operating characteristics (ROC) curve, the area under the curve (AUC), a precision–recall curve, and the area under the precision–recall curve (AUC-PR).

The evaluation metrics are defined as follows:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{6}$$

$$Recall = \frac{TP}{TP + FN} \tag{7}$$

$$Precision = \frac{TN}{TN + FP} \tag{8}$$

where TP, TN, FP, and FN stand for true positive, true negative, false positive, and false negative, respectively. Recall and precision represent the true positive rate (TPR) and the true negative rate (TNR) [49,50].

The ROC curve was used to analyze the performance of the violence recognition methods. Two parameters, recall and precision, were computed and plotted on a graph, with precision on the x-axis and recall on the y-axis [49]. Further, the AUC [51] was calculated to measure performance across all recognition thresholds, indicating how well the model distinguishes between classes. An AUC of 1 indicates perfect model performance. The AUC was computed using Equation (9).

$$AUC = \sum_{i=1}^{n-1} (FPR_{i+1} - FPR_i) \times \left(\frac{TPR_{i+1} + TPR_i}{2} \right) \tag{9}$$

where FPR stands for the false positive rate [51] and is calculated using Equation (10), where

$$FPR = \frac{TP}{TP + FN} \tag{10}$$

The AUC-PR was evaluated by plotting precision against recall [51] and calculated using Equation (11), where

$$AUC - PR = \sum_{i=1}^{n-1} (Recall_{i+1} - Recall_i) \times \left(\frac{Precision_{i+1} + Precision_i}{2} \right) \tag{11}$$

Additionally, the floating-point operations per second (FLOPS) were calculated to evaluate the computational efficiency of the deep learning architectures and to determine how many floating-point operations can be executed each second [48]. A lower FLOPS value indicates better efficiency and reduced computational resource requirements. The FLOPS were calculated using Equation (12):

$$FLOPS = \frac{FLOPs}{cycle} \times \frac{cycles}{second} \times cores \tag{12}$$

where *FLOPs* denotes the number of floating-point operations that need to be performed by the model, *cycles* (or *clock cycles*) refers to the number of cycles needed for a processor to execute one floating-point operation, $\frac{cycles}{second}$ represents the clock speed of the processor, and cores refer to the number of processing cores in the CPU or GPU.

6. Experiment Results

Experimental Settings: The proposed Int.2D-3D-CNN architecture was implemented on the Google Colab platform utilizing a Tesla T4 GPU. The implementation was performed in Python 3.13.0, using the Keras 3.10.0 API within TensorFlow 2.19.0 as the deep learning framework. Model training employed the stochastic gradient descent (SGD) optimizer with a momentum of 0.9, and experiments were conducted using learning rates of 0.01, 0.001, 0.0001, and 0.00001. Batch sizes of four and eight were tested, and each model was trained for 500 epochs. The proposed architecture used 16 video frames as the input, which helped reduce the computational cost.

According to previous studies, the Movie dataset was partitioned into training and test sets using an 80:20 ratio. For the Hockey Fight and Violent Flows datasets, both 80:20 and 75:25 ratios were evaluated. To ensure an unbiased distribution and avoid systematic ordering effects, data splitting was performed randomly using the '`train_test_split`' function from the Scikit-learn 1.4.1 library [52], with a fixed random seed to ensure reproducibility.

To illustrate the evaluation results, quantitative comparisons are provide between the proposed architecture and other state-of-the-art deep learning techniques on three benchmark violent video datasets, namely Hockey Fight, Movie, and Violent Flows, as presented in Sections 6.1–6.5.

6.1. Results of MobileNetV1, MobileNetV2, and C3D for Violence Recognition

The efficiency of pre-trained CNN models in recognizing violent content in videos was evaluated using MobileNetV1 and MobileNetV2 across three violent video datasets. Sixteen non-overlapping frames were selected for each video, with frames chosen by skipping one frame at a time to minimize data redundancy. The selected frames were sized $16 \times 224 \times 224 \times 3$, where 16 represents the number of frames, 224 represents the width and height, and 3 represents the channels. Both MobileNetV1 and MobileNetV2 models were independently retrained on the three datasets, with the final layer replaced by a softmax layer to classify violent or non-violent videos. The results were compared using different batch sizes (four and eight) and learning rates (0.01, 0.001, 0.0001, and 0.00001). The experimental results for MobileNetV1 and MobileNetV2 are presented in Tables 6 and 7, respectively.

Table 6 presents the experimental results obtained using the MobileNetV1 architecture. The model achieved an accuracy of 95.99% on the Hockey Fight dataset when trained with a batch size of eight and learning rates ranging from 0.01 to 0.0001. On the Movie dataset, the highest accuracy of 98.00% was attained using a batch size of four and a learning rate of 0.001. In the Violent Flows dataset, the model consistently achieved an accuracy of 91.94% across all configurations.

Table 6. Evaluation results of MobileNetV1 for violence recognition.

Dataset	Batch Size	Learning Rate	Accuracy (%)	Time	
				Train (h)	Test (ms)
Hockey Fight	4	0.01	94.80	0.58	2
		0.001	95.99	0.59	
		0.0001	95.20	0.58	
		0.00001	95.20	0.58	
	8	0.01	95.99	0.53	2
		0.001	95.99	0.53	
		0.0001	95.99	0.53	
		0.00001	95.60	0.53	
Movie	4	0.01	95.99	0.11	1
		0.001	98.00	0.11	
		0.0001	93.99	0.11	
		0.00001	93.99	0.11	
	8	0.01	92.00	0.11	2
		0.001	93.99	0.11	
		0.0001	95.99	0.11	
		0.00001	92.00	0.11	
Violent Flows	4	0.01	91.94	0.14	1
		0.001	91.94	0.14	
		0.0001	91.94	0.14	
		0.00001	91.94	0.14	
	8	0.01	91.94	0.13	1
		0.001	91.94	0.13	
		0.0001	91.94	0.13	
		0.00001	91.94	0.13	

Table 7 presents the results for MobileNetV2. The model achieved accuracies of 95.99% on the Hockey Fight dataset, 98.00% on the Movie dataset, and 91.94% on the Violent Flows dataset. The optimal configuration for the Hockey Fight dataset was a batch size of four with a learning rate of 0.00001. For the Movie dataset, all configurations yielded the highest observed accuracy. In the Violent Flows dataset, the best performance was achieved with a batch size of eight and a learning rate of 0.01.

MobileNetV2 achieved equivalent maximum accuracy performance to MobileNetV1 across the Hockey Fight, Movie, and Violent Flows datasets.

The experiments also included C3D, a 3D convolution architecture, to compare the performance of 2D-CNN and 3D-CNN models. The results for C3D are detailed in Table 8. In the experimental setup, a sequence of 16 consecutive frames was used as the input, each with dimensions of $112 \times 112 \times 3$, representing the width, height, and channels, respectively. The C3D model achieved accuracies of 76.40%, 86.00%, and 75.81% on the Hockey Fight, Movie, and Violent Flows datasets, respectively. However, the experimental results showed that the MobileNet architectures outperformed C3D across all three violence datasets.

Table 7. Evaluation results of MobileNetV2 for violence recognition.

Dataset	Batch Size	Learning Rate	Accuracy (%)	Time	
				Train (h)	Test (ms)
Hockey Fight	4	0.01	95.60	0.68	3
		0.001	95.20	0.69	
		0.0001	95.20	0.68	
		0.00001	95.99	0.53	
	8	0.01	95.20	0.60	3
		0.001	95.20	0.61	
		0.0001	95.20	0.61	
		0.00001	95.20	0.59	
Movie	4	0.01	98.00	0.17	2
		0.001	98.00	0.17	
		0.0001	98.00	0.17	
		0.00001	98.00	0.16	
	8	0.01	98.00	0.16	2
		0.001	98.00	0.15	
		0.0001	98.00	0.16	
		0.00001	98.00	0.17	
Violent Flows	4	0.01	87.10	0.17	2
		0.001	88.71	0.16	
		0.0001	87.10	0.16	
		0.00001	88.71	0.16	
	8	0.01	91.94	0.15	2
		0.001	82.26	0.15	
		0.0001	87.10	0.15	
		0.00001	88.71	0.15	

Table 8. Evaluation results of C3D for violence recognition.

