IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 17 January 2023, accepted 11 February 2023, date of publication 14 February 2023, date of current version 27 February 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3245521

RESEARCH ARTICLE

Toward Fast and Accurate Violence Detection for Automated Video Surveillance Applications

VIKTOR DÉNES HUSZÁR[1], **VAMSI KIRAN ADHIKARLA**[2], **(Member, IEEE), IMRE NÉGYESI**[1], **AND CSABA KRASZNAY**[1]

[1]Faculty of Military Science and Officer Training, National University of Public Service, 1083 Budapest, Hungary
[2]Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, 1083 Budapest, Hungary

Corresponding author: Vamsi Kiran Adhikarla (adhikarla.vamsi.kiran@itk.ppke.hu)

This work was supported by the National Research, Development, and Innovation Office under Grant TKP-2021_02-NVA-27.

ABSTRACT Surveillance cameras are increasingly being used worldwide due to the proliferation of digital video capturing, storage, and processing technologies. However, the large volume of video data generated makes it difficult for humans to perform real-time analysis, and even manual approaches can result in delayed detection of events. Automatic violence detection in surveillance footage has therefore gained significant attention in the scientific community as a way to address this challenge. With the advancement of machine learning algorithms, automatic video recognition tasks such as violence detection have become increasingly feasible. In this study, we investigate the use of smart networks that model the dynamic relationships between actors and/or objects using 3D convolutions to capture both the spatial and temporal structure of the data. We also leverage the knowledge learned by a pre-trained action recognition model for efficient and accurate violence detection in surveillance footage. We extend and evaluate several public datasets featuring diverse and challenging video content to assess the effectiveness of our proposed methods. Our results show that our approach outperforms state-of-the-art methods, achieving approximately a 2% improvement in accuracy with fewer model parameters. Additionally, our experiments demonstrate the robustness of our approach under common compression artifacts encountered in remote server processing applications.

INDEX TERMS Anomaly detection, anomaly localization, automated video surveillance, deep learning, efficient violence detection, human activity recognition, security, smart cities, video recognition, violence detection.

I. INTRODUCTION

Today, surveillance and security cameras are deployed in various public places to monitor public events and human activity. Video surveillance improves public safety and plays a crucial preventive role in protecting a specific territory against crimes. The recorded surveillance footage is often used as evidence in criminal prosecutions. To prevent crime and reduce the crime rate, detecting and recognizing anomalies such as violence as soon as possible is a crucial task for the military and law enforcement agencies. However, surveillance cameras generate a large amount of video data every single day and instances of violence occur very rarely compared to other normal activities. Therefore, it is impractical and cumbersome for humans to manually monitor this video data for instances of violence. Human error may also reduce the efficiency of a manual, labour-intensive approach. Therefore there is a significant need for automatic and efficient methods for detecting abnormal or violent activities, especially in surveillance videos.

Video classification using Human Activity Recognition (HAR) is a popular research topic in recent years and is analogous to the field of violence detection. In these methods, sensor data is used to provide information on simple or complex physical activities of humans, such as standing, talking and cooking. Earlier techniques for HAR involved detecting and tracking human body parts in consecutive video frames using image-level descriptors, such as Histogram

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

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/

of Oriented Gradients (HOG) or Histogram of Oriented optical Flow (HOF) [1]. Other advanced approaches involved computing spatio-temporal descriptors for motion [2], [3]. However, one of the major drawbacks of these techniques is that they often require good lighting conditions and clear visibility for successful operation. With the development of depth cameras, algorithms have emerged that use depth measurements from sensors such as Microsoft Kinect [4], [5], ASUS Xtion2 [6] or Intel RealSense [7] for HAR. One advantage of depth sensors is that they come with Software Development Kit (SDK) containing real-time algorithms for detecting skeletons [8]. Specifically, a skeleton joint coordinate can be obtained in three dimensions (3D) in real-time and series of these coordinates, when tracked over time, can be used to detect and describe human actions. As a result, several algorithms have been proposed in the literature for using depth sensors to perform HAR [9], [10], [11], [12], [13] or using a combination of color and depth sensors [14]. However, depth sensors, even the modern ones, often have substantial noise in their measurements. Without adequately filtering out this noise, it can be difficult to achieve good detection for HAR. Additionally, integrating depth sensors into use cases such as surveillance can increase the hardware costs and may not always be feasible.

The use of Convolutional Neural Networks (CNNs) has become increasingly common in computer vision due to their exceptional success in image recognition tasks [15], [16]. CNNs are evolving rapidly in many fields of research, and it is expected that future solutions will enhance the adoption of CNNs. With the availability of big data and the exponential growth of computing power, these learning algorithms continue to have significant development potential. Several successful methods have recently been proposed that extend the spatial CNNs, which are used for image recognition tasks, to the temporal domain for HAR in videos [17], [18], [19], [20] [21], [22], [23], [24]. One of the main advantages of using CNNs for HAR is that they can handle challenging cases such as changes in lighting conditions, background changes, camera movement, different dressing styles and varying body shapes of people. They can also handle videos with partially or completely occluded human body parts.

In this paper, we address the problem of violence detection using deep learning with CNNs. Specifically, the following are the contributions of this work:

1) We propose a deep learning-based approach for filtering videos based on their violent or normal content. Our method is computationally efficient, making it practical for real-world applications and performs better on popular video classification metrics than several state-of-the-art methods for violence detection. Additionally, our method is able to maintain high classification accuracy even in the presence of video compression artifacts.
2) We present a comprehensive video database for the study of violence, comprising both violent and normal videos. Our database combines and extends seven existing video databases, providing a diverse range of violent content in various contexts.
3) We present a fully functional stand-alone system that implements the proposed methods for automated violence detection.

The rest of the paper is organized as follows: In section II, we discuss related work. Section III describes the proposed approaches in detail. In section IV, we present the results and discuss the scope of generalization of our approach. In section V, we conclude the work and derive future directions for our current work.

II. RELATED WORK

In this section, we describe several classes of algorithms that have been proposed in the literature for detecting violence using deep learning. We note that in the literature, there are multiple variants of violence detection that are being studied under different names, such as anomaly detection, abnormal activity detection and fight detection. Our current work focuses on forms of violence that primarily involve humans and human interaction with objects.

Due to the lack of a substantial amount of labeled data containing diverse real-world violence samples, several studies in the literature have used training data containing only a few samples. There are also some large-scale, publicly available datasets for violence detection. However, for these datasets, the exact time and duration of the violence are not available. Algorithms trained on such data often strive to minimize unusual patterns among training samples in order to learn about rare violent activities [25], [26] [27], [28] [29]. These methods are described in the subsections II-A and II-B. We also introduce methods that use labeled training data for violence detection in subsection II-C.

A. MODELLING NORMAL PATTERNS

These techniques learn patterns of normal behavior from training videos that contain no violence. Since only normal videos without violence are used in the training phase, no specific labels are provided. During testing, these methods are expected to find samples that deviate from the learned normal behavior [30]. In [31], and [25], motion trajectories are used to learn about normal patterns. In [31], the authors suggested representing motion patterns using super-trajectories that describe motion of local groups of similarly moving points (pixels in a video sequence) and clustering these motion patterns hierarchically to derive prototype patterns for normal samples.

In [32] and [33], the authors used auto-encoders to learn regularities in video sequences. As inputs to the auto-encoders, they used state-of-the-art spatio-temporal motion features computed using HOG and HOF [34]. In [35], the authors used optical flows along with video sequences and constructed multiple auto-encoders. They used reconstruction loss [36] to detect abnormal or violent events.

Authors in [37], [38], and [39] also incorporated auto-encoders to learn normal behaviors, but without explicitly computing local motion patterns. This one-stage approach is faster in terms of computational speed because it does not require object detection or feature extraction. There were also approaches that augmented memory modules [37] to auto-encoders and used optical flow images [40], [41] to define flows of normal patterns. In [37], the authors augmented the output of an encoder in a variation of an auto-encoder CNN with a memory module that adaptively records prototypical patterns of normal data for more accurate detection of violent cases in a given database.

In [39] and [42], the authors employed a variation of auto-encoders to predict a future video frame from a given set of consecutive video frames. Then, they computed per-pixel differences between the predicted and ground truth frames to make a decision on whether the current video sequence is normal or not. Future frame prediction has gained increasing attention due to its potential applications in unsupervised feature learning for video representation [43]. In [39], the authors also quantized the output of the encoder using a predefined codebook (a concept similar to augmented memory modules) that further narrows the explanation of normal events and aids in better future frame prediction in normal videos. In [44], to generate more realistic and accurate future frames, the authors imposed a loss in the temporal space. In particular, they computed optical flows in video sequences using a pre-trained CNN [45] and formulated a loss function for an auto-encoder that ensures the optical flow of predicted frames is consistent with the ground truth. In addition to the methods that predict future frames, there were also efforts in the literature to predict transformations needed for generating future frames [46], [47], [48], [49].

B. MULTIPLE INSTANCE LEARNING

These methods also aim to learn about violent actions using video-level labels that are provided during the training phase. In contrast to methods that model normal patterns, these methods use both normal and violent data to train violent detection models [50], [51], [52], typically using Multiple Instance Learning (MIL) [53]. Sultani et al. [50] divided each video (in both normal and violent videos) into multiple temporal segments to form positive and negative bags that capture instances of the violent and normal events respectively. C3D [54] spatio-temporal features were then extracted from each segment and used to train multiple fully connected layers, which derive scores for the positive and negative bags. Due to the absence of segment-level labels, a novel ranking loss function was proposed that encourages the score indicating violence in the positive bag to be higher than the score in the negative bag. The ranking loss also imposed smoothness and sparsity constraints in the ranking loss to reduce false alarms.

By extending the approach of Sultani et al., Zhu et al. in [51] introduced temporal context information into the MIL ranking loss to compute video-wise scores, rather than segment-wise scores. They proposed a temporal augmented network that captures motion features using pre-computed optical flows, similar to an auto-encoder. The encoded motion patterns were used to train MIL ranking model for better localization of violence instances.

Philippe et al. [55] proposed a two-step approach where they first detect and track humans locally across a given segment of a video to form human tubes (spanning the entire segment) and then use multi-fold Multiple Instance Learning (MIL) with Support Vector Machines (SVM) [56] to learn about human tubes that contain the action described by the video-level labels. In [57], Yan et al. proposed a multi-task ranking model. In their approach, they segmented videos into supervoxels using a graph-based segmentation method to generate action tubes and action–actor tubes. Action tubes were then used as proposals for actions, e.g., walking, adult running, and crawling. Features were extracted from each tube to train the ranking model to select the most characteristic action tubes.

Arnab et al. [58] proposed a probabilistic variant of MIL, in which they estimate the uncertainty of an instance-level prediction. They used a pre-trained person detector trained on a large image dataset to detect persons over consecutive frames of a video to form person tubelets. A bag for MIL consists of all tubelets within a video, and it is annotated with the video-level label. During training, they also model the label noise through the uncertainty of sampling bags that do not contain any tubelets with the labeled action.

Mettes et al. in [59] aimed to find the spatio-temporal locations of actions in videos using pseudo-annotations. They investigated spatio-temporal pseudo-annotations from different sources such as action proposals, object proposals, person detection, motion, and center biases. They later combined the extracted pseudo-annotations using a correlation metric to train a classifier using MIL.

C. SUPERVISED LEARNING

There have been multiple approaches that use deep learning to classify violent videos using labeled data. These methods rely on video datasets with accurate visual information about the relevant class, such as videos in the violence class containing few or no normal events.

In [60], Long et al. proposed a method for classifying violent videos using the Motion SIFT (MoSIFT) algorithm to extract features and then applying Kernel Density Estimation (KDE) to filter out noise. These reduced MoSIFT features were then transformed into a video-level feature vector using sparse coding, and a Support Vector Machine (SVM) was trained on these vectors to classify videos.

In 2012, Hassner et al [61] proposed a method for real-time detection of violence in crowded scenes using the Violent Flows (ViF) descriptor to capture optical flow information between consecutive video frames and a linear SVM to classify the videos based on the computed ViF descriptors. They demonstrated that their method was effective at classifying videos containing crowd violence, and it was compared to other existing methods at the time. In a later study, Meng and Serrano [62] proposed a method for violence detection that combined feature extraction with deep learning using Convolutional Neural Networks (CNNs). Their approach involved using a Hough Forests spatio-temporal feature extractor in combination with a 2D CNN.

Sudhakaran and Lanz [63] proposed a method for encoding the difference between two successive frames using a combination of a CNN and a Long Short-Term Memory (LSTM) module and demonstrated that this approach had better performance than a model trained on raw frames. AlDahoul et al. [64] proposed a lightweight model with fewer parameters that used a CNN and an LSTM module to capture spatial features for violent video classification. Fath U Min Ullah et al. [65] proposed a Violence Detection Network (VD-Net) that first used object detection to detect humans and suspicious objects like guns to pre-filter video sequences for violence detection and then applied a combination of a convolutional LSTM and gated recurrent units [66] to the filtered video sequences for violence detection. Romas et al. [67] also proposed a CNN-LSTM-like architecture that was computationally light, using MobileNet V2 to extract spatial features for training an LSTM network.

Chollet et al. [68] used a model based on XceptionNet [69] to extract features from a video and then applied a bi-directional LSTM to analyze the extracted features in both forward and backward temporal directions for classification. Khan et al. [70] proposed a method that uniformly samples a video into segments, selects a representative frame from each segment using computed levels of saliency, and then fine-tunes a MobileNet model [71] using the representative frames to classify the corresponding segment as violent or non-violent. Li et al. [72] proposed a DenseNet-based [73] 3D CNN architecture that directly processes video data without explicitly computing features, and demonstrated good accuracy on standard databases with a relatively lightweight model. Fernando et al. [74] proposed an architecture based on a variant of DenseNet [75] that extracts feature maps and then applies self-attention mechanisms [76] to link different positions in a single sequence and generate a representation that focuses on the most relevant parts of the sequence. This representation is fed into bi-directional LSTM blocks and fully connected layers for classification. They demonstrated good accuracy on four different databases using this method, and also experimented with using both optical flow and pseudo-optical flow computed from adjacent frames as inputs to the DenseNet.

In a 2016 study, Dong et al. proposed a multi-stream deep convolutional neural network consisting of three streams (color, optical flow, and person-to-person acceleration) for violence detection. The acceleration stream aimed to capture the intense information that was hypothesized to be present in violent events, and three LSTMs were trained using the features from the three streams. The outputs from the streams were fused to classify a video. In a later study, Su et al. [77] proposed a method for violence detection that involved computing 3D skeleton point clouds from video and then using interaction learning on these point clouds to capture spatio-temporal features and model interactions between skeleton points. They used multiple Skeleton Points Interaction Learning (SPIL) modules together with a fully connected layer to classify violent videos from normal videos. In another study, Mu et al. [78] proposed a method for violence detection that used both visual and audio cues, as it was hypothesized that visual information may not be reliable for violence detection and that using audio could improve performance. They extracted audio features using 40-dimensional Mel Filter-Bank (MFB) coefficients and used an SVM to classify audio samples from input videos.

