Abnormal Behavior Detection in Uncrowded Videos with Two-Stream 3D Convolutional Neural Networks

Abid Mehmood

Department of Management Information Systems, College of Business Administration, King Faisal University, Al-Ahsa 31982, Saudi Arabia; aafzal@kfu.edu.sa

Abstract: The increasing demand for surveillance systems has resulted in an unprecedented rise in the volume of video data being generated daily. The volume and frequency of the generation of video streams make it both impractical as well as inefficient to manually monitor them to keep track of abnormal events as they occur infrequently. To alleviate these difficulties through intelligent surveillance systems, several vision-based methods have appeared in the literature to detect abnormal events or behaviors. In this area, convolutional neural networks (CNNs) have also been frequently applied due to their prevalence in the related domain of general action recognition and classification. Although the existing approaches have achieved high detection rates for specific abnormal behaviors, more inclusive methods are expected. This paper presents a CNN-based approach that efficiently detects and classifies if a video involves the abnormal human behaviors of falling, loitering, and violence within uncrowded scenes. The approach implements a two-stream architecture using two separate 3D CNNs to accept a video and an optical flow stream as input to enhance the prediction performance. After applying transfer learning, the model was trained on a specialized dataset corresponding to each abnormal behavior. The experiments have shown that the proposed approach can detect falling, loitering, and violence with an accuracy of up to 99%, 97%, and 98%, respectively. The model achieved state-of-the-art results and outperformed the existing approaches.

Keywords: convolutional neural network (CNN); action recognition; video analysis; abnormal behavior detection; two-stream convolutional neural networks; 3D convolutional neural networks

1. Introduction

Human action recognition in videos is a challenging task that has earned an increasing amount of research attention. Surveillance systems [1], intelligent scene modeling [2], and video annotation and retrieval [3] are some examples of applications of the recognition of human actions in videos. Video surveillance nowadays plays an important role in establishing a climate of safety, trust, and security despite its limitations and concerns with regards to privacy [4,5]. In video surveillance, abnormal behavior detection, which can indeed be viewed as a specific issue of human action recognition, is considered essential to ensure both indoor and outdoor safety. However, factors such as huge amounts of stored data or prolonged periods of its production often lead to a lack of efficiency in its treatment. Indeed, it is both time-consuming as well as laborious for a person to watch surveillance videos for a prolonged period [6]. Furthermore, as abnormal events are relatively infrequent, human oversight task becomes exceedingly overwhelming, and yet inefficient. Consequently, there is a growing need for automated systems for abnormal behavior detection.

Many interesting works aiming to alleviate the difficulties in recognizing abnormal behavior have been proposed in the literature [1,7]. Vision-based approaches have been most successfully applied to a variety of abnormal situations. As regards abnormal human behavior detection, most of the studies have addressed the recognition of one specific behavior in a so-called uncrowded scene [1]. In such scenes, falling and loitering are generally considered abnormal behaviors involving only one person. Falling detection has captured...
the interest of many researchers (e.g., [8–13]) that have proposed systems to ensure the safety of the elderly and people living alone. Another noteworthy topic in the context of uncrowded scenes is loitering detection. Loitering can be defined as the act of being in a specific public space for a given period with no clear objective. Some examples of the works in this category include [14–16]. In contrast, when a few people are involved instead of one person in the uncrowded scene, identifying violent behavior, such as fighting, kicking, and punching between two people, becomes intriguing. Numerous attempts have been made (e.g., [17–20]) to detect violent behaviors in an uncrowded setting. Driven by the growth of deep learning over the past decade, convolutional neural networks have made remarkable achievements in action recognition in videos in general [21–24] and abnormal behavior recognition [25] in particular. In this area, 2D convolutional neural networks (CNNs) combined with a two-stream network architecture are mostly utilized [26,27]. The original two-stream architecture combines an RGB stream for spatial information and an optical stream for temporal information to improve the prediction accuracy. Furthermore, 3D spatiotemporal convolutions have also been found efficient for video action recognition [28,29]. Even though 3D networks require a large number of parameters, their performance has shown significant improvement following the appearance of large-scale video datasets such as Kinetics and MiT [23,24].