Dataset	Batch Size	Learning Rate	Accuracy (%)	Time		Model Size (M)
				Train (h)	Test (ms)	
Hockey Fight	4	0.0001	76.40	1.87	13	298
		0.00001	72.80	1.75		
	8	0.0001	70.80	1.48	3	
		0.00001	72.80	1.95		
Movie	4	0.0001	86.00	0.24	2	
		0.00001	82.00	0.37		
	8	0.0001	84.00	0.24	2	
		0.00001	84.00	0.38		
Violent Flows	4	0.0001	70.97	0.52	2	
		0.00001	72.58	0.50		
	8	0.0001	70.97	0.48	2	
		0.00001	75.81	0.47		

6.2. Experimental Results of the Proposed Int.2D-3D-CNN Models

As shown in Section 6.1, using only the 3D-CNN architecture for video violence recognition did not result in high accuracy. In contrast, the 2D-CNN models achieved strong performance, reaching 98% accuracy on the Movie dataset. However, the performance was lower on the Hockey Fight and Violent Flows datasets, yielding accuracies of approximately 95% and 91%, respectively.

In this experiment, an architecture was proposed to enhance violence recognition models by integrating the strengths of 2D-CNN and 3D-CNN architectures. Spatial features were extracted from video frames using the lightweight 2D-CNN architectures (MobileNetV1 and MobileNetV2). These robust features were then aggregated and passed to the 3D-CNN model to learn temporal information between adjacent frames. For the 2D-CNN models, features were extracted from the last convolution layer, producing a size of $7 \times 7 \times 1024$. The feature maps were concatenated to form a size of $7 \times 7 \times 2048$ before being fed into the 3D-CNN. To identify the most suitable 3D-CNN model, we designed and experimented with five different 3D-CNN models, as shown in Table 9.

The experimental results for the integrated 2D-CNN and 3D-CNN (Int.2D-3D-CNN) model on three violent datasets are presented in Tables 10–12.

Table 9. Network architectures of five different 3D-CNN models.

Models	**Model 1**	**Model 2**	**Model 3**	**Model 4**	**Model 5**
	Input Deep Feature $(16 \times 7 \times 7 \times 2048)$				
	Batch Normalization $(16 \times 7 \times 7 \times 2048)$				
	Conv3D (1024) K $(1 \times 2 \times 2)$	Conv3D (1024) K $(1 \times 2 \times 2)$	Conv3D (1024) K $(1 \times 2 \times 2)$	Conv3D (1024) K $(1 \times 2 \times 2)$	Conv3D (1024) K $(1 \times 2 \times 2)$
	BN	Conv3D (512) K $(1 \times 2 \times 2)$	Conv3D (512) K $(1 \times 2 \times 2)$	Conv3D (512) K $(1 \times 2 \times 2)$	Conv3D (512) K $(1 \times 2 \times 2)$
	Dropout (0.2)	BN	BN	BN	BN
	GAP (1024)	Dropout (0.2)	Dropout (0.2)	GAP (512)	Dropout (0.2)
	Dense (2048) Dense (2)	GAP (512) Dense (2048)	GAP (512) Dense (2048)	Dense (2048) Dense (2)	GAP (512) Dense (2048)
		Dense (2)	Dense (2)		Dense (2)
Param	10,499,074	6,303,746	11,019,778	11,547,138	11,547,138
FLOPS $(\times 10^2)$	5.21	4.31	5.75	5.75	5.75

Table 10 presents the accuracy performance of five 3D-CNN models (Models 1–5) tested on the Hockey Fight dataset. The integrated 2D-CNN and 3D-CNN (Model 1) model, referred to as Int.2D-3D-CNN (M1), was trained for 2.06 h and required only ten milliseconds (ms) to recognize each video. Additionally, Int.2D-3D-CNN (M1) achieved the highest accuracy of 97.20% when trained with a learning rate of 0.01.

As shown in Table 11, the Int.2D-3D-CNN models achieved impressive performance on the Movie dataset. Specifically, Int.2D-3D-CNN (M1) achieved 100% accuracy with a learning rate of 0.001 and a batch size of eight. Training for Int.2D-3D-CNN (M1) took approximately 0.48 h, and the model required only 11 milliseconds to recognize violence in each video. However, the accuracy of other Int.2D-3D-CNN models was also above 96%.

Table 10. Accuracy performance of the proposed Int.2D-3D-CNN models on the Hockey Fight dataset.

Model	Learning Rate	Batch Size of 4			Batch Size of 8			Model Size (M)
		Time		Acc. (%)	Time		Acc. (%)	
		Train (h)	Test (ms)		Train (h)	Test (ms)		
1	0.01	2.06	10	97.20	2.20	10	96.00	80.17
	0.001	2.05	10	96.00	2.11	10	95.20	
	0.0001	2.06	10	95.60	2.03	10	96.40	
2	0.01	1.27	6	95.60	1.26	6	96.40	48.17
	0.001	1.27	6	96.00	1.26	6	95.20	
	0.0001	1.29	6	96.00	1.26	6	96.00	
3	0.01	2.06	11	95.60	2.27	11	95.60	84.15
	0.001	2.07	11	96.40	2.29	11	95.60	
	0.0001	2.09	11	96.00	2.29	11	95.60	
4	0.01	2.03	11	96.00	2.26	11	95.60	88.17
	0.001	2.06	11	95.60	2.29	11	95.60	
	0.0001	2.06	11	96.40	2.12	11	95.60	
5	0.01	2.38	11	96.00	2.29	10	96.00	88.17
	0.001	2.39	11	95.60	2.31	10	96.00	
	0.0001	2.38	11	95.60	2.36	10	95.60	

Table 11. Accuracy performance of the proposed Int.2D-3D-CNN models on the Movie dataset.

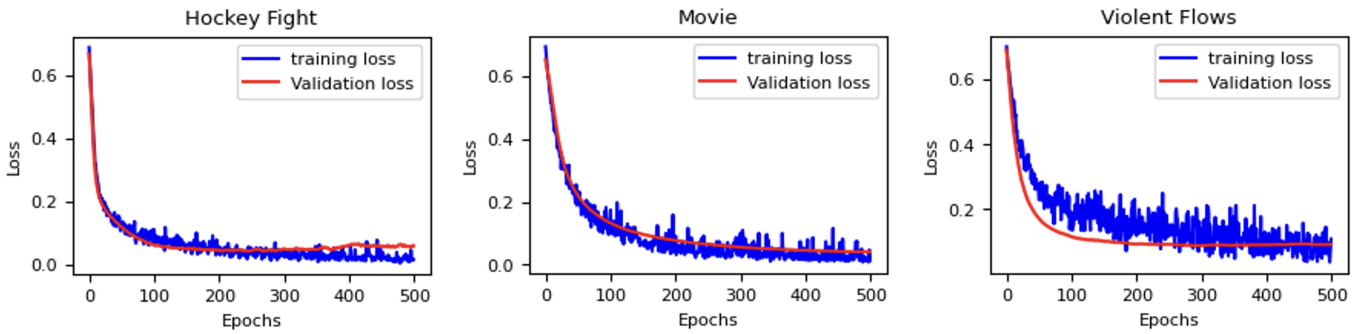
Model	Learning Rate	Batch Size of 4			Batch Size of 8			Model Size (M)
		Time		Acc. (%)	Time		Acc. (%)	
		Train (h)	Test (ms)		Train (h)	Test (ms)		
1	0.01	0.47	9	97.37	0.47	11	97.37	80.17
	0.001	0.48	9	97.37	0.47	11	100	
	0.0001	0.48	9	97.37	0.47	11	97.37	
2	0.01	0.26	6	96.00	0.26	6	96.00	48.17
	0.001	0.26	5	96.00	0.25	6	94.00	
	0.0001	0.27	6	96.00	0.27	6	96.00	
3	0.01	0.52	11	96.00	0.52	11	96.00	84.15
	0.001	0.52	11	96.00	0.52	11	96.00	
	0.0001	0.53	11	94.00	0.53	11	96.00	
4	0.01	0.42	11	96.00	0.42	11	96.00	88.17
	0.001	0.42	11	96.00	0.42	11	96.00	
	0.0001	0.43	11	94.00	0.47	11	94.00	
5	0.01	0.45	9	96.00	0.44	11	96.00	88.17
	0.001	0.45	9	96.00	0.45	11	94.00	
	0.0001	0.47	9	92.00	0.48	11	96.00	

The experimental results in Table 12 indicate that Int.2D-3D-CNN (M1) achieved the highest accuracy on the Violent Flows dataset, outperforming other models with an accuracy of 96.77% at a learning rate of 0.0001 and a batch size of eight.