D. COMPARISON OF THE STATE-OF-THE-ART METHODS

Table 1 compares different approaches for violence detection in videos proposed in the literature and lists their advantages and disadvantages. Successful algorithms for violence detection should be computationally fast, achieve high classification accuracy, and be adaptable to scenarios not present in the training data. Some normal actions involving close physical interaction between humans can mimic violent actions and can mislead the deep learning algorithms that are solely trained on normal videos. It is suggested that it is important to incorporate both normal and violent behaviors in the training data for better generalizability of the trained models. MIL using spatio-temporal feature-based methods can be computationally fast but may not achieve high classification accuracy, as they focus on predicting bag-level labels while neglecting the hidden temporal context information in violence and normal patterns.

From the results presented in the literature, it is evident that methods that use 3D deep learning architectures that capture spatio-temporal features in the data account for both the spatial structure of the video frames as well as the temporal dynamics between frames. This makes such 3D CNNs effective at tasks such as action recognition and violence detection, where the actions being performed and their temporal evolution are important factors to consider. However, it is important to note that the cost of extracting some of the spatio-temporal features is still prohibitive for practical applications. In the current work, a computationally light and accurate 3D deep learning architecture (see section III) is adapted and extended and labeled datasets are used (refer to section III-A) to develop efficient methods for violence detection.

III. FAST AND ACCURATE VIOLENCE DETECTION

ResNet [79] is a popular base architecture for image and video recognition tasks, known for its effectiveness and state-of-the-art results on benchmarks like ImageNet [80] and COCO [81] datasets. 3D ResNets [23] are an extension of the ResNet architecture, designed for learning spatiotemporal features from video data. They have achieved strong performance on various benchmarks and real-world applications, including the Kinetics-700 action recognition dataset [82] (where a variant called I3D [83] achieved state-of-the-art performance) and the Something-Something V2 action

TABLE 1. Comparison of approaches in the state-of-the-art for violence detection in videos.

Method	Strength(s)	Limitation(s)
Learning normal patterns	Computationally light	Poor generalizability and may not be suitable for practical applications.
Future frame prediction	Computationally light	Poor generalizability, difficulty interpreting normal actions, sensitivity to objects not present during training, memory modules in some architectures may restrict the interpretation of actions.
MIL using spatio-temporal features	Depending on features, can be computationally light	Trims videos into bags and focuses on predicting bag-level labels, ignoring temporal context information between bags, may not achieve high classification accuracy.
3D deep learning architectures	Good classification accuracy, adaptable to new scenarios	Extraction of 3D volumes can be memory-intensive, computing optical flow in some architectures can be time-consuming, training can be slow, computing some features can be expensive and may not be suitable for real-time applications.

recognition dataset (where a 3D ResNet called R(2+1)D [84] achieved state-of-the-art performance).

3D ResNets have higher accuracy than counterparts like 3D-MobileNet [85] due to factors such as more layers for learning complex spatio-temporal features and skip connections between the input and output of each layer that allow input to bypass intermediate layers. However, they are generally more computationally intensive due to a large number of model parameters. To improve computational efficiency, model complexity can be reduced through techniques such as reducing the number of layers, using fewer filters in convolutional layers, and using smaller input data, though this may decrease accuracy on complex tasks. Christoph et al. [86] experimented with various parameters of the 3D ResNet architecture to understand the effect of reduced model complexity on accuracy. They expanded the architecture along multiple axes to form spatio-temporal models and selected the axis that achieved the best trade-off between computational speed and accuracy, resulting in a series of models ranging from extra small (XS) to extra large (XL) in increasing complexity. Using the Kinetics-400 dataset [87], they showed that their expanded model, X3D-M, had the same accuracy as state-of-the-art video classification networks but with a 10X reduction in model parameters.

The X3D-M model is an appropriate choice for our violence detection task due to its high accuracy and reduced model complexity. As demonstrated by Christoph et al., the X3D-M model achieves similar accuracy to state-of-the-art video classification networks, but with a significantly lower parameter count. This reduction in model complexity makes the X3D-M model more efficient to train and deploy, particularly for resource-constrained systems. Also, the ResNet 3D backbone, which has a proven ability to learn complex spatio-temporal features, is particularly useful for our violence detection task, as it allows the model to capture the dynamic nature of the videos and learn robust representations of the data. The proposed system using X3D-M model architecture is detailed in section III-B.

A. DATASETS FOR EXPERIMENTS

Due to data protection laws such as GDPR [91], it is not possible to obtain large amounts of real-world footage containing violence for training deep learning models. Recently, the usage of synthetic training data has become more common in computer vision. The use of training data containing pasted object patches on real images has been shown to be effective for tasks such as 2D object detection [92], [93], [94] and human pose estimation [95]. However, for violence detection, we postulate that such fabricated training data may not fully capture the complex and diverse action patterns of violent actions with various nuances. Therefore, preparing and using synthetic training data is not considered in the scope of the current work.

In their study, P. Sernani et al. [96] proposed the AIRTLab dataset, which contains videos showing violence patterns performed by non-professional actors. They studied the use of 2D and 3D deep learning architectures for violence detection using their dataset and found that the studied models adapt well to their setting, where violence is mimicked by non-professional actors. However, they also noted that their results cannot be considered general, as their architectures were not validated on other datasets and no cross-validation experiments were performed. Therefore, we do not consider such datasets in our experiments.

In the current work, we have considered seven different datasets that are commonly used in the literature for experimentation with violence detection and to facilitate comparison of our results with other methods. We have also extended some of these datasets with annotations to assist in in-depth cross-validation experiments. These datasets are described in the following:

- **Crowd Violence** (CV) dataset [61] contains videos involving violence in crowds, collected from YouTube.
- **Hockey Fights** (HF) dataset [88] is a collection of fights between players in hockey games from the USA's National Hockey League (NHL).
- **Movie Fights** (MF) dataset [88] consists of a collection of scenes from action movies.

TABLE 2. Overview of the datasets used in our experiments. In addition to the existing databases in the literature (CV, HF, MF, RLVS, RWF-2K), we have annotated parts of two other datasets (UCFS and XD-V).

Dataset	Acronym	Total Samples	Avg. Frame Resolution(Pixels)	Avg. Video Duration(S) [min, max]	Avg. FPS [min, max]	Violent Samples	Normal Samples
Crowd Violence [61]	CV	246	320 × 240	3.49 [1.04, 6.44]	26.47 [25, 29.97]	123	123
Hockey Fights [88]	HF	1000	360 × 240	1.64 [1.6, 1.96]	25 [25, 25]	500	500
Movie Fights [88]	MF	200	320 × 240	1.74 [1.66, 2.04]	43.14 [25, 59.94]	100	100
Real Life Violence Situations [89]	RLVS	2000	variable	4.91 [1, 11.32]	27.21 [10.5, 37]	1000	1000
Real-World Fight-2000 [90]	RWF-2K	2000	variable	4.99 [0.32, 5]	29.99 [25, 30]	1000	1000
UCF-Crime Selected	UCFS	1082	320 × 240	4.25 [0.3, 5.03]	29.99 [25, 30]	541	541
XD-Violence Selected	XD-V	1126	640 × 360	4.15 [0.3, 5.04]	24 [24, 24]	563	563

- **Real Life Violence Situations** (RLVS) dataset [89] consists of fighting videos gathered from YouTube and real street cameras that depict real street fights.
- **Real-World Fight-2000** (RWF-2K) dataset [90] is a collection of large-scale fighting videos from YouTube. The dataset consists of trimmed video clips captured by surveillance cameras from real-world scenes.
- **UCF-Crime Selected** (UCFS) dataset is a subset of UCF-Crime dataset [50]. The UCF-Crime dataset consists of long untrimmed surveillance videos that cover 13 real-world anomalies, including Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fighting, Road Accident, Robbery, Shooting, Stealing, Shoplifting, and Vandalism, without annotations. While this is a large-scale dataset, videos in the violence class contain mix of violent and normal actions, which is undesirable. Therefore, we selected the classes of Abuse, Explosion, Fighting, Road Accident and Shooting from the UCF-Crime dataset and manually trimmed the videos to only contain violent parts for training and testing purposes.
- **XD-Violence Selected** (XD-V) dataset contains a subset of videos from the XD-Violence dataset [97]. The XD-Violence dataset consists of several untrimmed videos covering six anomalies, including Abuse, Car Accident, Explosion, Fighting, Riot, and Shooting, gathered from action movies and YouTube. Similar to the UCF-Crime dataset, we selected a set of videos belonging to the classes of Abuse, Explosion, Fighting, Road Accident and Shooting from the XD-Violence dataset and manually trimmed these videos to only contain violent parts for training and testing purposes.

All of the datasets also contain normal videos for training and testing that do not involve violence. In the case of the UCFS and XD-V datasets, we trimmed the normal videos to five-second video clips to match the average duration of normal clips in the other datasets. Additionally, in the case of the UCFS and XD-V datasets, we limited the maximum duration of a video clip containing violence to approximately five seconds. Table 2 provides more details about each of the datasets we used in our experiments.

B. MODEL ARCHITECTURE

We note that for accurate violence detection, it is important to a have properly labeled dataset containing a large number of diverse examples for training a deep learning model. Successful action recognition datasets such as Kinetics-400 [87], contain a minimum of 400 videos for each action class, such as standing, sitting and talking. All videos in the Kinectics-400 dataset have a fixed duration of five seconds. The authors obtained clips for each class from YouTube and then used Amazon Mechanical Turkers (AMT) to decide if a given clip contained the desired action. A clip was accepted if it received three or more confirmations (out of five) [87]. The dataset was also de-duplicated to reduce redundancies in the environment.

In several cases, actions involving violence are more complex than actions such as sitting and talking and the number of example violent videos collected in existing datasets may not be sufficient for training a model that generalizes well and can lead to overfitting. Additionally, as shown in Table 2, different datasets for violence detection contain clips with different durations in seconds and they are not well-organized to check for the validity of a specific action or for redundancies. To address these issues with existing datasets for violence detection, we follow training approaches that are inductive in nature. Specifically, we aim to make use of the knowledge learned using better-calibrated action recognition datasets to solve the efficient violence detection problem. To this end, we propose two different deep learning configurations that are described in the following subsections.

1) FINE-TUNED X3D-M MODEL

In the Fine-Tuned X3D-M (FT) model, we consider the X3D-M model architecture initialized with weights obtained by training on the Kinetics-400 dataset. Note that the original architecture used for training on the Kinetics-400 dataset contains two fully connected layers, with the output of the second fully connected layer representing the classification results for each class (the number of outputs of this layer is equal to the number of classes in the training dataset). Since we aim to predict if a clip contains violence or not (a binary classification), we modify the architecture into a regression model to generate a violence coefficient that indicates the probability of violence in a given video clip. Specifically, we trim the X3D-M model until the first fully connected layer and replace the second fully connected layer with one that outputs a floating-point variable, which is converted into the range of [0, 1] using a sigmoid function to derive the violence

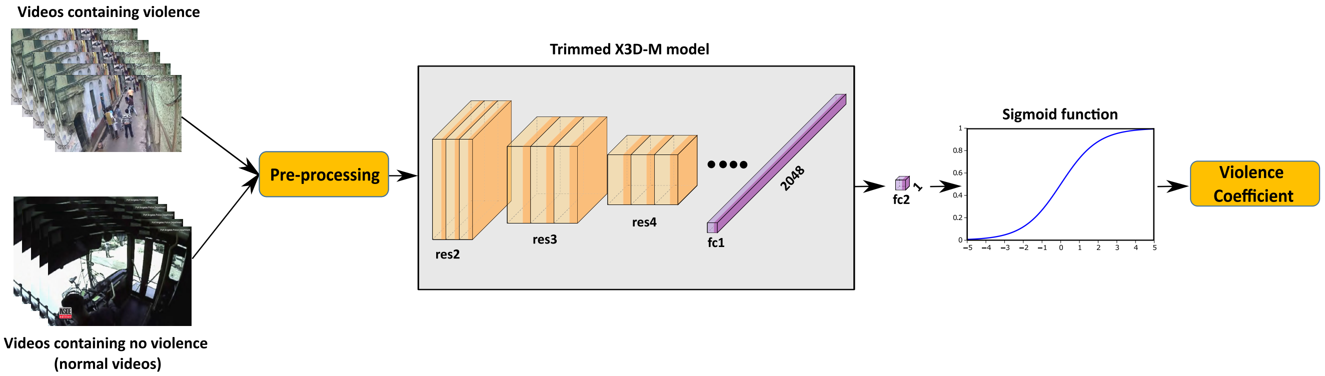


FIGURE 1. Our FT model: Batches of videos containing violence and non-violence are supplied for training. Each input video is pre-processed to obtain 16 uniformly sampled temporal frames for training a trimmed X3D-M model. The second fully connected layer of X3D-M model is replaced to output a floating-point variable, which is converted into range [0, 1] using a sigmoid function to derive the violence coefficient.

TABLE 3. In the FT model, we have 2976723 parameters that are involved in the combination of the trimmed X3D-M model and the replaced second fully connected layer. Since the parameters of the trimmed X3D-M model are also optimized during training, all 2976723 parameters are trainable.

Component	Output Shape	Params #
Trimmed X3D-M model	(2048)	2974674
Dense ("fc2" in Fig. 1)	(1)	2049
Total Parameters	2976723	

coefficient. Simply, during learning, we label the violence coefficient as 1 for samples of video clips containing violence and as 0 for samples of video clips containing no violence.

The architecture of the X3D-M model follows the fast pathway design of SlowFast networks [98] with down-sampled temporal input. Therefore, we pre-process the input videos as required by the X3D-M model. In particular, for a given video clip, we first extract 16 video frames by uniformly sampling in the temporal domain. Then, we transform the pixel value range of the extracted frames to be within [0, 1] to obtain floating-point images. Next, we normalize the video frames using mean and standard deviation and resize the frames so that the shortest side corresponds to 256 pixels. Finally, we center crop the resized frames to obtain 16 video frames with a spatial resolution of 256 × 256. Batches of pre-processed video frames are supplied to the FT model with corresponding labels for training. Note that the X3D-M model weights obtained by training on the Kinetics-400 dataset are only used for network initialization and these are further optimized during training on datasets for violence detection. The FT architecture is shown in Fig. 1, and Table 3 presents information on the corresponding model parameters.

2) TRANSFER-LEARNED X3D-M MODEL

Unlike the FT model, the Transfer-Learned X3D-M model (TL) uses the X3D-M model for feature extraction. Specifically, we provide pre-processed (following the method described in the FT model) batches of videos containing violence and non-violence as input to a trained X3D-M model that has been trained on the Kinetics-400 dataset and extract the output of the first fully connected layer to form a feature set. The extracted feature set is a vector containing 2048 elements, which is used to train three additional fully connected layers, as shown in Fig. 2. The output of the additional fully connected layers is a floating-point variable, and similar to the FT model, we transform this variable to be within the range of [0, 1] using a sigmoid function to obtain the violence coefficient. Table 4 presents information on the TL model parameters.