Previously, CNNs have been employed to jointly detect different abnormal behaviors. For example, Sha et al. [30] proposed a method to detect “illegal cutting” and “running” behaviors in a specific context of electrical industry. Nevertheless, to the best of our knowledge, no previous study has addressed various kinds of abnormal behaviors involving falls, loitering, and violence, which are generally found in an uncrowded scene. Moreover, most of the studies that addressed abnormality detection have not exploited the full potential of deep learning due to their reliance on models trained on relatively smaller video datasets. It should be noted here that training on a large-scale video dataset tremendously increases the recognition performance, as shown in studies such as [24,31]. Hence, there is a need for an approach that combines the detection of various abnormal behaviors by employing deep networks trained on large-scale video datasets to enhance the identification and classification performance. The second motivation for this work is the unexploited potential of existing pre-trained CNNs developed for related domains that can be utilized to enhance the performance of abnormal behavior recognition. In fact, in previous works, e.g., [31,32], an ImageNet pre-trained 2D CNN [33] has been shown to work effectively for a two-stream 3D model for video processing. In this way, an approach that extends the idea by finetuning a video pre-trained 3D network originally trained for general behavior classification to enable abnormal behavior detection is expected to further improve the recognition performance.

Here, the current work intends to propose an approach that addresses the issues stated above through the joint detection of three abnormal behaviors commonly found in videos of uncrowded scenes. Specifically, the approach adopts a 3D CNN (i.e., Inflated Inception 3D [31]) and uses it within a two-stream architecture [27] to combine video (RGB) and optical flow streams for enhanced action recognition performance. Also, it applies transfer learning by using pre-trained weights on the Kinetics dataset and modifying the model by adding new layers to specialize it for the abnormalities addressed within this study. The model was trained on a customized dataset for the abnormal human behaviors involving falling, loitering, and violence and it efficiently detects and classifies each behavior.

The main contributions of this study are listed in the following.

- The study provides a review of the recent developments to redefine the problem of abnormal human behavior detection and classification, distinctly from human action classification as well as general anomalies detection. To detect each abnormal behavior, its distinct representative patterns were identified.
- Unlike previous studies that focused on one specific behavior, this study proposes a new model that combines the detection of all three abnormal behaviors commonly found in uncrowded scenes. In this way, to the best of our knowledge, this is the first
study that applies transfer learning from the general action recognition domain to the detection of falling, loitering, and violence.

- A specialized dataset was prepared based on publicly available video datasets concerning the common patterns of acts found in behaviors included in the study.
- The study implements a model that makes use of advanced deep learning architectures to simultaneously incorporate both RGB and optical flow information. Moreover, unlike previous works in the domain of abnormal behavior detection that worked on (2D) RGB images extracted from videos, this model takes a 3D video stream as input.

The next section outlines the related studies. Section 3 describes the challenges involved in the detection of three abnormal behaviors. Section 4 presents the approach for recognizing and classifying abnormal behavior. Section 5 provides details of experiments and evaluation, whereas Section 6 concludes the paper while identifying some possible directions for future research.

2. Related Work

This section reviews the main methods that are differently related to the context of this study.

2.1. Traditional Methods

Nguyen et al. [10] have proposed a system for fall detection that is based on rules extracted from shape features. First, they convert the video frames into intensity images in which they detect the largest object and assume it to be the object of interest. Next, they apply a rule-based classification to classify the behavior as abnormal if a change in the shape of the interest object within the image is detected. A similar approach is proposed by Aslan et al. [11] for depth-sensor-based automatic fall detection. To improve the detection of falls in depth videos, they encode Curvature Scale Space (CSS) features with the Fisher Vector (FV) representation. First, they segment the human body and extract the silhouettes which are then used to calculate shape features. The features are then converted into FV representation. Next, they employ a one-vs-all SVM classifier to detect fall actions. Gomez et al. [15] proposed a system to detect the loitering of the elderly by tracking and analyzing their movement patterns. Their system monitors the individuals and applies a Generalized Sequential Patterns (GSP) algorithm to detect predefined patterns of repeated actions that characterize loitering. Huang et al. [14] have proposed an approach that detects loitering based on pedestrian activity area fitting within three categories i.e., rectangle, ellipse, and sector loitering. The type of loitering is determined by trajectory maps and analyzing the suspicious target trajectory. To detect the moving object, they use Gaussian mixture models. If the activity stays constrained in an area for a certain period, they recognize it as loitering. For violence detection, following a Bag of Words (BoW) approach, the method in [18] has used MoSIFT [34] to model the movements of the tracked objects. It then performs a spatio-temporal analysis of the video to detect anomalies.