Table 12. Accuracy performance of the proposed Int.2D-3D-CNN models on the Violent Flows dataset.

Model	Learning Rate	Batch Size of 4			Batch Size of 8			Model Size (M)
		Time		Acc. (%)	Time		Acc. (%)	
		Train (h)	Test (ms)		Train (h)	Test (ms)		
1	0.01	0.47	9	95.65	0.46	11	93.48	80.17
	0.001	0.46	9	95.65	0.46	11	95.65	
	0.0001	0.47	9	93.48	0.46	11	96.77	
2	0.01	0.32	6	87.10	0.30	6	91.94	48.17
	0.001	0.32	6	93.55	0.31	6	93.55	
	0.0001	0.34	6	90.32	0.32	6	91.94	
3	0.01	0.54	11	91.94	0.52	11	93.55	84.15
	0.001	0.55	11	91.94	0.53	11	91.94	
	0.0001	0.53	11	91.94	0.53	11	93.55	
4	0.01	0.54	11	93.55	0.53	11	91.94	88.17
	0.001	0.55	11	93.55	0.52	11	93.55	
	0.0001	0.53	11	93.55	0.53	11	93.55	
5	0.01	0.56	9	91.94	0.55	11	93.55	88.17
	0.001	0.56	9	90.32	0.55	11	93.55	
	0.0001	0.56	9	93.55	0.58	11	93.55	

Figure 9 illustrates the training and validation loss curves of the Int.2D-3D-CNN (M1) model across the three violence datasets. The loss curves gradually decreased until converging at their minimum, indicating close alignment. The closely matched training and validation loss curves indicate the effective learning capability of the proposed model and suggest that overfitting was well managed.



(**a**) (**b**) (**c**)

Figure 9. Training and validation loss curves of the proposed Int.2D-3D-CNN (M1) model on the (**a**) Hockey Fight, (**b**) Movie, and (**c**) Violent Flows datasets.

Based on the experimental results presented in Sections 6.1 and 6.2, the integrated 2D-CNN and 3D-CNN models, particularly Model 1 (Int.2D-3D-CNN (M1)), exhibited the highest performance on all three violence datasets: Hockey Fight, Movie, and Violent Flows.

6.3. Experimental Results of Training Int.2D-3D-CNN Models on the Combined Violence Dataset

In this experiment, the training sets of three violence datasets were combined, resulting in a total of 1157 training videos. The model was then evaluated separately on the test set of each individual dataset. The experimental results on the test set of these evaluations are presented in Tables 13–15.

Table 13. Accuracy results of the proposed Int.2D-3D-CNN models trained on the combined dataset and evaluated on the test set of the Hockey Fight dataset.

Model	Learning Rate	Batch Size of 4			Batch Size of 8			Model Size (M)
		Time		Acc. (%)	Time		Acc. (%)	
		Train (h)	Test (ms)		Train (h)	Test (ms)		
1	0.01	2.35	11	95.60	2.06	11	96.80	80.17
	0.001	2.35	11	96.40	2.24	11	97.60	
	0.0001	2.39	11	96.00	2.32	11	96.80	
2	0.01	1.27	11	96.00	1.28	8	96.40	48.17
	0.001	1.27	11	95.60	1.25	8	96.40	
	0.0001	1.31	11	96.40	1.28	8	96.40	
3	0.01	2.05	11	96.00	2.37	11	94.40	84.15
	0.001	2.07	11	89.60	2.36	11	84.40	
	0.0001	2.11	11	88.80	2.37	11	87.20	
4	0.01	2.03	11	96.40	2.07	11	95.20	88.17
	0.001	2.11	11	96.40	2.12	11	96.40	
	0.0001	2.10	10	96.00	2.21	11	96.00	
5	0.01	2.38	11	95.60	2.29	11	95.60	88.17
	0.001	2.39	11	95.60	2.31	11	96.00	
	0.0001	2.38	11	96.00	2.33	11	96.40	

The results shown in Table 13 present the classification accuracy of the proposed Int.2D-3D-CNN (M1) model, which was evaluated on the test set of the Hockey Fight dataset. The model achieved an accuracy of 97.60% with a learning rate of 0.001 and a batch size of eight.

Table 14. Accuracy results of the proposed Int.2D-3D-CNN models trained on the combined dataset and evaluated on the test set of the Movie dataset.

Model	Learning Rate	Batch Size of 4			Batch Size of 8			Model Size (M)
		Time		Acc. (%)	Time		Acc. (%)	
		Train (h)	Test (ms)		Train (h)	Test (ms)		
1	0.01	0.44	11	98.00	0.47	11	98.00	80.17
	0.001	0.47	11	96.00	0.47	11	96.00	
	0.0001	0.47	11	100.00	0.47	11	96.00	
2	0.01	0.26	6	968.00	0.26	6	96.00	48.17
	0.001	0.26	6	98.00	0.26	6	96.00	
	0.0001	0.27	6	96.00	0.27	5	96.00	
3	0.01	0.42	11	98.00	0.42	11	96.00	84.15
	0.001	0.43	11	98.00	0.42	11	98.00	
	0.0001	0.44	11	96.00	0.45	11	96.00	
4	0.01	0.42	11	98.00	0.42	11	98.00	88.17
	0.001	0.42	11	98.00	0.42	11	96.00	
	0.0001	0.44	11	96.00	0.46	11	96.00	
5	0.01	0.43	11	98.00	0.43	11	96.00	88.17
	0.001	0.43	11	98.00	0.43	11	96.00	
	0.0001	0.44	11	98.00	0.47	11	98.00	

Table 14 presents the accuracy results of the Int.2D-3D-CNN (M1) model evaluated on the test set of the Movie dataset. The highest performance was achieved with a learning

rate of 0.0001 and a batch size of four, yielding an accuracy of 100.00% and indicating optimal classification under this configuration.

Table 15. Accuracy results of the proposed Int.2D-3D-CNN models trained on the combined dataset and evaluated on the test set of the Violent Flows dataset.

Model	Learning Rate	Batch Size of 4			Batch Size of 8			Model Size (M)
		Time		Acc. (%)	Time		Acc. (%)	
		Train (h)	Test (ms)		Train (h)	Test (ms)		
1	0.01	0.51	11	93.55	0.72	11	88.71	80.17
	0.001	0.51	11	88.71	0.72	11	90.32	
	0.0001	0.51	11	85.45	0.72	11	87.10	
2	0.01	0.32	5	87.10	0.31	6	88.71	48.17
	0.001	0.33	6	88.71	0.31	6	88.71	
	0.0001	0.34	5	87.10	0.32	6	88.71	
3	0.01	0.52	11	90.32	0.52	11	90.32	84.15
	0.001	0.52	11	85.48	0.52	11	87.10	
	0.0001	0.52	11	87.10	0.53	11	87.10	
4	0.01	0.52	10	93.55	0.52	11	90.32	88.17
	0.001	0.52	11	87.10	0.52	11	90.32	
	0.0001	0.52	11	87.10	0.53	11	90.32	
5	0.01	0.53	11	91.94	0.53	11	88.71	88.17
	0.001	0.53	11	88.71	0.53	11	88.71	
	0.0001	0.53	11	87.10	0.55	11	88.71	

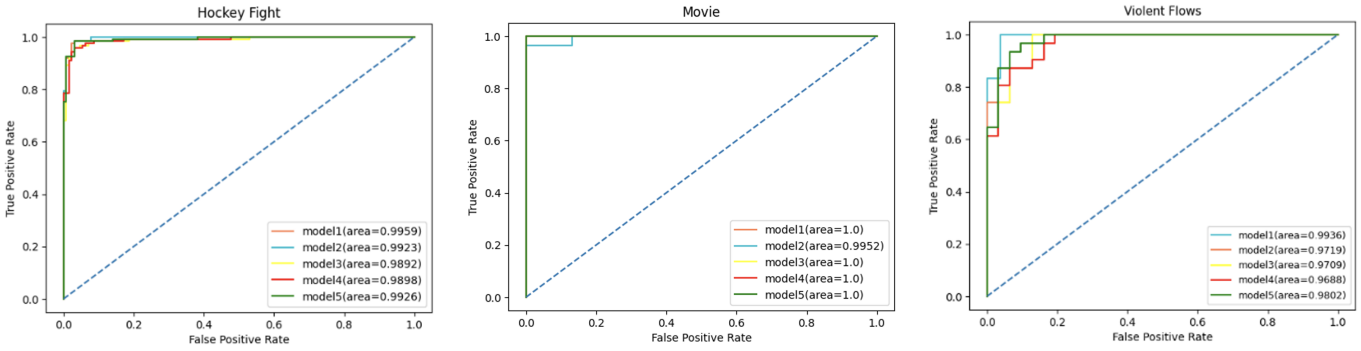
Table 15 shows the accuracy performance of the Int.2D-3D-CNN (M4) model on the test set of the Violent Flows dataset. The highest accuracy, 93.55%, is obtained using a learning rate of 0.01 and a batch size of four.