TABLE 4. In the TL model, we have 4040211 parameters. Since the parameters of the trimmed X3D-M model are not trained, 1065537 parameters are trainable and 2974674 parameters are non-trainable.

Component	Output Shape	Params #
Trimmed X3D-M model	(2048)	2974674
Dense ("fc2" in Fig. 2)	(512)	1049088
Dense ("fc3" in Fig. 2)	(32)	16416
Dense ("fc4" in Fig. 2)	(1)	33
Total Parameters	4040211	

C. LEARNING AND OPTIMIZATION

We do not apply data augmentation techniques in the training of the proposed models. We use Adagrad [99] to optimize our models with an initial learning rate of $1e^{-3}$. Both models are compiled to minimize the Binary Cross Entropy (BCE) between the estimated and ground truth violence coefficients. For training the TL model, we use a batch size of 30 samples collected from shuffled pre-computed X3D-M feature vectors. Since the FT model takes videos as input, to account for higher memory usage during training, we consider a batch size of 4 samples collected from shuffled videos. For regularity, within a training batch, for both models, we concatenate a batch of violent video clips with a batch of non-violent video clips. For ease of access, all our hyperparameters are listed in the table 5

IV. RESULTS AND DISCUSSION

In this section, we present the results from our various experiments using the proposed models and the various datasets described in section III-A. Most of the datasets used in the study already have a training and testing data split with 80% of the data as the training set and 20% as the

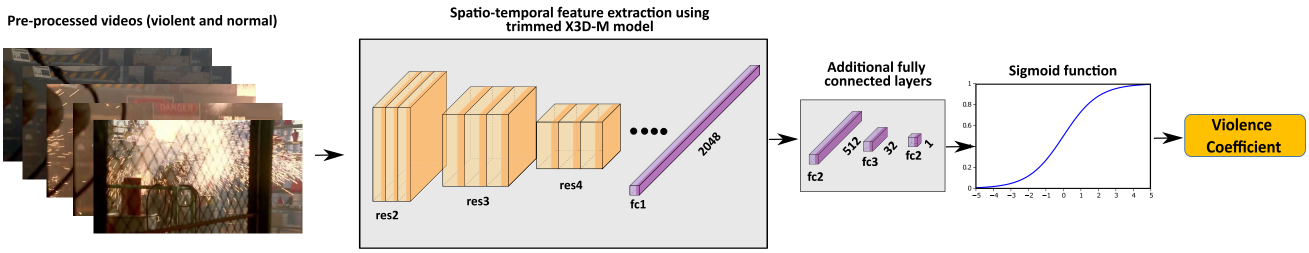

Pre-processed videos (violent and normal)
Spatio-temporal feature extraction using trimmed X3D-M model
res2
res3
res4
fc1
2048
Additional fully connected layers
fc2
512
fc3
32
fc2
1
Sigmoid function
1
0.8
0.6
0.2
0
−5
−4
−3
−2
−1
0
1
2
3
4
5
Violence Coefficient

FIGURE 2. Our TL model: Pre-processed videos containing both violence and non-violence are are input to a pre-trained X3D-M model for feature extraction. Three fully connected layers are trained using the extracted features to obtain the violence coefficient.

TABLE 5. List of hyperparameters and their corresponding values used in our trainings.

Hyperparameter	Value
Learning Rate	0.001
Batch Size	TL-30 & FT-4
Number of Epochs	50
Optimizer	Adagrad
Loss Function	Binary Cross Entropy loss
Dropout Rate	0.1
L2 Regularization	0.001

test set. For other datasets, for our experiments, we preserve this percentage and randomly select 20% of violent and non-violent samples to create a testing set for fair comparison across datasets. To facilitate fair comparison, all the models are trained for 50 epochs using a given training dataset. We use the PyTorch [100] deep learning library to train and test our models on a Nvidia GeForce GTX 1080 Ti GPU using the CUDA toolbox. We use the Ubuntu Linux operating system on an AMD Ryzen Threadripper 1950X 16-core processor. To evaluate the performance of various methods, we use the following metrics that are commonly used to evaluate the performance of classification algorithms using deep learning.

- **Accuracy** (ACC) [101] is the most popular metric for evaluating deep learning models for video classification. It is the ratio of the number of correct predictions (as violent or non-violent video clips) to the total number of predictions. To compute the accuracy, we used the provided ground truth binary labels - 0 (for video clips without violence) and 1 (for video clips with violence) - that are provided during training. Since we designed our networks to output floating-point violence coefficients, we round the predicted violence coefficients to the nearest integer before calculating the accuracy. In line with other methods in the literature, we report the accuracy score in percentages.
- **Area Under Curve** (AUC) [102] is a statistical measure to evaluate the performance of a classification model. It represents the area under the Receiver Operating Characteristic (ROC) curve, which graphically illustrates the effectiveness of a classifier in discriminating between the trained classes at various decision probability thresholds. Specifically, using the predicted violence coefficients, the ROC curve shows the relationship between True Positive Rate (TPR) (the number of times when violence cases are correctly identified as violence among the total cases when violence cases are correctly identified as violence and non-violence cases are correctly identified as non-violence) and False Positive Rate (FPR) (the number of times when non-violence cases are incorrectly identified as violence among the total cases when non-violence cases are incorrectly identified as violence and violence cases are incorrectly identified as non-violence). Higher values of the area under the ROC curve (that are close to 1) represent the ability of a model to effectively discern between violence and non-violence cases, while lower values represent the opposite.

We have conducted several experiments, including cross-dataset validation, to evaluate the performance of the proposed approaches using the considered datasets and metrics. The details and results of these experiments are presented in the following subsections.

A. EXPERIMENTS ON INDIVIDUAL DATASETS

Most datasets already have pre-defined data splits for training and testing, with 80% and 20% of the data respectively. We used these splits without modification for unbiased comparison. For the remaining datasets, we maintained this proportion of training and testing data by randomly selecting 20% of violent and non-violent samples for testing. We trained our models on the training data split and evaluated their performance on the testing data split for each dataset separately. The testing results using the ACC and AUC metrics are presented in Tables 6 & 7 respectively. The tables also show the performance of state-of-the-art methods discussed in section II on the respective datasetes. As mentioned, we created the UCFS and XD-V datasets and we report the results on these datasets using only our methods.

It is worth noticing that only a few studies in the literature report evaluations using the AUC metric. We argue that in applications such as violence detection, false positives (incorrectly reporting non-violent events as violent) should be explicitly considered when evaluating the performance of a model and the ACC metric does not directly account for false alarms.

The experimental results on individual datasets show that both of our proposed methods perform well on individual

TABLE 6. The ACC(%) scores of our FT and TL models along with the state-of-the-art methods on individual datasets. Based on the ACC metric, our FT method outperforms most of the state-of-the-art methods on all datasets except HF, with relatively fewer model parameters.

Method	CV	HF	MF	RLVS	RWF-2K	UCFS	XD-V	Params #
ViF [61]	81.3	82.9	-	-	-	-	-	-
3-stream+LSTM [103]	-	93.9	-	-	-	-	-	-
MoSIFT [60]	89	94	-	-	-	-	-	-
Bilinski [3]	96.4	96.8	99	-	-	-	-	-
Sudhakaran [63]	94.5	97.1	100	-	-	-	-	9.6M
Zihan Meng [62]	-	94.6	99	-	-	-	-	-
Li et al. [72]	97.17	98.3	100	-	-	-	-	7.4M
Akti 5-Frames [68]	-	95	90	-	-	-	-	9M
Akti 10-Frames [68]	-	96	87.5	-	-	-	-	9M
Khan et al. [70]	-	87	99.5	-	-	-	-	-
Pseudo-OF [74]	94.8	97.5	100	94.1	-	-	-	4.5M
OF [74]	96.9	**99.2**	100	95.6	-	-	-	4.5M
CNN-LSTM-IOT [64]	-	-	-	73.35	73.35	-	-	**1.266M**
Romas et al. [67]	-	-	99.5	73.35	82.3	-	-	4.074M
SPIL [77]	94.5	96.8	98.5	-	89.3	-	-	-
VD-Net [65]	-	98.5	-	-	88.2	-	-	4.4M
Choqueluque-Roman et al. [104]	-	97.3	-	92.88	88.71	-	-	>30M
ours (FT)	**99.5**	97.5	100	**96.7**	**91**	**90.1**	**93.59**	2.98M
ours (TL)	92	97.5	100	95.2	85	84.2	89.31	4.04M

TABLE 7. The AUC scores of our FT and TL models compared to various state-of-the-art methods. According to the AUC metric, the FT model outperforms the state-of-the-art methods on most of the datasets and has fewer model parameters.

Method	CV	HF	MF	RLVS	RWF-2K	UCFS	XD-V	Params #
MoSIFT [60]	0.935	0.96	-	-	-	-	-	-
Bilinski [3]	0.87	-	-	-	-	-	-	-
CNN-LSTM-IOT [64]	-	-	-	0.82	0.82	-	-	**1.266M**
VD-Net [65]	-	0.994	-	-	0.91	-	-	4.4M
Choqueluque-Roman et al. [104]	-	0.993	-	0.913	0.914	-	-	>30M
ours (FT)	**1.0**	**0.999**	**1.0**	**0.996**	**0.972**	**0.971**	**0.976**	2.98M
ours (TL)	0.99	0.994	**1.0**	0.993	0.944	0.936	0.959	4.04M

datasets. Overall, our FT model outperforms most of the state-of-the-art methods and our TL model also achieved decent performance on all datasets. We postulate that the FT model, which optimizes the parameters of the (trimmed) X3D-M model during learning, is more adaptable to a given dataset. On the MF dataset, the results for both TL and FT models suggest overfitting, which is consistent with the results from most methods in the literature. This suggests that the MF dataset may contain more regular examples with less diversity and may be less challenging for deep learning video classification models. The Tables 6 & 7 also show the model parameter count for various models under comparison and our models have fewer parameters than the state-of-the-art methods.

Bilinski et al. [3] achieved a higher accuracy than our TL model on the CV dataset. They used improved Fisher vectors for spatio-temporal feature extraction, which can be context-dependent. For example, the CV dataset only contains examples of violence involving a crowd and their results show that their method performs better in such scenarios. It is important to note that statistical feature extraction methods like this can be sensitive to variations in the video capture environment and may result in false alarms. When evaluated using the AUC metric, our TL model performs better than the method of Bilinski et al. [3] on the CV dataset (see Table 7).

Sudhakaran et al. [63] used a pre-trained AlexNet model trained on ImageNet for their method. They used the difference between consecutive video frames as input to capture temporal information. The results show that their method performs better on the CV dataset compared to our TL model. We should note that our TL model extracts features using a pre-trained X3D-M model trained on the Kinetics-400 dataset. This dataset contains a smaller number of examples with several people appearing in individual frames of the videos. In contrast, the ImageNet dataset contains a relatively higher number of examples with several people appearing in one frame. Therefore, we suggest that the extracted X3D-M features might be noisy and result in lower accuracy on datasets involving crowds such as the CV dataset.

Li et al. [72] used a DenseNet 3D-CNN to train and extract spatio-temporal features from videos. Their model was initialized with parameters from a pretrianed model trained on the Kinetics-400 dataset, similar to our FT model. However, their model had more CNN layers and higher model parameters, which contributed to its better accuracy on the CV and HF datasets compared to our TL model. It should be noted that DenseNet uses multi-layer feature concatenation for improved feature representation, but this approach requires more GPU memory and longer training times. Choqueluque-Roman et al. [104] followed an approach that used an I3D architecture in combination with a ResNet50 for feature extraction using human action tubes for training a deep learning model based on MIL. Their results showed that, according to the accuracy and AUC metrics, our models achieved better performance with relatively fewer model parameters, which confirms that training based on MIL may not achieve high classification accuracy.

Violence-Net [74] also used DenseNet for training and extracting feature maps. According to the ACC metric (see Table 6), their method using optical flow input achieved better scores than our FT model on the HF dataset. However, their architecture contains more model parameters and involves computing optical flow information, making it computationally more complex than ours. When pseudo-OF was used as input in their method, the accuracy decreased compared to our FT model. On the CV dataset, their model with more number of parameters achieved higher accuracy than our TL model. As previously mentioned, the extracted X3D-M features from videos involving crowds can be noisy and lead to less accurate results.

The method proposed by Romas et al. [67] used MobileNet V2 architecture for spatial feature extraction and LSTM modules for learning about temporal associations. Despite having a similar number of model parameters as our TL model, our methods achieved higher accuracy. As demonstrated by our results, methods that capture 3D spatio-temporal features directly from the video data, such as our proposed models, represent temporal associations more accurately and are therefore more effective at detecting violence in videos. This is due to the ability of our proposed models to accurately capture the full context and dynamics of the events depicted in the video, leading to improved performance in violence detection tasks.

The SPIL method [77] achieved higher accuracy scores than our TL model on the CV and RWF-2K datasets. However, this method requires significant computational resources due to the need to estimate 3D skeleton point clouds for interaction learning, making it impractical for practical applications.

The Violence Detection Network (VD-Net) [65] achieved better accuracy on HF and RWF-2K datasets compared to our TL model and has slightly more model parameters. VD-Net first detects humans and suspicious objects such as guns, which requires more computational resources than our TL model. However, the AUC scores for the TL model are comparable to VD-Net.

Finally, the CNN-LSTM-IOT model [64] has fewer parameters than all of the models under comparison, including ours, and it has been demonstrated that it can run on a low-cost Internet of Things (IoT) device like a Raspberry Pi. However, the model relies on spatial features for learning and performs poorly on the RLVS and RWF-2K datasets.

In summary, our experiments on individual datasets demonstrated that our FT model outperformed most of the state-of-the-art methods on most datasets while having fewer model parameters. Our TL model also achieved decent performance on all the datasets, despite having fewer trainable parameters than the FT model, as shown in Tables 3 & 4. This suggests that the TL model is relatively less adaptable to specific scenarios.

B. EXPERIMENTS ON GENERALIZABILITY

To study the adaptability of our proposed approaches to unseen videos, we conducted cross-dataset experiments where we trained a model on one dataset and evaluated its performance on another dataset. Table 8 shows the results from such one-on-one cross-validation tests in the top section (columns 5-8). It should be noted that, among the considered datasets, different datasets have different numbers of videos containing instances of violence and non-violence actions. In general, the number of samples available for training can greatly affect the learning capabilities of a deep learning model. Few and less diverse training samples can lead to model overfitting, where the model models some noise or random fluctuations in the training data is modeled very well, but it cannot generalize to new data. In our case, since we follow an inductive training approach using a pre-trained X3D-M model on the Kinetics-400 data, we suggest that our models are least influenced by the number of training samples, and our cross-validation results essentially show the ability of our models to learn the concept of violence.