2.2. Deep Learning-Based Methods

In a broader context, the recent studies in the human action recognition and classification domain extend more classical architectures, such as [27,35,36], to further improve performance. In [21], the authors propose a method to use pre-trained 2D CNNs for both spatial and temporal streams in action classification problems. To do so, instead of using an optical stream, they adopt a pre-trained 2D CNN for the temporal stream. They form a Stacked Grayscale 3-channel Image (SG3I) by selecting three RGB frames at arbitrary points in the video. SG3Is are then used to fine-tune the pre-trained 2D CNN. [37] proposes a Combined Video-stream Deep Network that extracts the spatio-temporal features from video clips using a ResNet. They apply ResNets models pre-trained in Kinetics dataset to UCF-101 and extract video stream features. Next, similar to our approach, they use optical flow graphs of the UCF-101 dataset as input to optical stream and extract the optical features therein. Finally, features from both streams are combined. A different approach to
human action recognition is proposed in [38]. This study is centered around the distance transform and entropy features extracted from images of human silhouettes obtained after background subtraction. As these features provide the shape and local variation information, they are input to deep networks to recognize human actions.

Another related line of research addresses anomaly detection in video surveillance. In [39], a method has been proposed for surveillance based on Unmanned Aerial Vehicles (UAVs). To extract features from UAV videos, they utilized a 3-way process consisting of the use of a CNN, Histogram of Oriented Gradient (HOG), and HOG3D. Finally, a one-class support vector machine has been applied for classification. Recently, Castellano et al. [40] proposed the use of a particular scheme of deep learning using a fully convolutional neural network (FCN) as a regressor for crowd counting applied to aerial scenes shot from UAVs. They train two FCNs simultaneously on the captured images of the crowd as well as the corresponding crowd heatmaps. Similarly, in [41], taking into account the limited computational capabilities of UAVs, a lightweight crowd detection method was proposed to ensure the safe landing of UAVs. An essentially different solution has been proposed in [42]. This study exploits the one-shot learning strategy for anomaly recognition to develop a method for one-shot anomaly detection for surveillance systems. Their method is based on a lightweight siamese 3D CNN that efficiently determines the similarity between two anomaly sequences. In [6], the spatio-temporal features are extracted by utilizing a pre-trained ResNet-50 architecture. Then, the extracted features are passed to a multi-layer Bi-directional Long Short-term Memory (BDLSTM) model to classify anomalies in surveillance videos. Sha et al. [30] proposed a method to detect five different behaviors (of which two were considered abnormal) in a specific industrial setting. They used DenseNet in a two-stream architecture to extract spatial and temporal features and applied a particular technique to alleviate the problems resulting from imbalanced data. The model was trained and tested on a self-constructed dataset. Ref. [43] used an ImageNet pretrained VGG16 architecture and InceptionV3 to finetune on UCF-Crime dataset [44]. In the specific context of the current study, some approaches aim at detecting specific abnormal behaviors. Nunez et al. [8] propose an approach for fall detection that uses CNN to learn the detection of falls from optical flow images. They train the VGG-16 network on ImageNet and optical stacks of UCF-101 [45]. Finally, they apply transfer learning and fine-tune the network on fall datasets. This approach is similar to ours, but our 3D architecture allows more effective video processing. In contrast with Nunez et al.'s work, instead of using the modified VGG-16 architecture, Wang et al. [46] extract features from color images using PCAnet [47] and then apply an SVM to detect falls. In [48], a vision component is first used to extract frames of moving people from videos, and then a combination of histograms, local binary patterns, and features extracted by Caffe [49] are used to recognize a silhouette. Finally, two SVM classifiers are used to detect falls. In a similar study, Yao et al. [12] proposed a fall system that uses geometric features to train a CNN. They first segment the head and the torso by the traditional ellipse fitting method and extract motion features. A shallow CNN structure is then used to learn these features and achieve high accuracy on the authors’ self-collected dataset. In contrast with these approaches, instead of using feature engineering, the current work relies on features learned by CNN. Khraief et al. [13] employed a CNN architecture composed of four streams to detect falls using multimodal data captured by RGB-D cameras. They used a combination of RGB and depth images along with three other modalities to deal distinctly with static appearance, shape variations and motion information.

In [16], a framework for the detection of multiple events including loitering, intrusion, and unattended object has been presented. This work uses a set of variables corresponding to various characteristics of different surveillance scenarios. These variables provide a knowledge-based understanding of the environment which is then manipulated to recognize the activity patterns within the scene. Ding et al. [17] use a 9-layers 3D CNN to detect fights. Even though they use 3D convolutions, their work adopts 2D pooling which results in losing the temporal information from the input. Asad et al. [20] proposed...
a violence detection method based on CNNs and recurrent neural networks to learn appearance and motion features from fused feature maps. They used a particular residual block to learn combined spatial features from sequential frames within a video. The features are finally concatenated and fed into LSTM units to obtain temporal dependencies.