Based on the experimental results of the Int.2D-3D-CNN models in Sections 6.2 and 6.3, the model trained on the combined dataset outperformed the model trained on the individual dataset when evaluated on the Hockey Fight dataset, achieving a 0.4% improvement in accuracy. However, the accuracy on the Violent Flows dataset was slightly lower, with a decrease of 3.27%. In contrast, both models achieved the same accuracy of 100% on the Movie dataset under both experimental settings.

6.4. Performance Evaluation of Int.2D-3D-CNN Models Using ROC, AUC-PR Curves, and Confusion Matrices

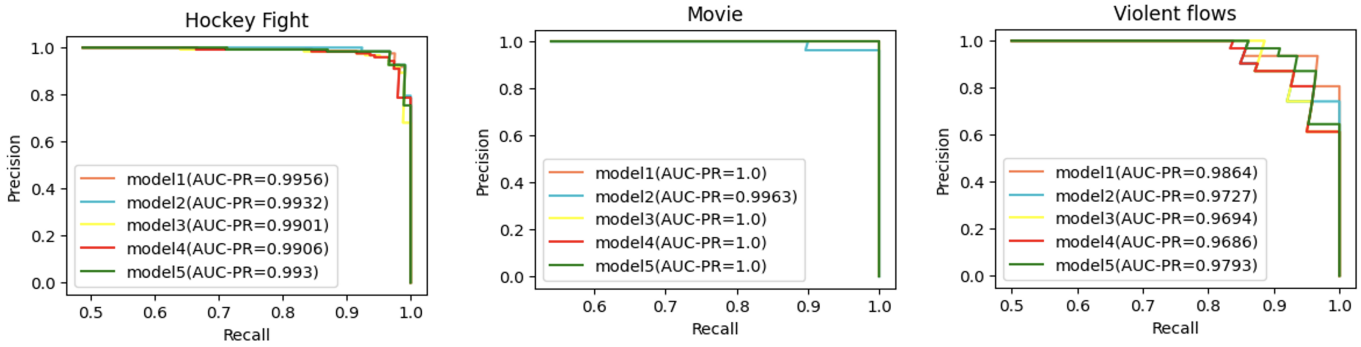
This section evaluates the proposed Int.2D-3D-CNN models using established classification metrics, including ROC curves and AUC-PR, as shown in Figures 10 and 11. In addition, the confusion matrix is presented to highlight the classification and misclassification performance of the Int.2D-3D-CNN (M1) model.

Figure 10 shows the ROC curves for the Int.2D-3D-CNN models evaluated on the Hockey Fight (Figure 10a), Movie (Figure 10b), and Violent Flows (Figure 10c) datasets. The Int.2D-3D-CNN (M1) model achieved impressive AUC values, reflecting its excellent classification accuracy. Specifically, the Hockey Fight dataset yielded an AUC of 0.9956. The Movie dataset resulted in an AUC of 1.0, and the Violent Flows dataset achieved an AUC of 0.9936. These high AUC values confirm the efficacy and reliability of the proposed model in violence recognition, reflecting the performance of the model in achieving high true positive rates while minimizing false positives.



(**a**) (**b**) (**c**)

Figure 10. ROC curves and AUC for the Int.2D-3D-CNN models on the (**a**) Hockey Fight, (**b**) Movie, and (**c**) Violent Flows datasets.



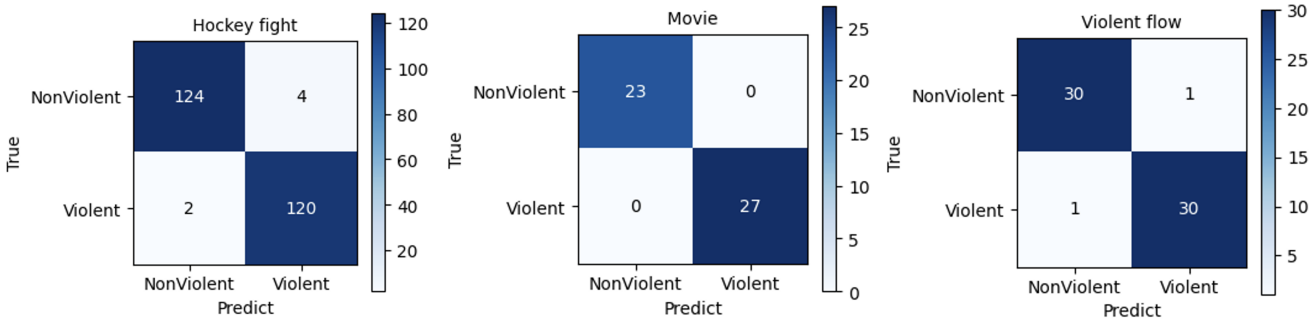
(**a**) (**b**) (**c**)

Figure 11. PR curves and AUC-PR values for the Int.2D-3D-CNN models on the (**a**) Hockey Fight, (**b**) Movie, and (**c**) Violent Flows datasets.

Figure 11 presents the precision–recall (*PR*) curves for the Int.2D-3D-CNN (M1) model evaluated on the Hockey Fight, Movie, and Violent Flows datasets. The model achieved remarkable precision and recall, with the Hockey Fight dataset producing an AUC-PR value of 0.9956, the Movie dataset achieving a perfect AUC-PR of 1.0, and the Violent Flows dataset obtaining an AUC-PR of 0.9864.

Figure 12 illustrates the confusion matrices for the Int.2D-3D-CNN (M1) model evaluated on the Hockey Fight, Movie, and Violent Flows datasets. The matrices indicate the high performance of the proposed model in distinguishing between violent and non-violent instances. For the Hockey Fight dataset, the model misclassified only four non-violent videos as violent and two violent videos as non-violent, indicating minimal *FPs* and *FNs*. A similar pattern of high accuracy and precision was observed with the Violent Flows dataset. Notably, the model achieved perfect classification on the Movie dataset, with no misclassifications.

Figure 13a illustrates an example of a *FP* prediction in the Hockey Fight dataset. In this instance, the players appear very small relative to the scene, limiting the model capacity to extract meaningful spatial features. Additionally, the close proximity of the players can confuse the model, leading to a misclassification of the event as violent when it is not. In Figure 13b, a *FN* prediction is shown, where the model failed to identify a violent event and instead classified it as non-violent. The player in the frame exhibits minimal aggressive motion, which reduced the effectiveness of the 2D-CNN in capturing spatial features related to violence, resulting in this misclassification.



(**a**) (**b**) (**c**)

Figure 12. Confusion matrices for the Int.2D-3D-CNN models on the (**a**) Hockey Fight, (**b**) Movie, and (**c**) Violent Flows datasets.

For the Violent Flows dataset, Figure 13c shows a video where the model incorrectly classified a non-violent scene as violent. The crowded scene and significant motion caused the model to misinterpret the situation as violent. In contrast, Figure 13d shows a *FP* case where a violent video was misclassified as non-violent. The poor lighting in this sequence likely impaired the detection of critical details, such as body movements or aggressive postures, resulting in an incorrect prediction.

(**a**) (**b**)

(**c**) (**d**)

Figure 13. Examples from the Hockey Fight dataset (**a**,**b**) and the Violent Flows dataset (**c**,**d**) illustrate videos that were correctly classified only by our Int.2D-3D-CNN model, while other 2D-CNN and 3D-CNN models misclassified these videos.

6.5. Performance Comparison of the Proposed Int.2D-3D-CNN Model and State-of-the-Art Methods

This section compares the experimental results of the proposed model with state-of-the-art methods for recognizing violent scenes in videos across three datasets. The comparison results are presented in Tables 16–18.