Both ACC and AUC metrics show that there are several inconsistencies in the results across the considered datasets. To provide deeper insights into our cross-validation results, we plot the ACC and AUC scores obtained by training on a specific dataset and averaging the testing scores on the rest of the datasets in Figures 3 & 4 respectively for both FT and TL models. Each plot also shows the standard deviation of the metric scores obtained from the testing datasets, indicated by the red color lines. According to the metric scores, the trained FT and TL models on the CV dataset did not generalize well to other datasets (see bar plots in Figures 3(a) & 4(a)). This is anticipated since the CV dataset contains only examples of mass violence, and the other datasets do not contain many such examples. Also, the trained FT model on the HF dataset poorly generalized to other datasets, indicating that the HF dataset does not contain diverse examples of violence and contains monotonous fighting videos between hockey players. However, the TL model trained on this dataset showed better generalization than the FT model as indicated by the metric scores.

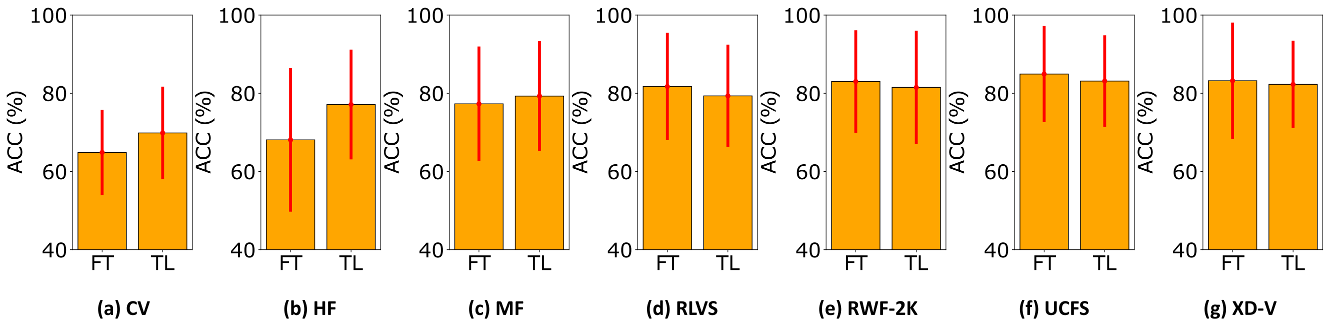
FT and TL models trained individually on datasets - MF, RLVS, RWF-2K, UCFS & XD-V performed satisfactorily in our cross-validation tests and generalized well to other datasets with average ACC scores close to or above 80% and average AUC scores close to or above 0.8. When considering both metrics, FT and TL models trained on UCFS and XD-V datasets exhibited the best generalization ability in our cross-validation studies. This suggests that these datasets, which we compiled, contain the most representative and diverse samples for violent and non-violent actions.

For closer examination, we also conducted leave-one-out cross-validation tests where we trained our models on all datasets except one, which was reserved for testing. The results of these tests are presented in the middle section of Table 8. The tests suggest that when the CV

TABLE 8. Cross dataset experiment results - One-on-one cross-validation test results are shown in the top section, leave one out cross-validation test results are shown in the middle section, and the bottom section shows the performance of our models on the training/testing folds used in Violence-Net [74] . To compare, ACC scores for Violence-Net using both OF and Pseudo-OF inputs are also provided for relevant datasets.

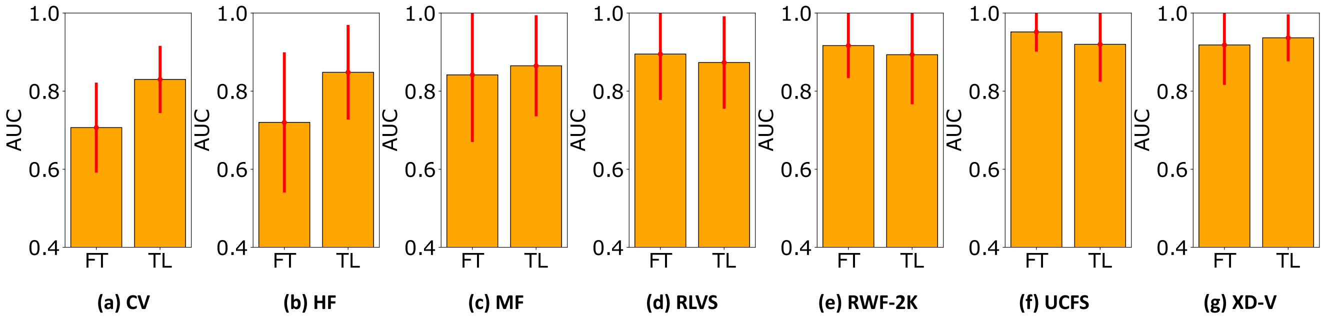
Dataset Training	Dataset Testing	ACC-OF [74]	ACC- Pseudo-OF [74]	ACC-FT (ours)	ACC-TL (ours)	AUC-FT (ours)	AUC-TL (ours)
CV	HF	**65.16**	64.76	56.5	50.6	0.68	**0.78**
CV	MF	60.02	59.48	56	**72**	0.54	**0.88**
CV	RLVS	58.76	58.32	78.58	**86.56**	0.84	**0.95**
CV	RWF-2K	-	-	**76.05**	73.05	0.81	**0.87**
CV	UCFS	-	-	68.11	**72.46**	0.75	**0.79**
CV	XD-V	-	-	53.99	**64.48**	0.62	**0.71**
HF	CV	62.56	61.22	**97.95**	**97.95**	0.99	**1**
HF	MF	65.18	64.86	45.5	**76**	0.55	**0.86**
HF	RLVS	58.22	57.36	76.58	**86.11**	0.83	**0.94**
HF	RWF-2K	-	-	69.9	**78.2**	0.76	**0.86**
HF	UCFS	-	-	64.51	**65.8**	0.68	**0.77**
HF	XD-V	-	-	54.09	**58.7**	0.51	**0.66**
MF	CV	52.32	51.77	86.48	**96.31**	0.99	**1**
MF	HF	54.92	53.5	**98.3**	87.8	**1**	0.95
MF	RLVS	56.72	55.8	79.29	**87.41**	0.94	**0.95**
MF	RWF-2K	-	-	75.6	**78.35**	0.84	**0.86**
MF	UCFS	-	-	**68.39**	66.36	0.71	**0.77**
MF	XD-V	-	-	55.77	**59.5**	0.57	**0.66**
RLVS	CV	-	-	90.98	**98.36**	0.99	**1**
RLVS	HF	69.24	68.86	**97**	84	**0.99**	0.95
RLVS	MF	75.82	74.64	**90**	85.5	**0.99**	0.95
RLVS	RWF-2K	67.84	66.68	**82.2**	78.55	**0.9**	0.87
RLVS	UCFS	-	-	65.71	**67.19**	0.75	**0.78**
RLVS	XD-V	-	-	**64.48**	62.43	**0.75**	0.69
RWF-2K	CV	-	-	90.98	**98.77**	0.95	**1**
RWF-2K	HF	-	-	**87.1**	81.9	0.93	**0.95**
RWF-2K	MF	-	-	**90.5**	87.5	**0.99**	0.96
RWF-2K	RLVS	-	-	**96.44**	92.13	**1**	0.98
RWF-2K	UCFS	-	-	**66.82**	**66.82**	**0.8**	0.78
RWF-2K	XD-V	-	-	**66.25**	61.9	**0.83**	0.69
UCFS	CV	-	-	80.33	**99.18**	0.89	**1**
UCFS	HF	-	-	64	**76.5**	0.92	**0.95**
UCFS	MF	-	-	**98**	86.5	**1**	0.96
UCFS	RLVS	-	-	91.47	**91.73**	**0.99**	0.98
UCFS	RWF-2K	-	-	**93.6**	78.35	**1**	0.89
UCFS	XD-V	-	-	**82.15**	66.52	**0.91**	0.74
XD-V	CV	-	-	81.97	**99.59**	0.9	**1**
XD-V	HF	-	-	54.3	**69.2**	0.72	**0.95**
XD-V	MF	-	-	**94**	83	**0.99**	0.96
XD-V	RLVS	-	-	**93.73**	90.27	**0.98**	**0.98**
XD-V	RWF-2K	-	-	**87.65**	74.8	**0.97**	0.89
XD-V	UCFS	-	-	**87.62**	76.8	**0.95**	0.84
HF+MF+RLVS+RWF-2K+UCFS+XD-V	CV	-	-	**88.93**	72.13	**0.95**	0.92
CV+MF+RLVS+RWF-2K+UCFS+XD-V	HF	-	-	96.9	**97.1**	**1**	0.99
CV+HF+RLVS+RWF-2K+UCFS+XD-V	MF	-	-	97	**99.5**	**1**	**1**
CV+HF+MF+RWF-2K+UCFS+XD-V	RLVS	-	-	**99.45**	90.37	**1**	0.969
CV+HF+MF+RLVS+UCFS+XD-V	RWF-2K	-	-	**97.25**	91.45	**1**	0.98
CV+HF+MF+RLVS+RWF-2K+XD-V	UCFS	-	-	75.97	**92.33**	0.86	**0.98**
CV+HF+MF+RLVS+RWF-2K+UCFS	XD-V	-	-	70.25	**93.07**	0.91	**0.98**
HF+MF+CV	RLVS	70.08	69.84	**99.45**	98.95	**1**	**1**
HF+MF+RLVS	CV	76	75.68	77.87	**80.74**	**0.95**	**0.95**
HF+RLVS+CV	MF	81.51	80.49	**99.5**	99	**1**	**1**
RLVS+MF+CV	HF	79.87	78.63	61.4	**97.1**	0.94	**0.99**

dataset was left out of the training, the TL model did not achieve a good ACC score. This is expected because the TL model extracts features from training videos using a pre-trained X3D-M model that was trained on the Kinetics-400 dataset, which does not contain many examples involving crowd participation. However, the FT model achieved decent accuracy, indicating that the datasets other than CV contain a sufficient number of examples for learning about violence involving crowds. In line with the results obtained in the one-on-one cross-validation tests, leaving out the UCFS or XD-V datasets from training resulted in poor performance for the FT model. However, the performance of the TL model did not drop when these datasets were left out of the training, indicating that the TL model generalizes better than the FT model. To confirm this, we collected all instances of the one-on-one cross-validation tests when a specific dataset was being tested for further examination. In Figures 5 and 6, we plot the ACC and AUC scores obtained by averaging the testing accuracy scores on a specific dataset when all other datasets were used individually for training



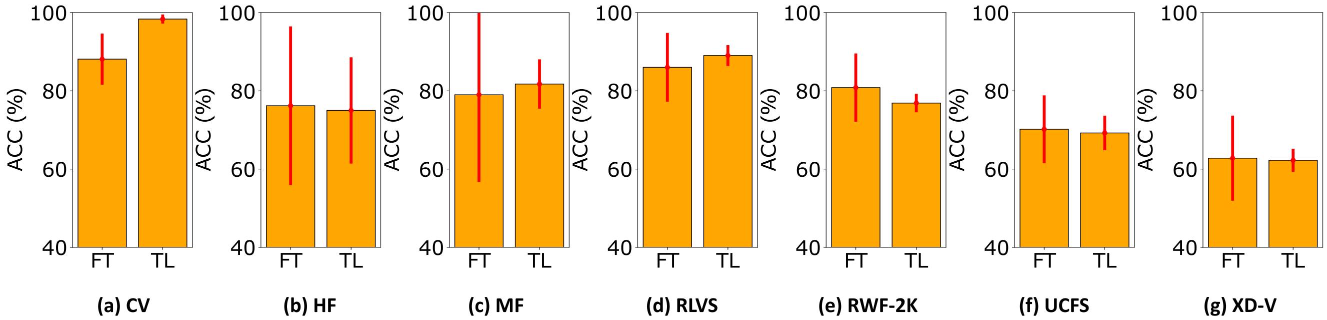
(a) CV (b) HF (c) MF (d) RLVS (e) RWF-2K (f) UCFS (g) XD-V

FIGURE 3. ACC scores for each dataset obtained by training on that specific dataset and averaging the testing accuracy scores on the rest of the datasets. The scores are shown for both FT and TL models. The red lines indicate the standard deviation of the testing scores from the mean value.



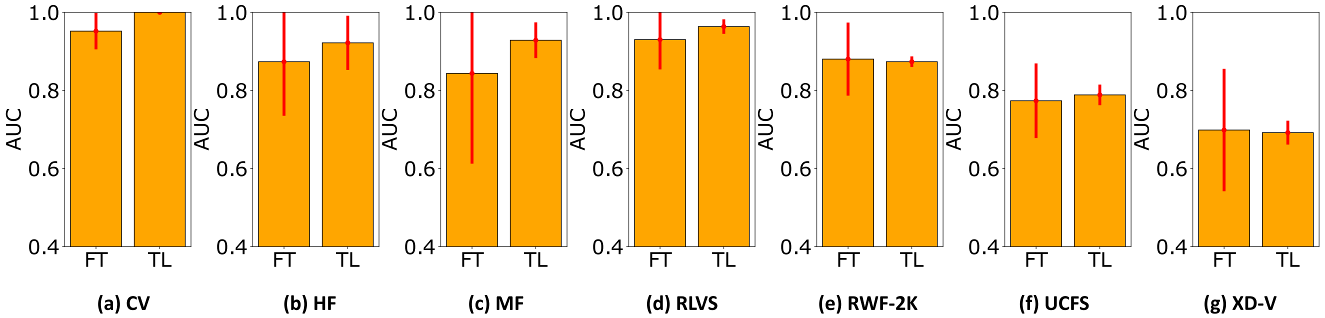
(a) CV (b) HF (c) MF (d) RLVS (e) RWF-2K (f) UCFS (g) XD-V

FIGURE 4. AUC scores for each dataset obtained by training on that specific dataset and averaging the testing accuracy scores on the rest of the datasets. The scores are shown for both FT and TL models. The red lines indicate the standard deviation of the testing scores from the mean value.



(a) CV (b) HF (c) MF (d) RLVS (e) RWF-2K (f) UCFS (g) XD-V

FIGURE 5. ACC scores for each dataset obtained by averaging the testing accuracy scores when all other datasets are used individually for training. The average scores are indicated for both FT and TL models. Each plot also includes red lines indicating the standard deviation in ACC scores obtained during testing.



(a) CV (b) HF (c) MF (d) RLVS (e) RWF-2K (f) UCFS (g) XD-V

FIGURE 6. AUC scores for each dataset obtained by averaging the testing accuracy scores when all other datasets are used individually for training. The average scores are indicated for both FT and TL models. Each plot also includes red lines indicating the standard deviation in AUC scores obtained during testing.

for both the FT and TL models. Each plot also shows the standard deviation of the metric scores obtained during testing, indicated by red lines. Based on these plots, it is evident that overall, the TL model showed better capability to

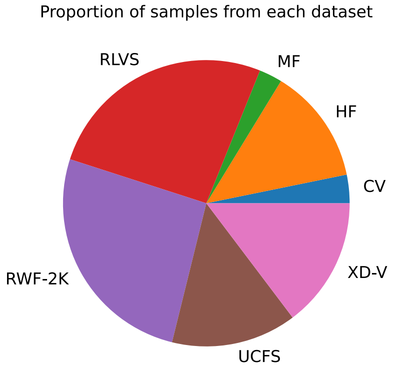


FIGURE 7. The pie chart illustrates the distribution of samples from different datasets used for training and testing in the combined dataset experiments.

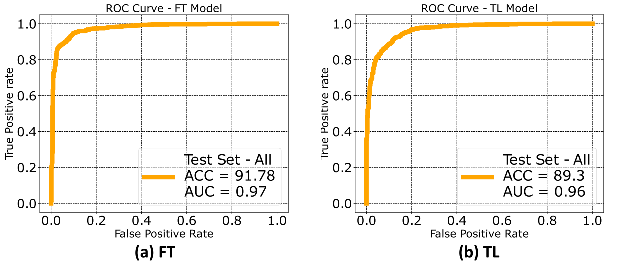


FIGURE 8. Performance of FT and TL models (left and right respectively) on all combined dataset shown using ROC curve. Both ACC and AUC scores show that the FT model performs better than the TL model on this dataset.

generalize and had lower standard deviation within the testing accuracy scores for individual datasets when compared to the FT model.

To the best of our knowledge, results from cross-validation studies are rarely presented in the literature for violence detection algorithms. For comparison, we have also included the cross-validation results from Violence-Net [74] using both OF and pseudo-OF inputs (columns 2-3) in the Table 8. Only ACC scores are provided since AUC scores are not presented in their original study. Also the authors of Violence-Net only used four datasets in their experiments, so results are presented only for these four datasets. The comparison results show that, on average, our TL and FT models consistently outperformed Violence-Net using both OF and pseudo-OF inputs. This suggests that our approaches are more accurate and better able to generalize to unseen scenarios for violence detection when compared to Violence-Net.

C. EXPERIMENTS WITH ALL COMBINED DATASET

In this section, we describe our experiments using combined dataset and discuss the performance of the FT and TL models on this dataset. To ensure a fair distribution of training samples from each dataset, we selected and grouped the predefined 80% of the data from each dataset for training and the remaining 20% for testing. Figure 7 illustrates the proportion of samples from each dataset. The ROC curves, including the obtained ACC and AUC scores are presented in Figure 8. Results from both metrics suggest that our models performed satisfactorily on this dataset, with the FT model achieving slightly better performance. It is worth noting again that the TL model has fewer trainable parameters than the FT model.

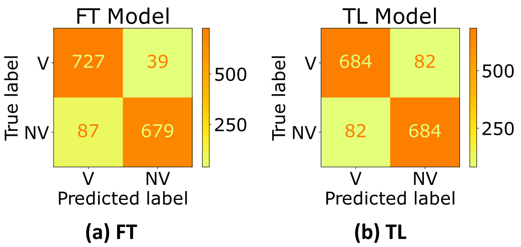
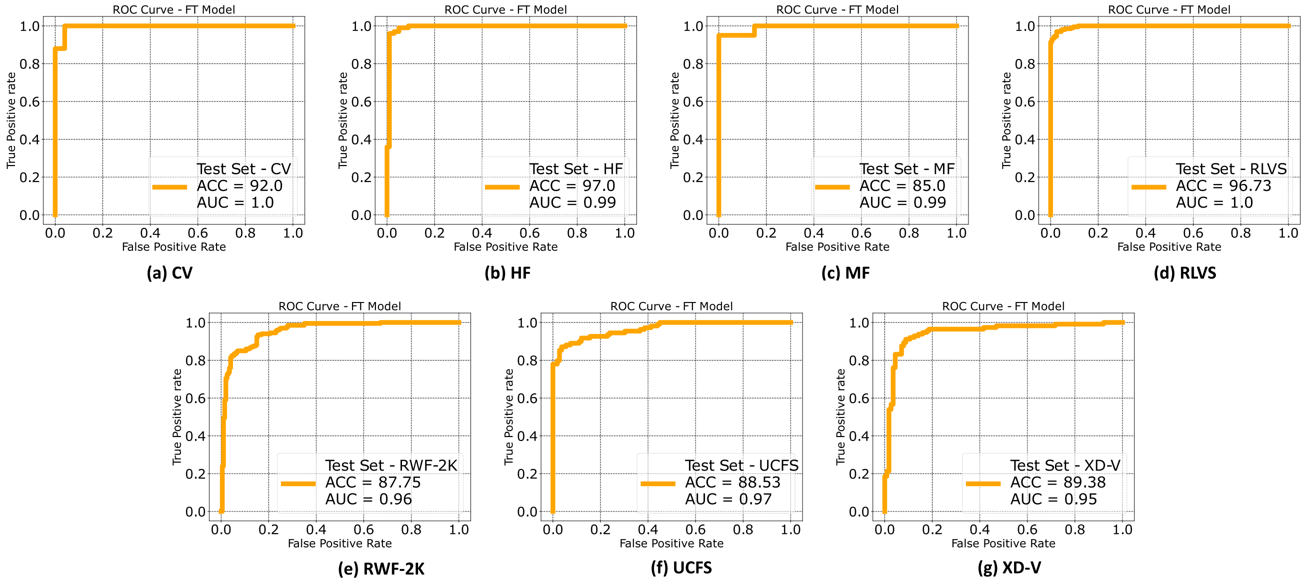


FIGURE 9. Confusion matrices for FT and TL models obtained after testing on all combined dataset. Results show that TL model produced more number of combined false negatives & false positives than FT model.

For further analysis, we present the confusion matrices for both models in Figure 9. The rows of the confusion matrix represent the true labels, or the expected output, for the Violent (V) or Non-Violent (NV) classes, while the columns represent the predicted labels. In our case, the following are the four numbers presented in the confusion matrices:

- **True Positives (TP)** - the number of videos actually containing violence that were predicted as containing violence. TP are shown in the first row, first column of the confusion matrix.
- **False Negatives (FN)** - the number of videos actually containing violence that were predicted as not containing violence. FN are shown in the first row, second column of the confusion matrix.
- **False Positives (FP)** - the number of videos actually not containing violence that were predicted as containing violence. FP are shown in the second row, first column of the confusion matrix.
- **True Negatives (TN)** - the number of videos actually not containing violence that were predicted as not containing violence. TN are shown in the second row, second column of the confusion matrix.

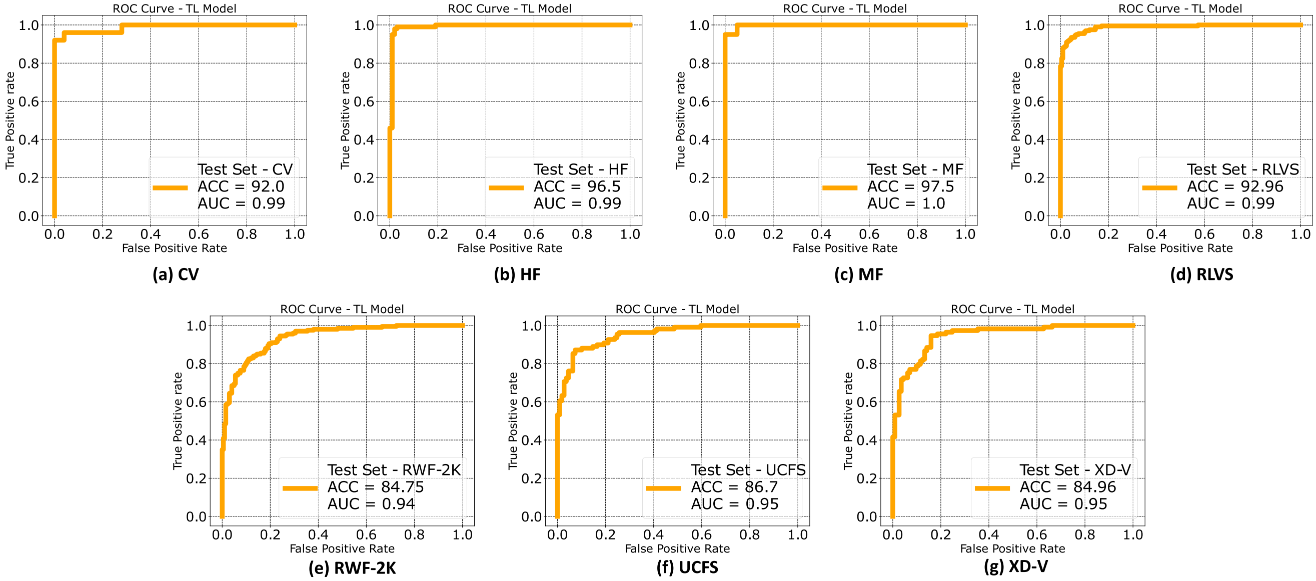
From the confusion matrices, it is evident that the TL model produced a greater number of combined FP & FN than the FT model. For detailed evaluation, we also studied and presented the metric scores and confusion matrices for individual datasets. Figures 10 & 12 show the results from the FT model, while Figures 11 & 13 show the results from the TL model. We note that overall, for both models, the number of FP & FN is balanced for all datasets, indicating that the training samples from both the violence and non-violence classes are balanced. Additionally, from the confusion matrices for individual datasets, it is clear that for most of the datasets, the TL model produced a greater number of combined FP & FN. Our hypothesis is that the fixed nature of the extracted X3D-M features in the TL model does not provide sufficient flexibility to accurately recognize the attributes of violent actions.



(a) CV (b) HF (c) MF (d) RLVS

(e) RWF-2K (f) UCFS (g) XD-V

FIGURE 10. ROC curves including ACC and AUC scores obtained by testing on individual datasets using FT model trained on all combined dataset.



(a) CV (b) HF (c) MF (d) RLVS

(e) RWF-2K (f) UCFS (g) XD-V

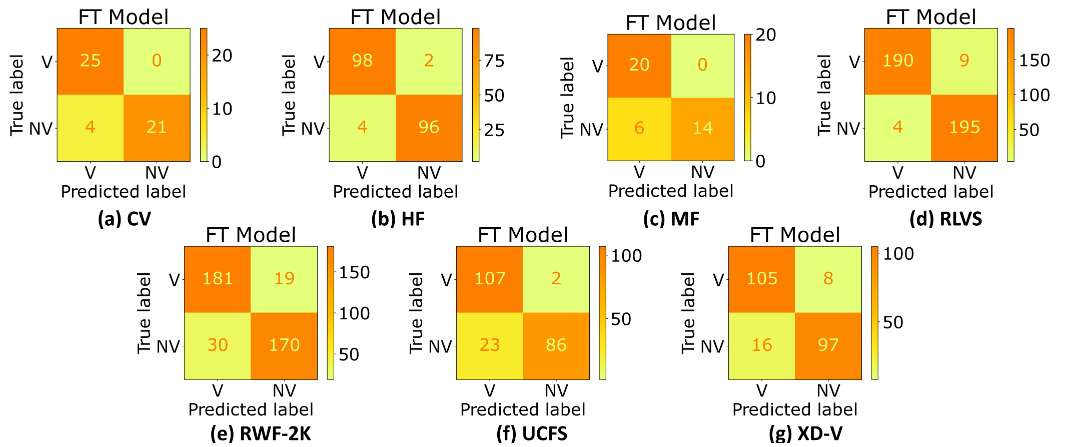
FIGURE 11. ROC curves including ACC and AUC scores obtained by testing on individual datasets using TL model trained on all combined dataset.

D. ANALYSIS OF THE DATASETS AND CHALLENGES

Even though the CV dataset has a smaller number of examples, both models trained on the combined dataset performed well on it. However, our leave-one-out cross-validation results indicate that when the CV dataset was excluded from training, the models did not perform well. This suggests that the CV dataset contains diverse and representative examples of crowd violence. However, it should be noted that the dataset only includes examples of violence involving crowds and the models trained on it did not generalize well to other types of datasets.

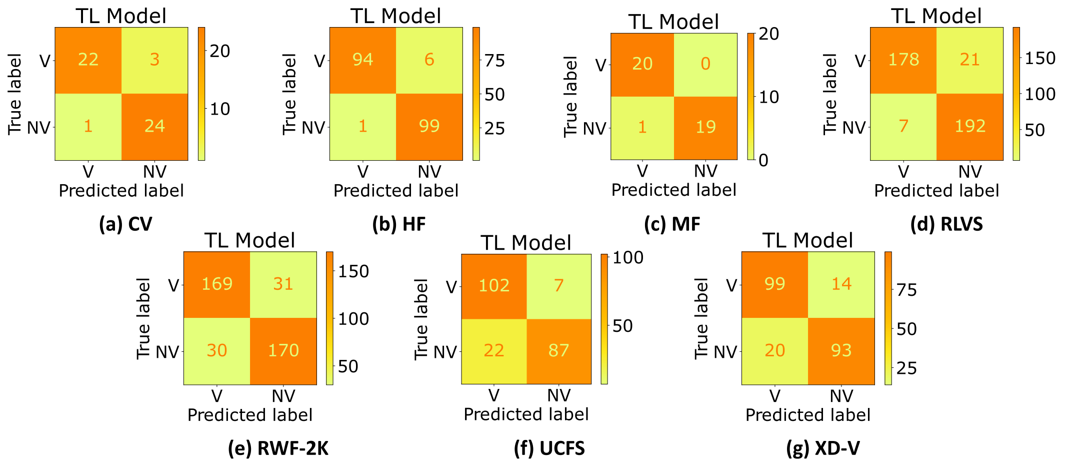
The HF dataset, on the other hand, contains a relatively larger number of training samples, primarily consisting of monotonous fighting videos between hockey players. Both our FT and TL models trained on the combined dataset performed well on this dataset as well. However, our leave-one-out cross-validation test revealed that excluding this dataset did not significantly decrease the accuracy of our models. Additionally, the model trained solely on the HF dataset did not generalize well to other datasets, as shown in figure 3. In line with our previous results on generalizability, highly monotonic datasets like the HF dataset are less useful for developing robust deep-learning models for violence detection.

Since our models use pre-processed input containing 16 uniformly sampled temporal frames, the duration of a video and the number of frames per second can affect the model's performance. The MF dataset has significant



(a) CV (b) HF (c) MF (d) RLVS (e) RWF-2K (f) UCFS (g) XD-V

FIGURE 12. Confusion matrices obtained by testing on individual datasets using FT model trained on all combined dataset.