The comparison results in Table 16 indicate that the proposed model achieved an accuracy of 98%, surpassing several state-of-the-art methods. Furthermore, the model maintained strong performance with 97.6% accuracy when the dataset was divided into 75% for training and 25% for testing, outperforming many existing approaches under this configuration. However, the Hybrid CNN and 3D-ResNet+ATDS models achieved slightly higher accuracies of over 99%, exceeding the performance of the proposed method by approximately 1% to 1.3%.

As indicated in Table 17, most methods achieved a perfect accuracy of 100% on the Movie dataset, with the exception of the CNN+ConvLSTM2D method, which attained an

accuracy of 99.2%. Notably, while previous approaches were evaluated using sequences of 20, 40, and 50 frames, the proposed model achieved comparable performance using only 16 frames.

Table 16. Comparison of the Int.2D-3D-CNN (M1) model with state-of-the-art methods on the Hockey Fight dataset.

Year/Reference	Method	No. of Frame	Data Splitting (Train/Test) (%)	Acc. (%)
2019 [11]	MobileNet	N/A	75/25	87.00
2019 [13]	VGG16 + LSTM	20	80/20	88.20
2019 [12]	Multi-Stream (SVM)	40	90/10	89.10
2019 [14]	3D-CNN	16	75/25	96.00
2020 [53]	Keyframe + AlexNet (SVM)	50	80/20	98.14
2021 [28]	VGG13 + BiConvLSTM	20	80/20	96.96
2022 [5]	(AlexNet, SqueezNet) + ConvLSTM	20	80/20	97.00
2022 [54]	VD-Net	N/A	80/20	98.50
2022 [55]	3D ConvNet + Spatial Attention	40	80/20	99.40
2022 [41]	ViViT + Data Augmentation	56	60/40	97.14
2024 [56]	CNN + ConvLSTM2D	20	80/20	97.96
2024 [57]	Hybrid CNN	N/A	N/A	99.30
2025 [58]	3D-ResNet + ATDS	N/A	80/20	99.00
	Int.2D-3D-CNN (M1)	16	80/20	98.00
	(Our proposed)	16	75/25	97.60

Table 17. Comparison of the Int.2D-3D-CNN (M1) model with state-of-the-art methods on the Movie dataset.

Year/Reference	Method	No. of Frame	Data Splitting (Train/Test) (%)	Acc. (%)
2019 [12]	Multi-Stream (SVM)	40	90/10	100
2020 [53]	Keyframe + AlexNet (SVM)	50	80/20	100
2021 [28]	VGG13 + BiConvLSTM	20	80/20	100
2022 [5]	(AlexNet, SqueezNet) + ConvLSTM	20	80/20	100
2024 [56]	CNN + ConvLSTM2D	20	80/20	99.2
	Int.2D-3D-CNN (M1)	16	80/20	100
	(Our proposed)			

As shown in Table 18, the proposed method achieved an accuracy of 98%, outperforming the (AlexNet and SqueezeNet) + ConvLSTM and 3D ConvNet + Spatial Attention approaches. Notably, while these comparative methods trained on 20 and 40 input frames, the proposed model achieved comparable results using only 16 frames. Only the hybrid CNN, Keyframe + AlexNet (SVM), and ViViT + Data Augmentation models slightly outperformed the proposed approach, with marginal accuracy differences ranging from approximately 0.46% to 0.65%. It is important to note that both the ViViT + Data Augmentation and Keyframe + AlexNet (SVM) models utilized a greater number of input frames, and the ViViT + Data Augmentation model further incorporated data augmentation techniques during training.

Table 18. Comparison of the Int.2D-3D-CNN (M1) model with state-of-the-art methods on the Violent Flows dataset.

Year/Reference	Method	No. of Frame	Data Splitting (Train/Test) (%)	Acc. (%)
2019 [13]	VGG16 + LSTM	20	80/20	90.01
2021 [28]	VGG13 + BiConvLSTM	20	80/20	90.60
2020 [53]	Keyframe + AlexNet (SVM)	50	80/20	98.65
2022 [5]	(AlexNet, SqueezNet) + ConvLSTM	20	80/20	96.00
2022 [55]	3D ConvNet + Spatial Attention	40	80/20	97.49
2022 [41]	ViViT + Data Augmentation	56	60/40	98.46
2024 [56]	CNN + ConvLSTM2D	20	80/20	91.01
2024 [57]	Hybrid CNN	16	80/20	98.46
	Int.2D-3D-CNN (M1)	16	80/20	98.00
	(Our proposed)	16	75/25	96.77

7. Discussion

Based on the experimental results and analysis, this section outlines the aspects that impact the performance of the proposed architecture, followed by a discussion of the limitations and directions for future research.

7.1. Performance Analysis

First Analysis: As presented in Section 6.3, training on the combined violence dataset, which integrates videos from the Hockey Fight, Movie, and Violent Flows datasets, does not consistently result in improved performance when compared to training on each dataset individually. This limitation is primarily attributed to domain-specific variations among the datasets, such as differences in scene structure, motion patterns, and representations of violent behavior. These discrepancies introduce distributional inconsistencies that challenge the model to generalize across diverse data sources. Additionally, dataset-specific biases may cause the model to overfit to dominant patterns from one source while failing to capture distinctive features of others.

Second Analysis: The int.2D-3D-CNN architectures often outperforms the use of 2D-CNN architectures alone, particularly for tasks involving video data. While 2D-CNN architectures capture spatial features within individual frames, they cannot understand temporal information and contextual changes over time. In contrast, 3D-CNN architectures extend feature extraction to the temporal dimension, capturing motion and sequential patterns across multiple frames. The 3D-CNNs provide a more detailed representation of dynamic contexts, enhancing the ability of the model to detect temporal dependencies and motion patterns from violent videos, which are essential for action recognition and video classification tasks. By integrating 2D-CNNs and 3D-CNNs, the model capitalized on the strengths of both spatial and temporal feature extraction, improving performance in tasks that require an understanding of both static and dynamic aspects of the data, as presented in Figure 13.

Figure 13a,b show the effectiveness of the integrated 2D-CNN and 3D-CNN model in video violence recognition. The rapid movements of hockey players lead to blurriness across frames, causing misclassification when using only the 2D-CNN or 3D-CNN models. Additionally, in Figure 13c,d, it is seen that the Int.2D-3D-CNN model successfully extracted patterns of violence from the crowd, even under low-light conditions, and correctly classified these videos.

Third Analysis: The consistent accuracy achieved by MobileNetV1 and MobileNetV2 (95.99% on the Hockey Fight dataset, 98.00% on the Movie dataset, and 91.94% on the Violent Flows dataset), as shown in Tables 6 and 7, can be attributed to several factors. First, each dataset may exhibit relatively homogeneous characteristics with limited intra-class variability, which reduces sensitivity to variations in training parameters. Second, the models may have reached their representational capacity on these datasets, resulting in a saturation point where additional hyperparameter tuning yields no substantial improvement in performance. Third, given the relatively small size of the datasets, modifications of the batch size and learning rate are likely to have a limited impact on generalization performance. This observation is further supported by the regularization effect of depthwise separable convolutions in MobileNets, which contribute to model stability across different training configurations.

Fourth Analysis: The comparison with state-of-the-art methods, as presented in Tables 16–18, indicates that the proposed Int.2D-3D-CNN (M1) model outperforms existing approaches on the Hockey Fight and Violent Flows datasets and achieves complete accuracy on the Movie dataset. While LSTM-based methods are effective in capturing long-term dependencies in sequential data, their method tend to be computationally intensive, especially when combined with larger CNN architectures, such as VGG13 or VGG16 [13,28].

In comparison, the proposed Int.2D-3D-CNN (M1) model utilizes lightweight CNNs (MobileNetV1 and MobileNetV2), which not only enhance performance efficiency but also reduce computational costs. Moreover, the Int.2D-3D-CNN (M1) model processes fewer frames (16 frames), compared to LSTM-based methods (typically 20 frames), which generally require a higher number of frames to effectively capture temporal dependencies. By analyzing fewer frames, the proposed model reduces memory usage and processing time, making it more suitable for real-time applications. This indicates the capability of our model to extract essential features for violent video recognition while maintaining computational efficiency. These characteristics are essential for deploying the model in resource-constrained environments or time-sensitive applications, such as video surveillance systems.