(a) CV (b) HF (c) MF (d) RLVS (e) RWF-2K (f) UCFS (g) XD-V

FIGURE 13. Confusion matrices obtained by testing on individual datasets using TL model trained on all combined dataset.

fluctuations in the FPS values of the training videos (as seen in table 2), which is not favorable for training our violence detection models. Additionally, this dataset has the least number of training samples compared to others and models trained solely on this dataset did not generalize well to other datasets. We hypothesize that these drawbacks of this dataset could be the reason for the decrease in the performance of the FT model (trained on the combined dataset) on this dataset. On the other hand, due to better generalizability, the TL model trained on the combined dataset performed well on this dataset.

In addition, our leave-one-out cross-validation test shows that the MF, RLVS, and RWF-2K datasets do not contribute significantly to model generalizability. The RLVS dataset mainly contains examples of two people fighting, which are also present in other datasets such as UCFS and XD-V. The RWF-2K dataset contains videos that are encoded at 30 frames per second, but we have observed that there are videos captured at very low fps, resulting in repeated frames to create 30 fps videos. Additionally, most examples in this dataset are repetitive in terms of environment and lighting conditions and lack diversity. However, it is important to note that the RLVS and RWF-2K datasets contain the highest number of examples, which can lead the model trained on the combined dataset to better represent scenarios in these datasets. We hypothesize that due to the aforementioned drawbacks specific to each of these two datasets, our models

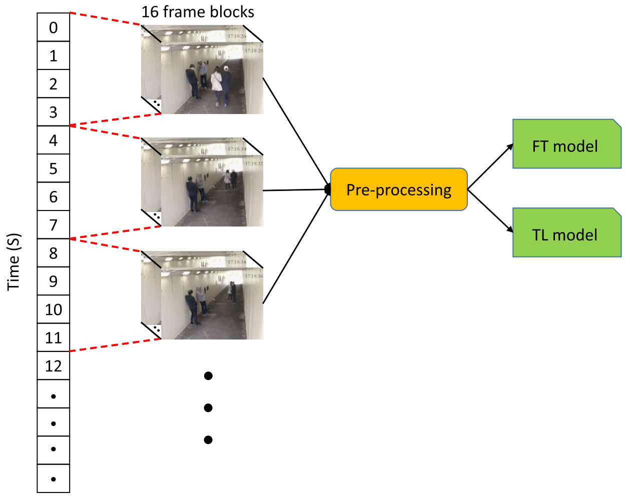


FIGURE 14. Schematic diagram of our standalone application. 16 frame-blocks are extracted from each four-second video clip which are pre-processed and input to FT or TL model.

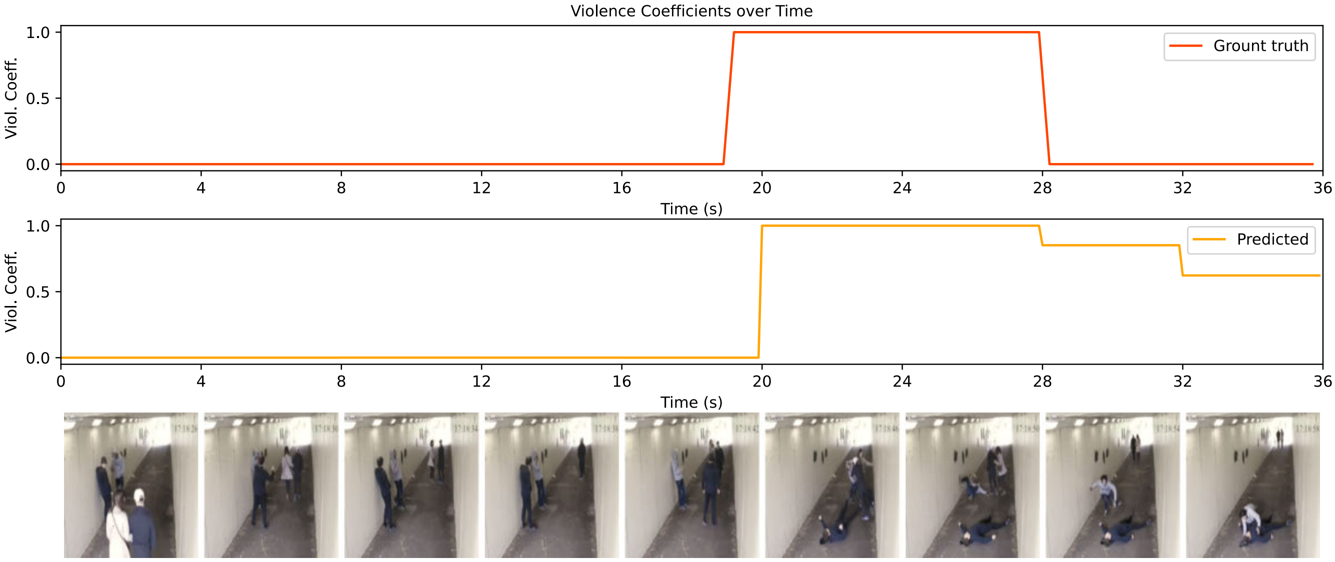


FIGURE 15. Results of our standalone system using FT model on a video clip from the testing set of the original UCF-Crime dataset. The video clip includes an instance of violence between two normal events. Row 1: Ground truth of violence coefficients over time. Row 2: Predicted violence coefficients on 4-second video segments. Row 3: Keyframes extracted from the video.

trained on the combined dataset did not perform very well on the RLVS and RWF-2K datasets.

Finally, our results show that models trained on our UCFS and XD-V datasets generalize better to other datasets (as seen in figure 3). Also, when these datasets were excluded from training, the performance of our models dropped significantly, indicating that these datasets contain well-calibrated, diverse video footage, which is highly relevant for training practical deep learning algorithms for violence detection (as seen in table 8). However, these datasets contain a fewer number of training examples compared to RLVS and RWF-2K. Additionally, the UCFS and XD-V datasets contain forms of violence such as explosions and road accidents, which are not distinctly available in other datasets. Due to this, we hypothesize that our models trained on the combined dataset did not perform very well on the UCFS and XD-V datasets. Overall, with fewer false positives and false negatives, our FT model performed better on the combined dataset than the TL model.

E. EXPERIMENTS WITH VIDEO COMPRESSION

Depending on the available hardware resources, it may be necessary to stream the surveillance video to a remote server for actual classification and violence detection. Additionally, depending on the available network resources, there may not be sufficient bandwidth to stream the video in its native resolution and quality. In several fields where video streaming is involved, video compression techniques are commonly applied to reduce the video bit-rate, which can introduce artifacts in the video. To study the effect of such video artifacts on the performance of our TL and FT models, we generated compressed video streams with varying bit-rates - 300, 500, 1000 and 1500 Kbps.

For this experiment, we randomly selected two datasets, RWF-2K and CV and compressed the testing videos from these two datasets. Multiple videos were generated with the different bit-rates using ffmpeg [105]. We used the models trained on the combined dataset for this experiment and the testing results are presented in Table 9. Our study shows that both TL and FT models did not show significant fluctuations in the performance and performed well even under extreme compression (300 Kbps). This suggests that our trained models did not model the noise in the training videos and focused on learning the concept of violence.

TABLE 9. Results from video compression experiments - The top section shows results for the CV dataset and the bottom section shows results for the RWF-2K dataset using ACC and AUC metrics.

Dataset (bitrate in Kbps)	ACC-FT	ACC-TL	AUC-FT	AUC-TL
CV (1500)	92.0	92.0	0.99	0.98
CV (1000)	92.0	92.0	1	0.98
CV (500)	92.0	92.0	0.99	0.98
CV (300)	92.0	92.0	0.99	0.98
RWF-2K (1500)	88.25	85.0	0.95	0.93
RWF-2K (1000)	89.0	85.0	0.95	0.93
RWF-2K (500)	88.5	84.25	0.95	0.93
RWF-2K (300)	88.5	83.0	0.96	0.92

F. STANDALONE IMPLEMENTATION AND PERFORMANCE

We have implemented a standalone application for violence detection using the PyTorch deep learning library and using our FT and TL models that are trained on the combined dataset. The application design is outlined in Figure 14 and can be easily extended for usage in surveillance applications. The incoming video stream is divided into non-overlapping video segments of four seconds, from which 16 video frames are extracted per segment using uniform temporal sampling. These 16-frame blocks are pre-processed and then used as input for either the FT model or TL model to determine a violence coefficient for the current segment. The application

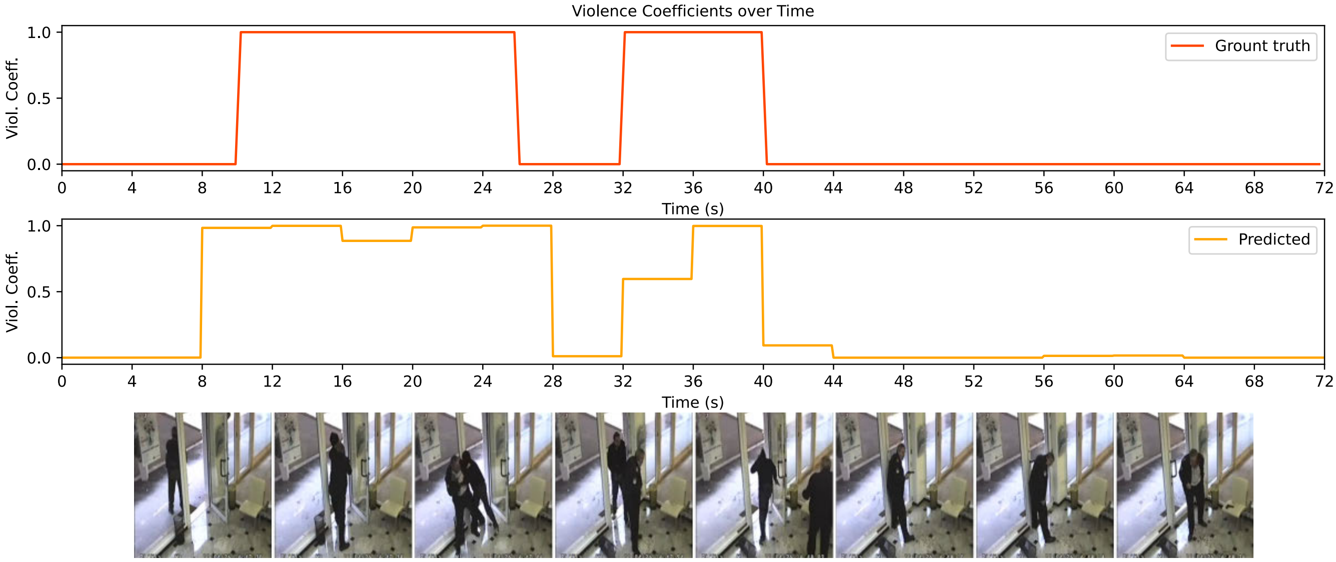


FIGURE 16. Results of our standalone system using FT model on a longer video clip from the testing set of the UCF-Crime dataset with two instances of violence. Row 1: Ground truth of violence coefficients over time. Row 2: Predicted violence coefficients on 4-second video segments. Row 3: Key frames extracted from the video.

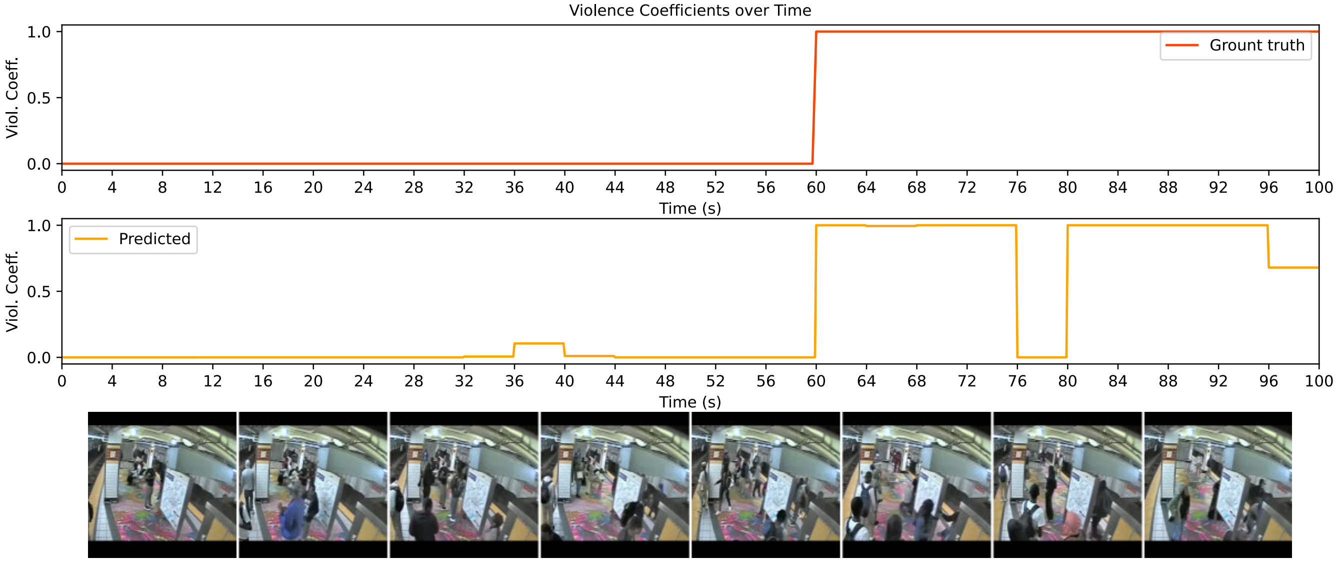


FIGURE 17. Results of our standalone system using FT model on a complex and longer video sequence from the testing set of the UCF-Crime dataset, showing a crowd involved in violence at a metro station. Row 1: Ground truth of violence coefficients over time. Row 2: Predicted violence coefficients on 4-second video segments. Row 3: Keyframes extracted from the video.

was implemented on an Ubuntu Linux operating system using an AMD Ryzen Threadripper 1950X 16-core processor and a Nvidia GeForce GTX 1080 Ti GPU with the CUDA toolbox for running our trained PyTorch models.

Our results indicate that, when combined with block extraction and pre-processing, both the FT and TL models require an average of 0.06 seconds on average to infer a violence coefficient for each four second-video segment. The pre-processing was implemented on the CPU, consuming an average of 0.04 seconds. Therefore, the average time required to run the FT or TL model is 0.02 seconds. It should be noted that, the dense or fully connected layers of the models consume minimal computational resources in practice. As a result, even though the TL model has more parameters than the FT model, the average time required to run both models is similar.

To give a thorough understanding of the performance of our standalone system, we have graphically represented the progression of violence coefficients over time using the FT model that showed the most optimal results on the combined dataset. In figures 15 to 19, we illustrate our classification outcomes on selected video samples that exhibit the capability of our system in identifying violence. Each figure comprises of three sections: the top row displays the actual graph of violence coefficients over time, where the coefficients are set to one during the occurrence of violence.