7.2. Limitations and Future Work

For future work, efforts will focus on improving both the robustness and efficiency of the proposed model to support real-world deployment. This includes expanding the experimental scope to incorporate more diverse video content that reflects varied environments, lighting conditions, camera angles, and cultural contexts. To this end, we plan to include datasets that encompass a wider range of violent activities, such as riots, domestic violence, street fights, robberies, burglaries, and assaults, using real-world surveillance footage from sources like UCF-Crime, RWF-2000, and VioPeru [59–61]. Furthermore, we aim to evaluate the proposed model on more recent and challenging benchmarks, including RLVS, UBI Fights, CCTV-Fights, and SCVD, under consistent experimental settings. This will enable a more comprehensive assessment of the model in terms of both recognition accuracy and computational efficiency, thereby strengthening its generalization capabilities in diverse and practical scenarios.

To optimize performance for real-time applications, several efficiency-oriented techniques will be explored, including model pruning, quantization, and architectural refinement [62]. In addition, attention mechanisms, including spatiotemporal, temporal, and self-attention, will be investigated to enable the model to focus on salient regions and time frames, particularly in complex scenes where violent behaviors are subtle or visually similar to non-violent actions [48,63,64]. Future work will also involve a comparative analysis of the Int.2D-3D-CNN model using alternative backbone architectures, such as

EfficientNet [65], to evaluate the trade-off between classification accuracy and computational efficiency. To further enhance recognition performance, upcoming experiments will incorporate video vision transformers (ViViT) [39] within the evaluation framework.

In terms of practical deployment, the proposed model shows potential for integration into real-time monitoring systems, smart city infrastructure, and in-vehicle safety platforms aimed at enhancing passenger protection [66]. These systems facilitate early threat detection and enable timely interventions, thereby helping to prevent violent incidents in both public and private environments. However, the use of AI-based monitoring technologies necessitates careful consideration of associated ethical implications. The adoption of automated violence detection models raises significant concerns related to privacy, informed consent, and the risk of misuse. To mitigate surveillance overreach and uphold civil liberties, it is imperative to ensure transparency, accountability, and strict adherence to legal and ethical standards. Future research should involve interdisciplinary collaboration to examine these concerns in depth and to support the responsible integration of AI technologies in public safety domains [67,68].

8. Conclusions

This research proposes an integrated 2D-CNN and 3D-CNN architecture, namely Int.2D-3D-CNN, aimed at improving video-based violence recognition. The model combines spatial and temporal analysis, combining lightweight MobileNetV1 and MobileNetV2 for extracting frame-level spatial features, with a streamlined 3D-CNN that includes a single 3D convolutional layer to capture motion and sequential patterns across frames. Experimental results on three benchmark datasets, namely Hockey Fight, Movie, and Violent Flows, indicate that the proposed architecture consistently outperforms conventional 2D-CNN- and C3D-based models. The Int.2D-3D-CNN model effectively capture both static and dynamic characteristics of violent scenes, achieving robust performance under challenging conditions, including distant viewpoints, occlusions, high crowd density, and low-light environments. The model achieved accuracies of 98%, 100%, and 98% on the Hockey Fight, Movie, and Violent Flows datasets, respectively, indicating strong generalization and competitive performance compared to state-of-the-art methods.

Author Contributions: Conceptualization, W.G., E.O., and O.S.; methodology, W.G. and O.S.; software, W.G.; validation, W.G., S.P., and E.O.; formal analysis, W.G. and O.S.; investigation, W.G. and S.P.; resources, W.G.; data curation, W.G.; writing—original draft preparation, W.G.; writing—review and editing, O.S.; visualization, W.G., S.P., and E.O.; supervision, O.S.; funding acquisition, O.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Mahasarakham University under grant number 6717011/2567. The APC was funded by Mahasarakham University.

Data Availability Statement: The original contributions presented in this study are included in the article. Further inquiries can be directed to the corresponding author.

Acknowledgments: This research project was financially supported by Mahasarakham University.

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

Abbreviations

The following abbreviations are used in this manuscript:

2D-CNN	Two-Dimensional Convolutional Neural Network
2MPD-3DFCN-AttBiDLSTM	Two-Stream Multi-Scale Patch-based Pyramidal Dilated 3D Fully Connected Network with Attentive Bidirectional Long Short-Term Memory

3D	Three-Dimensional
3D Conv	Three-Dimensional Convolution Layer
3D GAP	Three-Dimensional Global Average Pooling
3D-CNN	Three-Dimensional Convolutional Neural Network
3D-ResNet+ATDS	Three-Dimensional Residual Network with Adaptive Temporal Down-Sampling
Acc.	Accuracy
AI	Artificial Intelligence
AlexNet	Alex Network
AUC	Area Under the Curve
AUC-PR	Area Under the Precision–Recall Curve
BiConvLSTM	Bidirectional Convolutional LSTM
BiLSTM	Bidirectional LSTM
BN	Batch Normalization
BoW	Bag of Visual Words
C3D	Deep 3-Dimensional Convolutional Network
convLSTM	Convolutional Long Short-Term Memory
DL-STFEE	Double-Layer Spatial–Temporal Feature Extraction and Evaluation
FC	Fully Connected
FLOPS	Floating-Point Operations per Second
FLOPs	Number of Floating-Point Operations
FN	False Negative
FP	False Positive
FPR	False Positive Rate
HOG	Histogram of Oriented Gradients
hr	Hour
Int.2D-3D-CNNs	Integrated 2D and 3D Convolutional Neural Networks
KDE	Kernel Density Estimation
KTH	KTH Action Database
LSTM	Long Short-Term Memory
LV	Realistic Surveillance Video Dataset
MobileNet	Convolutional Neural Networks for Mobile Vision
MoSIFT	Motion SIFT
ms	Millisecond
OviF	Oriented Violent Flow
ResNet	Residual Network
ROC	Receiver Operating Characteristics
RWF-2000	Open Large Scale Video Database for Violence Detection
SGD	Stochastic Gradient Descent
SIFT	Scale-Invariant Feature Transform
STIPs	Space–Time Interest Points
SVM	Support Vector Machine
TN	True Negative
TNR	True Negative Rate
TP	True Positive
TPR	True Positive Rate
UCF Crime	University of Central Florida Crime Dataset
UCF-101	University of Central Florida-101 Dataset
UCSD	University of California, San Diego. Anomaly Detection Dataset
UMN	University of Minnesota. Abnormal Events Detection Dataset
VD-Net	Violence Detection Network
VGGNet	Visual Geometry Group Network
ViF	Violent Flow
ViViT	Video Vision Transformer