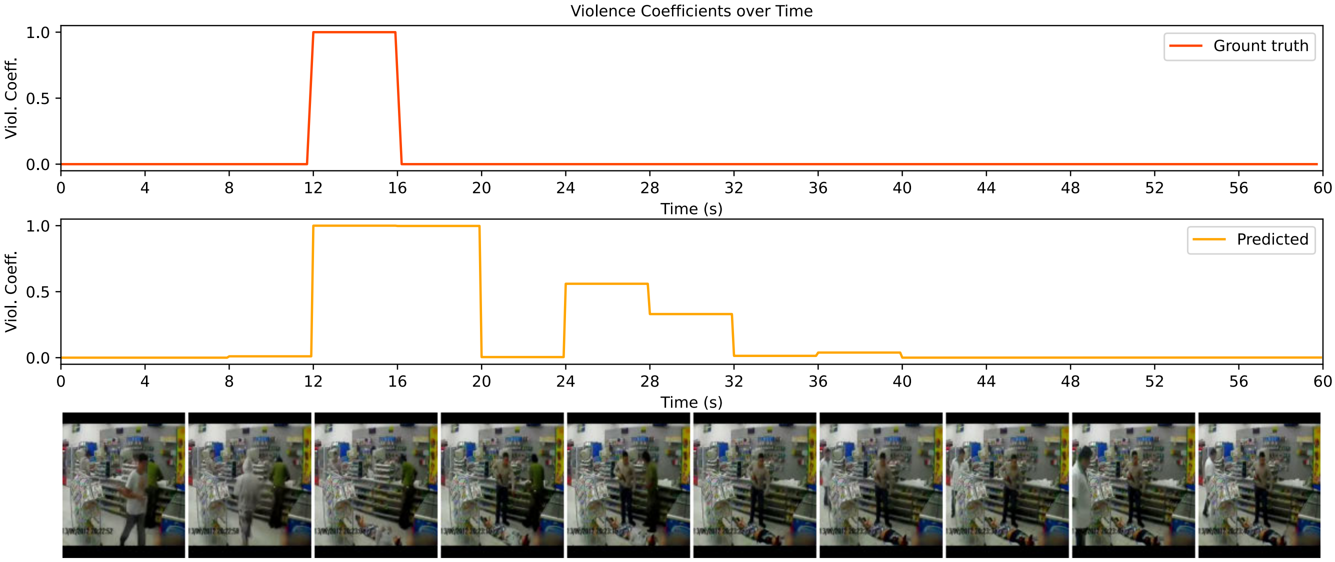


FIGURE 18. Results of our standalone system using FT model on a video clip from the testing set of the UCF-Crime dataset that contains a single and short instance of violence in the form of shooting. Row 1: Ground truth of violence coefficients over time. Row 2: Predicted violence coefficients on 4-second video segments. Row 3: Keyframes extracted from the video.

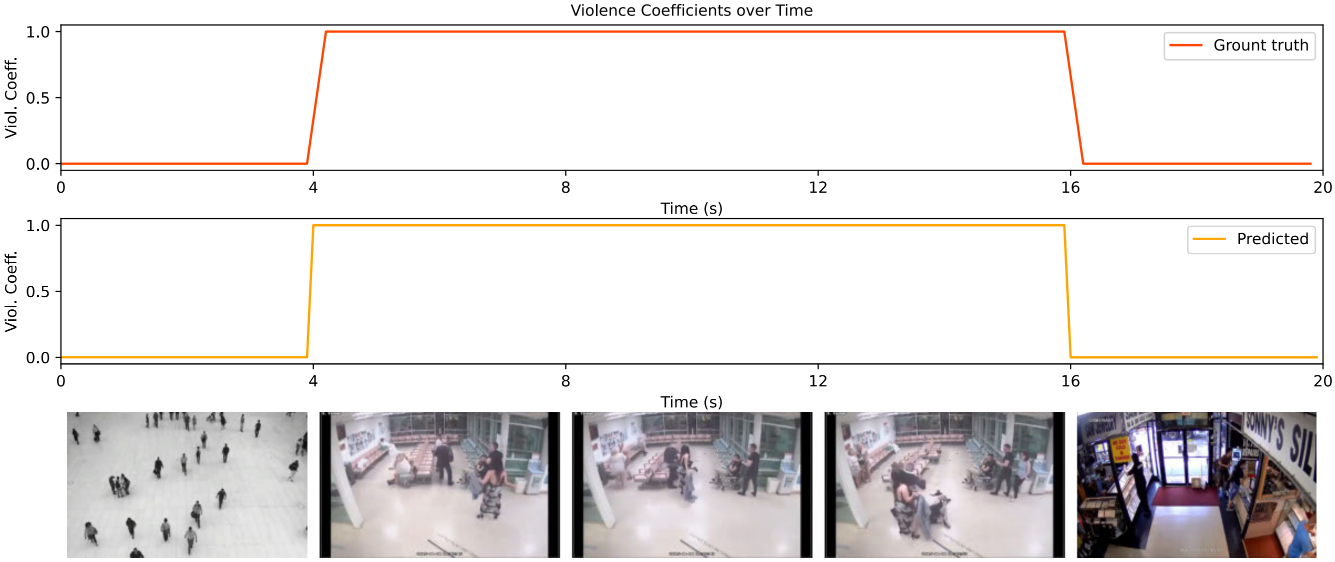


FIGURE 19. Results of our standalone system using FT model on a video compiled from 3 random videos of Smart-City CCTV Violence Detection Dataset, one violent and two non-violent videos concatenated in such a way that the violent video is placed in between two non-violent videos. Row 1: Ground truth of violence coefficients over time. Row 2: Predicted violence coefficients on 4-second video segments. Row 3: Keyframes extracted from the video.

The middle row illustrates the predicted violence coefficients by our FT model on a series of non-overlapping video segments with a duration of four seconds. The bottom row shows key frames extracted from the videos.

Figures 15 to 18 demonstrate the performance of our standalone system on video clips from the testing set of the original UCF-Crime dataset. These video clips include different scenarios such as instances of violence amidst normal events (as illustrated in figure 15), multiple occurrences of violence (as illustrated in figure 16), a crowd engaging in violence at a metro station (complex and long video sequence as illustrated in figure 17), and a single, short instance of violence in the form of shooting (as illustrated in figure 18). The predicted violence coefficients align closely with the ground truth, indicating the algorithm's capability to accurately identify and predict instances of violence in video segments of various lengths and complexities.

To further evaluate our system, we also created a video sequence by combining random video clips from the Smart-City CCTV Violence Detection Dataset [106], which was not used in our study. As shown in figure 19, our results exhibit outstanding performance on this compiled sequence as well. This illustrates the adaptability of our algorithm and its capability to perform well on new and unseen data.

Figures 15 to 19 also demonstrate the areas where our standalone system falls short, which require further improvement in future research. We have noticed certain situations where our tested FT model triggers false alarms in the standalone implementation. For instance, when a person suddenly starts running or crawling (as shown in the keyframes from the 28th second in figure 15), it is detected as violence, but with a lower level of violence coefficient. In the original UCF-crime dataset, activities such as crawling or sudden fleeing are not considered violence. Nevertheless, in real-world surveillance scenarios, such actions may appear suspicious and require more investigation.

In situations where there is occlusion and the individuals or objects engaged in violence are only partially visible, the model may have difficulty identifying the violence. This can be observed in the predicted violence coefficients between the 32nd and 36th second in figure 16, where a person is holding a gun in his hand which is partly visible and hidden by his body.

As previously noted, videos that include people in crowds situated closely together can lead to inaccuracies in our system. Figure 17 between seconds 76 and 80 illustrates this scenario, where the predicted violence coefficient suddenly falls to zero even though violence is happening during this time. As mentioned earlier, our training dataset has limited examples of crowds, and including more such examples in future work is suggested.

V. CONCLUSION AND FUTURE WORK

In this work, we addressed the problem of efficient violence detection for automated surveillance applications by adapting the computationally lightweight X3D-M deep learning architecture for learning and detecting violence patterns from videos. We proposed two architectures, FT and TL, for classifying video clips containing violence, which leverage action recognition features learned from the Kinetics-400 dataset.

In order to perform a detailed analysis and performance evaluation of the proposed approaches, we collected and extended seven different datasets in our study. In the past, several deep learning-based methods for violence detection have focused on datasets involving mostly fighting between two or more people for experiments. However, it is important to note that the spectrum of actions and visual patterns representing violence is far wider. For example, violence happening between a group of people in the form of a fight is visually very different from violence involving the use of objects such as a gun or violence involving explosions. To also incorporate such cases, we annotated several videos from the UCF and XD-Violence datasets for our experiments.

Using our collected videos, the FT model optimizes the X3D-M parameters learned from the Kinetics-400 dataset, while the TL model extracts spatio-temporal features first, without modifying the X3D-M parameters (trained on the Kinetics-400 dataset), to train multiple fully connected layers. Our experiments with individual datasets show that both models performed well in terms of ACC and AUC scores on the collected datasets. However, the FT model performed better than most of the state-of-the-art methods on popular datasets with relatively fewer model parameters.

In the previous works on violence detection, cross-dataset evaluations have not been thoroughly studied. We argue that these evaluations are crucial for understanding the prominence of various datasets as well as developed deep learning models. In this work, we bridge this gap by providing comprehensive evaluations, including one-on-one cross-dataset validation and leave-one-out cross-validation. Our cross-dataset tests showed that the TL model generalizes better to unseen scenarios than the FT model. However, when tested on the combined dataset, the FT model achieved better performance, while the TL model produced a higher number of combined false positives and false negatives. Further tests on individual datasets show that models trained on the combined dataset did not perform well in several cases when compared to the performance of models trained on individual datasets. This highlights the inconsistencies in the publicly available datasets for violence detection. Additionally, results from comparisons with several methods in literature have shown limitations of both the developed methods and existing datasets.

We note that the existing public datasets for violence detection are inconsistent in terms of the video duration, FPS, the number of videos available for training and testing and the forms of violence depicted. Furthermore, these existing datasets are not particularly representative of surveillance applications. Our results indicate that, in the future, there is a great need for the development of diverse and meaningful large-scale datasets, including footage from real-world surveillance, to make these technologies practically feasible. In the future, we plan to take steps towards constructing such a large-scale dataset. Once such a datasets is available, we also plan to re-evaluate the models presented in this work for more general results.

We also presented a computationally light and functional standalone system architecture for implementing the proposed models in practical surveillance applications. In this architecture, we extracted and evaluated non-overlapping video segments having a duration of four seconds from the incoming video stream. This strategy may fail in cases where an event of violence begins at the end of a segment and ends before the end of the next segment. In the future, we also plan to develop smart strategies to handle such scenarios, such as reducing the size of video segments adaptively and/or using overlapped segments. The main focus in developing such strategies will be on achieving the best computational speed and accuracy trade-off.

ACKNOWLEDGMENT

This research was supported by the National Research, Development and Innovation Office through the grant TKP- 2021_02-NVA-27.

REFERENCES

[1] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[2] L. Liu, L. Shao, and P. Rockett, "Genetic programming-evolved spatio-temporal descriptor for human action recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–12.

[3] P. Bilinski and F. Bremond, "Human violence recognition and detection in surveillance videos," in *Proc. 13th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2016, pp. 30–36.

[4] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced computer vision with Microsoft Kinect sensor: A review," *IEEE Trans. Cybern.*, vol. 43, no. 5, pp. 1318–1334, Jun. 2013.

[5] L. Cruz, D. Lucio, and L. Velho, "Kinect and RGBD images: Challenges and applications," in *Proc. 25th SIBGRAPI Conf. Graph., Patterns Images Tuts.*, Aug. 2012, pp. 36–49.

[6] ASUS. (2017). *Xtion 2 Depth Sensor*. Accessed: Oct. 29, 2022. [Online]. Available: https://www.asus.com/ch-en/networking-iot-servers/smart-home/security-camera/xtion-2/

[7] Intel. (2022). *Real Sense Depth Camera D435*. Accessed: Oct. 29, 2022. [Online]. Available: https://www.intelrealsense.com/depth-camera-d435/

[8] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, "Real-time human pose recognition in parts from single depth images," in *Proc. CVPR*, Jun. 2011, pp. 1297–1304.

[9] W. Ding, K. Liu, X. Fu, and F. Cheng, "Profile HMMs for skeleton-based human action recognition," *Signal Process., Image Commun.*, vol. 42, pp. 109–119, Mar. 2016.

[10] R. Vemulapalli, F. Arrate, and R. Chellappa, "Human action recognition by representing 3D skeletons as points in a lie group," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 588–595.

[11] R. Chaudhry, F. Ofli, G. Kurillo, R. Bajcsy, and R. Vidal, "Bio-inspired dynamic 3D discriminative skeletal features for human action recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2013, pp. 471–478.

[12] L. Xia, C.-C. Chen, and J. K. Aggarwal, "View invariant human action recognition using histograms of 3D joints," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Workshops*, Jun. 2012, pp. 20–27.

[13] J. Wang, Z. Liu, Y. Wu, and J. Yuan, "Mining actionlet ensemble for action recognition with depth cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 1290–1297.

[14] J. Wang, Z. Liu, and Y. Wu, *Human Action Recognition With Depth Cameras* (SpringerBriefs in Computer Science). Berlin, Germany, 2014.

[15] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[16] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 2, pp. 84–90, Jun. 2017.

[17] U. Buchler, B. Brattoli, and B. Ommer, "Improving spatiotemporal self-supervision by deep reinforcement learning," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 770–786.

[18] J. Wang, J. Jiao, L. Bao, S. He, Y. Liu, and W. Liu, "Self-supervised spatio-temporal representation learning for videos by predicting motion and appearance statistics," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 4006–4015.

[19] X. Long, C. Gan, G. de Melo, J. Wu, X. Liu, and S. Wen, "Attention clusters: Purely attention based local feature integration for video classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 7834–7843.

[20] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, "Rethinking spatiotemporal feature learning: Speed-accuracy trade-offs in video classification," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2018, pp. 305–321.

[21] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "ViViT: A video vision transformer," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 6836–6846.

[22] L. Wang, W. Li, W. Li, and L. Van Gool, "Appearance-and-relation networks for video classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1430–1439.

[23] D. Tran, H. Wang, M. Feiszli, and L. Torresani, "Video classification with channel-separated convolutional networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 5552–5561.

[24] K. Cao, J. Ji, Z. Cao, C.-Y. Chang, and J. C. Niebles, "Few-shot video classification via temporal alignment," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 10618–10627.

[25] S. Wu, B. E. Moore, and M. Shah, "Chaotic invariants of Lagrangian particle trajectories for anomaly detection in crowded scenes," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2054–2060.

[26] B. Antic and B. Ommer, "Video parsing for abnormality detection," in *Proc. Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2415–2422.

[27] B. Zhao, L. Fei-Fei, and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *Proc. CVPR*, Jun. 2011, pp. 3313–3320.