References

1. Yang, X.; Tang, S.; Guo, T.; Huang, K.; Xu, J. Design of indoor fall detection system for the elderly based on ZYNQ. In Proceedings of the IEEE 9th Joint International Information Technology and Artificial Intelligence Conference (ITAIC), Chongqing, China, 11–13 December 2020; Volume 9, pp. 1174–1178. [CrossRef]
2. Rajavel, R.; Ravichandran, S.K.; Harimoorthy, K.; Nagappan, P.; Gobichettipalayam, K.R. IoT-based smart healthcare video surveillance system using edge computing. *J. Ambient. Intell. Humaniz. Comput.* **2022**, *13*, 3195–3207. [CrossRef]
3. Romeo, L.; Marani, R.; D'Orazio, T.; Cicirelli, G. Video based mobility monitoring of elderly people using deep learning models. *IEEE Access* **2023**, *11*, 2804–2819. [CrossRef]
4. Şengönül, E.; Samet, R.; Al-Haija, Q.A.; Alqahtani, A.; Alturki, B.; Alsulami, A.A. An analysis of artificial intelligence techniques in surveillance video anomaly detection: A comprehensive survey. *Appl. Sci.* **2023**, *13*, 4956. [CrossRef]
5. Jahlan, H.M.B.; Elrefaei, L.A. Detecting violence in video based on deep features fusion technique. *arXiv* **2022**, arXiv:2204.07443. [CrossRef]
6. Kuppusamy, P.; Bharathi, V.C. Human abnormal behavior detection using CNNs in crowded and uncrowded surveillance—A survey. *Meas. Sens.* **2022**, *24*, 100510. [CrossRef]
7. de Souza, F.D.M.; Chávez, G.C.; do Valle, E.A., Jr.; Araújo, A.d.A. Violence detection in video using spatio-temporal features. In Proceedings of the 23rd Conference on Graphics, Patterns and Images (SIBGRAPI), Gramado, Brazil, 30 August–3 September 2010; pp. 224–230. [CrossRef]
8. Das, S.; Sarker, A.; Mahmud, T. Violence detection from videos using HOG features. In Proceedings of the 4th International Conference on Electrical Information and Communication Technology (EICT), Khulna, Bangladesh, 20–22 December 2019; pp. 1–5. [CrossRef]
9. Gao, Y.; Liu, H.; Sun, X.; Wang, C.; Liu, Y. Violence detection using oriented violent flows. *Image Vis. Comput.* **2016**, *48–49*, 37–41. [CrossRef]
10. Sarker, I.H. Deep learning: A comprehensive overview on techniques, taxonomy, applications and research directions. *SN Comput. Sci.* **2021**, *2*, 420. [CrossRef]
11. Khan, S.U.; Haq, I.U.; Rho, S.; Baik, S.W.; Lee, M.Y. Cover the violence: A novel deep-learning-based approach towards violence-detection in movies. *Appl. Sci.* **2019**, *9*, 4963. [CrossRef]
12. Carneiro, S.A.; da Silva, G.P.; Guimarães, S.J.F.; Pedrini, H. Fight detection in video sequences based on multi-stream convolutional neural networks. In Proceedings of the 32nd Conference on Graphics, Patterns and Images (SIBGRAPI), Rio de Janeiro, Brazil, 28–31 October 2019; pp. 8–15. [CrossRef]
13. Soliman, M.M.; Kamal, M.H.; Nashed, M.A.E.M.; Mostafa, Y.M.; Chawky, B.S.; Khattab, D. Violence recognition from videos using deep learning techniques. In Proceedings of the 9th International Conference on Intelligent Computing and Information Systems (ICICIS), Cairo, Egypt, 8–10 December 2019; pp. 80–85. [CrossRef]
14. Ullah, F.U.M.; Ullah, A.; Muhammad, K.; Haq, I.U.; Baik, S.W. Violence detection using spatiotemporal features with 3D convolutional neural network. *Sensors* **2019**, *19*, 2472. [CrossRef]
15. Li, C.; Zhu, L.; Zhu, D.; Chen, J.; Pan, Z.; Li, X.; Wang, B. End-to-end multiplayer violence detection based on deep 3D CNN. In Proceedings of the the VII International Conference on Network, Communication and Computing (ICNCC), Taipei, Taiwan, 14–16 December 2018; pp. 227–230. [CrossRef]
16. Li, J.; Jiang, X.; Sun, T.; Xu, K. Efficient violence detection using 3D convolutional neural networks. In Proceedings of the IEEE 16th International Conference on Advanced Video and Signal Based Surveillance (AVSS), Taipei, Taiwan, 18–21 September 2019; pp. 1–8. [CrossRef]
17. Dalal, N.; Triggs, B. Histograms of oriented gradients for human detection. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 886–893. [CrossRef]
18. Patil, C.M.; Jagadeesh, B.; Meghana, M.N. An approach of understanding human activity recognition and detection for video surveillance using HOG descriptor and SVM classifier. In Proceedings of the International Conference on Current Trends in Computer, Electrical, Electronics and Communication (CTCEEC), Mysore, India, 8–9 September 2017; pp. 481–485. [CrossRef]
19. Nievas, E.B.; Suarez, O.D.; García, G.B.; Sukthankar, R. Violence detection in video using computer vision techniques. In Proceedings of the Computer Analysis of Images and Patterns (CAIP), Seville, Spain, 29–31 August 2011; Volume 6855, pp. 332–339. [CrossRef]
20. Xu, L.; Gong, C.; Yang, J.; Wu, Q.; Yao, L. Violent video detection based on MoSIFT feature and sparse coding. In Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Florence, Italy, 4–9 May 2014; pp. 3538–3542. [CrossRef]
21. Hassner, T.; Itcher, Y.; Kliper-Gross, O. Violent flows: Real-time detection of violent crowd behavior. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Providence, RI, USA, 16–21 June 2012; pp. 1–6. [CrossRef]

22. Zhou, P.; Ding, Q.; Luo, H.; Hou, X. Violent interaction detection in video based on deep learning. *J. Phys. Conf. Ser.* **2017**, *844*, 012044. [CrossRef]
23. Tyagi, B.; Nigam, S.; Singh, R. A review of deep learning techniques for crowd behavior analysis. *Arch. Comput. Methods Eng.* **2022**, *29*, 5427–5455. [CrossRef]
24. Humeau-Heurtier, A. Texture feature extraction methods: A survey. *IEEE Access* **2019**, *7*, 8975–9000. [CrossRef]
25. Pu, S.; Chu, L.; Hou, Z.; Hu, J.; Huang, Y.; Zhang, Y. Spatial-temporal feature extraction and evaluation network for citywide traffic condition prediction. *IEEE Trans. Intell. Veh.* **2023**, *9*, 5377–5391. [CrossRef]
26. Sudhakaran, S.; Lanz, O. Learning to detect violent videos using convolutional long short-term memory. In Proceedings of the 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), Lecce, Italy, 29 August–1 September 2017; pp. 1–6. [CrossRef]
27. Sumon, S.A.; Shahria, T.; Goni, R.; Hasan, N.; Almarufuzzaman, A.M.; Rahman, R.M. Violent Crowd Flow Detection Using Deep Learning. In Proceedings of the Intelligent Information and Database Systems, Yogyakarta, Indonesia, 8–11 April 2019; Nguyen, N.T., Gaol, F.L., Hong, T.P., Trawiński, B., Eds.; Springer: Cham, Switzerland, 2019; pp. 613–625. [CrossRef]
28. Naik, A.; Gopalakrishna, M.T. Deep-violence: Individual person violent activity detection in video. *Multimed. Tools Appl.* **2021**, *80*, 18365–18380. [CrossRef]
29. Hanson, A.; PNVR, K.; Krishnagopal, S.; Davis, L. Bidirectional convolutional LSTM for the detection of violence in videos. In Proceedings of the Computer Vision—ECCV 2018 Workshops, Munich, Germany, 8–14 September 2018; Leal-Taixé, L., Roth, S., Eds.; Springer: Cham, Switzerland, 2019; pp. 280–295. [CrossRef]
30. Vosta, S.; Yow, K.C. A CNN-RNN combined structure for real-world violence detection in surveillance cameras. *Appl. Sci.* **2022**, *12*, 1021. [CrossRef]
31. Getsopon, W.; Surinta, O. Fusion lightweight convolutional neural networks and sequence learning architectures for violence classification. *ICIC Express Lett. Part B Appl.* **2022**, *13*, 1027–1035. [CrossRef]
32. Chen, J.; Wang, J.; Yuan, Q.; Yang, Z. CNN-LSTM model for recognizing video-recorded actions performed in a traditional Chinese exercise. *IEEE J. Transl. Eng. Health Med.* **2023**, *11*, 351–359. [CrossRef]
33. Hu, Z.p.; Zhang, L.; Li, S.f.; Sun, D.g. Parallel spatial-temporal convolutional neural networks for anomaly detection and location in crowded scenes. *J. Vis. Commun. Image Represent.* **2020**, *67*, 102765. [CrossRef]
34. Kokila, M.L.S.; Christopher, V.B.; Sajan, R.I.; Akhila, T.S.; Kavitha, M.J. Efficient abnormality detection using patch-based 3D convolution with recurrent model. *Mach. Vis. Appl.* **2023**, *34*, 54. [CrossRef]
35. Maqsood, R.; Bajwa, U.I.; Saleem, G.; Raza, R.H.; Anwar, M.W. Anomaly recognition from surveillance videos using 3D convolution neural network. *Multimed. Tools Appl.* **2021**, *80*, 18693–18716. [CrossRef]
36. Pratama, R.A.; Yudistira, N.; Bachtiar, F.A. Violence recognition on videos using two-stream 3D CNN with custom spatiotemporal crop. *Multimed. Tools Appl.* **2023**, *83*, 61995–62017. [CrossRef]
37. Keceli, A.S.; Kaya, A. Violent activity classification with transferred deep features and 3D-CNN. *Signal Image Video Process.* **2023**, *17*, 139–146. [CrossRef]
38. Bertasius, G.; Wang, H.; Torresani, L. Is space-time attention all you need for video understanding? In *Machine Learning Research, Proceedings of the 38th International Conference on Machine Learning, Virtual, 18–24 July 2021*; Meila, M., Zhang, T., Eds.; PMLR: Birmingham, UK, 2021; Volume 139, pp. 813–824.
39. Arnab, A.; Dehghani, M.; Heigold, G.; Sun, C.; Lucic, M.; Schmid, C. ViViT: A video vision transformer. *CoRR* **2021**, arXiv:2103.15691.
40. Liu, Z.; Ning, J.; Cao, Y.; Wei, Y.; Zhang, Z.; Lin, S.; Hu, H. Video Swin transformer. In Proceedings of the 2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 18–24 June 2022; IEEE: New York, NY, USA, 2022; pp. 3192–3201. [CrossRef]
41. Singh, S.; Dewangan, S.; Krishna, G.S.; Tyagi, V.; Reddy, S.; Medi, P.R. Video vision transformers for violence detection. *arXiv* **2022**, arXiv:2209.03561. [CrossRef]
42. Fish, E.; Weinbren, J.; Gilbert, A. Two-stream transformer architecture for long video understanding. *arXiv* **2022**, arXiv:2208.01753. [CrossRef]
43. Selva, J.; Johansen, A.S.; Escalera, S.; Nasrollahi, K.; Moeslund, T.B.; Clapés, A. Video transformers: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12922–12943. [CrossRef]
44. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. MobileNets: Efficient convolutional neural networks for mobile vision applications. *arXiv* **2017**, arXiv:1704.04861. [CrossRef]

45. Guo, Y.; Li, Y.; Wang, L.; Rosing, T. Depthwise convolution is all you need for learning multiple visual domains. In Proceedings of the the 33rd AAAI Conference on Artificial Intelligence (AAAI-19), AAAI'19/IAAI'19/EAAI'19, Honolulu, HI, USA, 27 January–1 February 2019; AAAI Press: Washington, DC, USA, 2019; Volume 33, pp. 8368–8375. [CrossRef]
46. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted residuals and linear bottlenecks. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVRP), Salt Lake City, UT, USA, 18–23 June 2018; pp. 4510–4520. [CrossRef]
47. Tran, D.; Bourdev, L.D.; Fergus, R.; Torresani, L.; Paluri, M. Learning spatiotemporal features with 3D convolutional networks. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 4489–4497. [CrossRef]
48. Cerar, G.; Bertalanič, B.; Fortuna, C. Resource-aware deep learning for wireless fingerprinting localization. In *Machine Learning for Indoor Localization and Navigation*; Tiku, S., Pasricha, S., Eds.; Springer: Cham, Switzerland, 2023; pp. 473–490. [CrossRef]
49. Nahm, F.S. Receiver operating characteristic curve: Overview and practical use for clinicians. *Korean J. Anesthesiol.* **2022**, *75*, 25–36. [CrossRef]
50. Alguliyev, R.; Aliguliyev, R.; Sukhostat, L. Radon transform based malware classification in cyber-physical system using deep learning. *Results Control Optim.* **2024**, *14*, 100382. [CrossRef]
51. Davis, J.; Goadrich, M. The relationship between precision-recall and ROC curves. In Proceedings of the the 23rd International Conference on Machine Learning (ICML), Pittsburgh, PA, USA, 25–29 June 2006; pp. 233–240. [CrossRef]
52. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
53. Almazroey, A.A.; Jarraya, S.K. Fight detection in crowd scenes based on deep spatiotemporal features. In Proceedings of the 2nd International Conference on Artificial Intelligence, Robotics and Control (AIRC), Cairo, Egypt, 12–14 December 2020; ACM: New York, NY, USA, 2020; pp. 18–23. [CrossRef]
54. Ullah, F.U.M.; Muhammad, K.; Haq, I.U.; Khan, N.; Heidari, A.A.; Baik, S.W.; de Albuquerque, V.H.C. AI-assisted edge vision for violence detection in IoT-based industrial surveillance networks. *IEEE Trans. Ind. Inform.* **2022**, *18*, 5359–5370. [CrossRef]
55. Mahmoodi, J.; Nezamabadi-pour, H.; Abbasi-Moghadam, D. Violence detection in videos using interest frame extraction and 3D convolutional neural network. *Multimed. Tools Appl.* **2022**, *81*, 20945–20961. [CrossRef]
56. Trinh, T.D.; Vu-Ngoc, T.S.; Le-Nhi, L.T.; Le, D.D.; Nguyen, T.B.; Pham, T.B. Violence detection in videos based on CNN feature for ConvLSTM2D. In Proceedings of the The Fifth Workshop on Intelligent Cross-Data Analysis and Retrieval, ICMR '24, Phuket, Thailand, 10–14 June 2024; ACM: New York, NY, USA, 2024; pp. 33–36. [CrossRef]
57. Mahmoodi, J.; Nezamabadi-pour, H.; Mirzaei, B. Improved violence detection in video analysis with a hybrid CNN approach using 3D and 2D convolutional networks. In Proceedings of the 2024 19th Iranian Conference on Intelligent Systems (ICIS), Sirjan, Iran, 23–24 October 2024; IEEE: New York, NY, USA, 2024; pp. 30–35. [CrossRef]
58. Zhang, X.; Liu, Z.; Wang, Q. Violence recognition with adaptive temporal down-sampling. In Proceedings of the 2025 8th International Conference on Advanced Algorithms and Control Engineering (ICAACE), Shanghai, China, 21–23 March 2025; IEEE: New York, NY, USA, 2025; pp. 1551–1560. [CrossRef]
59. Sultani, W.; Chen, C.; Shah, M. Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 6479–6488. [CrossRef]
60. Cheng, M.; Cai, K.; Li, M. RWF-2000: An open large scale video database for violence detection. *CoRR* **2019**, arXiv:1911.05913.
61. Huillcen Baca, H.A.; Palomino Valdivia, F.d.L.; Gutierrez Caceres, J.C. Efficient human violence recognition for surveillance in real time. *Sensors* **2024**, *24*, 668. [CrossRef] [PubMed]
62. Cai, X.; Yan, Z.; Duan, F.; Hu, D.; Zhang, J. Lightweight convolution neural network based on feature concatenate for facial expression recognition. In Proceedings of the Intelligent Computing in Engineering, Hanoi, Vietnam, 8–9 August 2019; Solanki, V.K., Hoang, M.K., Lu, Z.J., Pattnaik, P.K., Eds.; Springer: Singapore, 2020; Volume 1125, pp. 1141–1148. [CrossRef]
63. Wang, C.Y.; Mark Liao, H.Y.; Wu, Y.H.; Chen, P.Y.; Hsieh, J.W.; Yeh, I.H. CSPNet: A new backbone that can enhance learning capability of CNN. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Seattle, WA, USA, 14–19 June 2020; pp. 1571–1580. [CrossRef]
64. Li, K.; Wang, Y.; Zhang, J.; Gao, P.; Song, G.; Liu, Y.; Li, H.; Qiao, Y. UniFormer: Unifying convolution and self-attention for visual recognition. *IEEE Trans. Pattern Anal. Mach. Intell.* **2023**, *45*, 12581–12600. [CrossRef]
65. Tan, M.; Le, Q.V. EfficientNet: Rethinking model scaling for convolutional neural networks. *CoRR* **2019**, arXiv:1905.11946.
66. Kumar, P.; Shih, G.L.; Guo, B.L.; Nagi, S.K.; Manie, Y.C.; Yao, C.K.; Arockiyadoss, M.A.; Peng, P.C. Enhancing smart city safety and utilizing AI expert systems for violence detection. *Future Internet* **2024**, *16*, 50. [CrossRef]

67. Phillips, C.; Jiao, J. Artificial intelligence & smart city ethics: A systematic review. In Proceedings of the 2023 IEEE International Symposium on Ethics in Engineering, Science, and Technology (ETHICS), West Lafayette, IN, USA, 18–20 May 2023; pp. 1–5. [CrossRef]
68. Monika, E.; Rajesh Kumar, T. Advancements in AI-based crime detection and prediction. In Proceedings of the 2024 8th International Conference on Electronics, Communication and Aerospace Technology (ICECA), Coimbatore, India, 6–8 November 2024; pp. 1557–1562. [CrossRef]