[28] L. Kratz and K. Nishino, "Anomaly detection in extremely crowded scenes using spatio-temporal motion pattern models," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1446–1453.

[29] W. Li, V. Mahadevan, and N. Vasconcelos, "Anomaly detection and localization in crowded scenes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 1, pp. 18–32, Jan. 2014.

[30] A. D. Giorno, J. A. Bagnell, and M. Hebert, "A discriminative framework for anomaly detection in large videos," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 334–349.

[31] R. Bensch, N. Scherf, J. Huisken, T. Brox, and O. Ronneberger, "Spatiotemporal deformable prototypes for motion anomaly detection," *Int. J. Comput. Vis.*, vol. 122, no. 3, pp. 502–523, May 2017.

[32] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis, "Learning temporal regularity in video sequences," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 733–742.

[33] W. Luo, W. Liu, and S. Gao, "A revisit of sparse coding based anomaly detection in stacked RNN framework," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 341–349.

[34] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 3551–3558.

[35] D. Xu, E. Ricci, Y. Yan, J. Song, and N. Sebe, "Learning deep representations of appearance and motion for anomalous event detection," 2015, *arXiv:1510.01553*.

[36] Y. Cong, J. Yuan, and J. Liu, "Sparse reconstruction cost for abnormal event detection," in *Proc. CVPR*, Jun. 2011, pp. 3449–3456.

[37] H. Park, J. Noh, and B. Ham, "Learning memory-guided normality for anomaly detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 14372–14381.

[38] Y. S. Chong and Y. H. Tay, "Abnormal event detection in videos using spatiotemporal autoencoder," in *Proc. Int. Symp. Neural Netw.*, Cham, Switzerland: Springer, 2017, pp. 189–196.

[39] S. Szymanowicz, J. Charles, and R. Cipolla, "Discrete neural representations for explainable anomaly detection," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 148–156.

[40] M. Ravanbakhsh, M. Nabi, E. Sangineto, L. Marcenaro, C. Regazzoni, and N. Sebe, "Abnormal event detection in videos using generative adversarial nets," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 1577–1581.

[41] M. Ravanbakhsh, E. Sangineto, M. Nabi, and N. Sebe, "Training adversarial discriminators for cross-channel abnormal event detection in crowds," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2019, pp. 1896–1904.

[42] L. Wang, F. Zhou, Z. Li, W. Zuo, and H. Tan, "Abnormal event detection in videos using hybrid spatio-temporal autoencoder," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2018, pp. 2276–2280.

[43] M. Mathieu, C. Couprie, and Y. LeCun, "Deep multi-scale video prediction beyond mean square error," 2015, *arXiv:1511.05440*.

[44] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection—A new baseline," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6536–6545.

[45] A. Dosovitskiy, P. Fischer, E. Ilg, P. Hausser, C. Hazirbas, V. Golkov, P. V. D. Smagt, D. Cremers, and T. Brox, "FlowNet: Learning optical flow with convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2758–2766.

[46] B. Chen, W. Wang, and J. Wang, "Video imagination from a single image with transformation generation," in *Proc. Thematic Workshops ACM Multimedia*, Oct. 2017, pp. 358–366.

[47] J. van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala, "Transformation-based models of video sequences," 2017, *arXiv:1701.08435*.

[48] C. Vondrick and A. Torralba, "Generating the future with adversarial transformers," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1020–1028.

[49] R. Villegas, J. Yang, S. Hong, X. Lin, and H. Lee, "Decomposing motion and content for natural video sequence prediction," 2017, *arXiv:1706.08033*.

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

[51] Y. Zhu and S. Newsam, "Motion-aware feature for improved video anomaly detection," 2019, *arXiv:1907.10211*.

[52] C. He, J. Shao, and J. Sun, "An anomaly-introduced learning method for abnormal event detection," *Multimedia Tools Appl.*, vol. 77, no. 22, pp. 29573–29588, Nov. 2018.

[53] F. Herrera, S. Ventura, R. Bello, C. Cornelis, A. Zafra, D. Sánchez-Tarragó, and S. Vluymans, *Multiple Instance Learning: Foundations and Algorithms*. Cham, Switzerland: Springer, 2016.

[54] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3D convolutional networks," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 4489–4497.

[55] P. Weinzaepfel, X. Martin, and C. Schmid, "Human action localization with sparse spatial supervision," 2016, *arXiv:1605.05197*.

[56] W. S. Noble, "What is a support vector machine?" *Nature Biotechnol.*, vol. 24, no. 12, pp. 1565–1567, 2006.

[57] Y. Yan, C. Xu, D. Cai, and J. J. Corso, "Weakly supervised actor-action segmentation via robust multi-task ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1298–1307.

[58] A. Arnab, C. Sun, A. Nagrani, and C. Schmid, "Uncertainty-aware weakly supervised action detection from untrimmed videos," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 751–768.

[59] P. Mettes, C. G. M. Snoek, and S.-F. Chang, "Localizing actions from video labels and pseudo-annotations," 2017, *arXiv:1707.09143*.

[60] L. Xu, C. Gong, J. Yang, Q. Wu, and L. Yao, "Violent video detection based on MoSIFT feature and sparse coding," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 3538–3542.

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

[62] I. Serrano, O. Deniz, J. L. Espinosa-Aranda, and G. Bueno, "Fight recognition in video using Hough forests and 2D convolutional neural network," *IEEE Trans. Image Process.*, vol. 27, no. 10, pp. 4787–4797, Oct. 2018.

[63] S. Sudhakaran and O. Lanz, "Learning to detect violent videos using convolutional long short-term memory," in *Proc. 14th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Aug. 2017, pp. 1–6.

[64] N. AlDahoul, H. A. Karim, R. Datta, S. Gupta, K. Agrawal, and A. Albunni, "Convolutional neural network–long short term memory based IoT node for violence detection," in *Proc. IEEE Int. Conf. Artif. Intell. Eng. Technol. (IICAIET)*, Sep. 2021, pp. 1–6.

[65] F. U. M. Ullah, K. Muhammad, I. U. Haq, N. Khan, A. A. Heidari, S. W. Baik, and V. de Albuquerque, "AI-assisted edge vision for violence detection in IoT-based industrial surveillance networks," *IEEE Trans. Ind. Informat.*, vol. 18, no. 8, pp. 5359–5370, Aug. 2021.

[66] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*.

[67] R. Vijeikis, V. Raudonis, and G. Dervinis, "Efficient violence detection in surveillance," *Sensors*, vol. 22, no. 6, p. 2216, Mar. 2022.

[68] S. Akti, G. A. Tataroglu, and H. K. Ekenel, "Vision-based fight detection from surveillance cameras," in *Proc. 9th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Nov. 2019, pp. 1–6.

[69] F. Chollet, "Xception: Deep learning with depthwise separable convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 1251–1258.

[70] S. U. Khan, I. U. Haq, S. Rho, S. W. Baik, and M. Y. Lee, "Cover the violence: A novel deep-learning-based approach towards violence-detection in movies," *Appl. Sci.*, vol. 9, no. 22, p. 4963, Nov. 2019.

[71] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, "MobileNets: Efficient convolutional neural networks for mobile vision applications," 2017, *arXiv:1704.04861*.

[72] J. Li, X. Jiang, T. Sun, and K. Xu, "Efficient violence detection using 3D convolutional neural networks," in *Proc. 16th IEEE Int. Conf. Adv. Video Signal Based Surveill. (AVSS)*, Sep. 2019, pp. 1–8.

[73] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

[74] F. J. Rendón-Segador, J. A. Álvarez-García, F. Enríquez, and O. Deniz, "ViolenceNet: Dense multi-head self-attention with bidirectional convolutional LSTM for detecting violence," *Electronics*, vol. 10, no. 13, p. 1601, Jul. 2021.

[75] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. Van Gool, "Temporal 3D ConvNets: New architecture and transfer learning for video classification," 2017, *arXiv:1711.08200*.

[76] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 30, 2017, pp. 1–11.

[77] Y. Su, G. Lin, J. Zhu, and Q. Wu, "Human interaction learning on 3D skeleton point clouds for video violence recognition," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2020, pp. 74–90.

[78] G. Mu, H. Cao, and Q. Jin, "Violent scene detection using convolutional neural networks and deep audio features," in *Proc. Chin. Conf. Pattern Recognit.* Cham, Switzerland: Springer, 2016, pp. 451–463.

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

[80] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015. [Online]. Available: http://www.image-net.org/challenges/LSVRC/2009/

[81] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: Common objects in context," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2014, pp. 740–755.

[82] J. Carreira and A. Zisserman, "A short introduction to the kinetics human action video dataset," 2019, *arXiv:1906.05408*.

[83] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4724–4733.

[84] D. Tran, H. Wang, L. Torresani, J. Ray, Y. LeCun, and M. Paluri, "A closer look at spatiotemporal convolutions for action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 6546–6555.

[85] O. Kopuklu, N. Kose, A. Gunduz, and G. Rigoll, "Resource efficient 3D convolutional neural networks," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshop (ICCVW)*, Oct. 2019, pp. 1–10.

[86] C. Feichtenhofer, "X3D: Expanding architectures for efficient video recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 203–213.

[87] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, M. Suleyman, and A. Zisserman, "The kinetics human action video dataset," 2017, *arXiv:1705.06950*.

[88] E. B. Nievas, O. D. Suarez, G. B. García, and R. Sukthankar, "Violence detection in video using computer vision techniques," in *Proc. Int. Conf. Comput. Anal. Images Patterns*. Cham, Switzerland: Springer, 2011, pp. 332–339.

[89] M. M. Soliman, M. H. Kamal, M. A. El-Massih Nashed, Y. M. Mostafa, B. S. Chawky, and D. Khattab, "Violence recognition from videos using deep learning techniques," in *Proc. 9th Int. Conf. Intell. Comput. Inf. Syst. (ICICIS)*, Dec. 2019, pp. 80–85.

[90] M. Cheng, K. Cai, and M. Li, "RWF-2000: An open large scale video database for violence detection," in *Proc. 25th Int. Conf. Pattern Recognit. (ICPR)*, Jan. 2021, pp. 4183–4190.

[91] *Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the Protection of Natural Persons With Regard to the Processing of Personal Data and on the Free Movement of Such Data, and Repealing Directive 95/46/EC (General Data Protection Regulation) (Text With EEA Relevance)*, document L 119/5, European Commission, 2016.

[92] D. Dwibedi, I. Misra, and M. Hebert, "Cut, paste and learn: Surprisingly easy synthesis for instance detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 1301–1310.

[93] G. Georgakis, A. Mousavian, A. C. Berg, and J. Kosecka, "Synthesizing training data for object detection in indoor scenes," 2017, *arXiv:1702.07836*.

[94] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, "Training deep networks with synthetic data: Bridging the reality gap by domain randomization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2018, pp. 969–977.

[95] G. Varol, J. Romero, X. Martin, N. Mahmood, M. J. Black, I. Laptev, and C. Schmid, "Learning from synthetic humans," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 109–117.

[96] P. Sernani, N. Falcionelli, S. Tomassini, P. Contardo, and A. F. Dragoni, "Deep learning for automatic violence detection: Tests on the AIRTLab dataset," *IEEE Access*, vol. 9, pp. 160580–160595, 2021.

[97] P. Wu, J. Liu, Y. Shi, Y. Sun, F. Shao, Z. Wu, and Z. Yang, "Not only look, but also listen: Learning multimodal violence detection under weak supervision," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2020, pp. 322–339.

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

[99] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *J. Mach. Learn. Res.*, vol. 12, no. 7, pp. 1–39, 2011.

[100] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. 31st Conf. Neural Inf. Process. Syst.*, 2017, pp. 1–4.

[101] *Accuracy (Trueness and Precision) of Measurement Methods and Results*. International Organization for Standardization, Geneva, Switzerland, 1994.

[102] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.

[103] Z. Dong, J. Qin, and Y. Wang, "Multi-stream deep networks for person to person violence detection in videos," in *Proc. Chin. Conf. Pattern Recognit.* Cham, Switzerland: Springer, 2016. pp. 517–531.

[104] D. Choqueluque-Roman and G. Camara-Chavez, "Weakly supervised violence detection in surveillance video," *Sensors*, vol. 22, no. 12, p. 4502, Jun. 2022.

[105] FFmpeg Developers. (2016). *ffmpeg Tool (Version be1d324)*. Accessed: Nov. 17, 2022. [Online]. Available: http://ffmpeg.org/

[106] T. Aremu, L. Zhiyuan, R. Alameeri, and A. El Saddik, "SIViDet: Salient image for efficient weaponized violence detection," 2022, *arXiv:2207.12850*.

VIKTOR DÉNES HUSZÁR is currently a Doctoral Researcher with the Doctoral School of Military Engineering, National University of Public Service. His primary research interests include computer vision and artificial intelligence. He serves as the Chairperson for FITEQ—the governing body of Teqball and the Head for the Digital Committee of FINA—the governing body of aquatics sports. He is an international speaker on computer vision-based technologies. He is a member of the Hungarian Economic Association. He has received several Hungarian and international awards, including the Industrial Innovation Award, the Red Dot Award, and the IF Design Award.

VAMSI KIRAN ADHIKARLA (Member, IEEE) received the double master's degree (M.Tech. and M.Sc.) from Blekinge Tekniska Hgskola, Sweden, and JNTU Hyderabad, India, in 2011, and the Ph.D. degree in information science from the Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, Hungary, in 2015.

From 2015 to 2017, he worked as a Postdoctoral Researcher at the Department of Computer Graphics, Max-Plank-Institute for Informatics. From 2012 to 2015, he worked as a Marie Curie Early Stage Researcher at Holografika, Hungary. He is currently a Researcher with the Faculty of Information Technology and Bionics, Pázmány Péter Catholic University, Hungary. His research interests include real-time computer vision, deep learning for visual analysis, 3-D computer vision, and 3-DTV. In 2021, he received the Marie Skłodowska-Curie Individual Fellowship Award.

IMRE NÉGYESI is an habilitated Associate Professor with the National University of Public Service, where he has been leading the Department of Informatics with the Faculty of Military Science and Officer Training, since 2012. His research interests include possibilities of military applications of artificial intelligence and social and ethical issues related to it. He is a member of the Public Board of the Hungarian Academy of Sciences and a registered Expert of the Hungarian Military Society.

CSABA KRASZNAY is an Associate Professor with the National University of Public Service, where he is currently the Head of the Institute of Cybersecurity. His research interest includes cybersecurity. He received the CISA certification in 2005, CISM and CISSP in 2006, CEH in 2008, ISO 27001 Lead Auditor in 2012, and CSSLP in 2015. He is a Board Member of Voluntary Cyberdefense Cooperation and a member of the ISACA Budapest Chapter, the Magyary Zoltán E-government Association, the Hungarian Association of Military Science, the Scientific Association for Infocommunications, and the Hungarian Association for Electronic Signature. In 2011, he was voted as "Security Expert of the Year."