3. Background: Challenges in Detecting Three Behaviors

As mentioned above, our system deals with commonly found abnormal behaviors to provide flexibility of detection in different environments (indoor and outdoor) to enhance the level of scene understanding with minimal fine-tuning. Therefore, to develop a system that classifies abnormal behaviors with high accuracy, two main types of challenges must be recognized and overcome. The first type of difficulty is related to the variance in specific movement patterns entailed in each behavior. So, the distinctive movement patterns involved in each act must be specified in detail and ample examples of each kind must be provided to the classification system so that it learns features related to each behavior correctly. The second challenge relates to the essential similarity in the acts of different behaviors leading to wrong predictions. Though such problems are inherent in the joint detection of various behaviors, they must be alleviated as much as possible by specifying the overlaps and providing examples that enable the system to distinguish features in apparently similar activities. In the following, each behavior is defined by specifying the representative movement patterns involved in it followed by the definition of overlapping patterns.

3.1. Falling

The falls have been categorized into three major types based on the position of the actor in the literature [50] i.e., standing, sitting, and lying down. In each position, the fall may occur in a backward, forward, or sideward direction. The key possibilities of each of these cases include:

i. A person is found walking in a scene and then destabilizing in backward, forward, or sideward direction and later found on the floor.

ii. A person initially static in a horizontal position and then exhibiting a slow backward, forward, or sideward fall eventually ending in a resting position on the floor.

iii. A person initially found in a horizontal position exhibiting an event such as slipping, tripping, stumbling, losing balance, and eventually falling given that the movement speed is fast.

iv. A person found in any of the above compounded with the acts of balancing attempts such as lifting arms or hands and holding nearby objects or walls.

3.2. Loitering

Loitering detection is critical as it can help to identify the cases where (i) vulnerable people such as older patients with Alzheimer’s disease need attention, and (ii) potential suspects involved in illegal activity are noticed [14]. Different studies (e.g., [14, 16]) have defined the act of loitering based on two main characteristics. First, loitering is the act of lingering in a certain area for more than a specified time. Note that the value for the time span must be tunable as the threshold value would differ from one application to another. So, for example, it will be shorter in a critical area such as an authorized building as compared to the corridor of a shopping mall. Second, whereas normal pedestrians typically walk in a straight line with an aim, loitering is characterized by the random and arbitrary movement of an individual. Here, various possibilities about each of these cases are defined. Along the time span dimension, the possibilities are:

i. A person is detected in a scene and stays in the scene for a specific duration.

ii. A person is found in a scene and is moving at a slow speed for a specific duration.

Similarly, concerning movement type, Huang et al. [14] have found loitering to be a case of movement that forms a certain motion trajectory within a specific activity area. In this way, the following cases are possible:
i. A person is detected in a scene and moves within an activity area that forms a regular shape such as a rectangle or ellipse.

ii. The person continues to move around a point of interest thus resulting in an activity area that revolves around one point.

3.3. Violence

Concerning violence, the current study is focused on the violent behavior in uncrowded scenes involving two persons only such as fighting, kicking, and punching of two people. In this way, violence detection involves challenges due to two main factors: diversity of the fight patterns and variety of appearance. The typical patterns in fights include punching, kicking, jabbing, hitting with an object, and clinching. Hockey fights can be a useful source of information to learn each of these diverse patterns as most of them are explicitly found therein. In this way, the system must be provided with exclusive examples containing reliable cues of each pattern. However, as the sports footage is usually consistent in appearance, the other challenge here is to ensure that the system has obtained adequate knowledge of patterns that enables it to generalize and translate the learned features to other scenarios. This must be addressed by ensuring that fight scenes with a varied context are provided.

3.4. Joint Detection of Falling, Loitering, and Violence

The difficulties in the combined detection consist in the possibility of the system mistaking one abnormal activity with another due to their characteristic similarity in some ways. In the following, some cases involving such confusions are described.

i. Loitering and falling: Loitering without intent may involve lifting of arms, touching nearby objects or walls, crouching down, bending down, and lying on the floor. On the other hand, these acts are typical characteristics of the visuals of a person falling.

ii. Violence and falling: Crouching down, bending down, destabilized posture, posing with raised arms in an attempt to balance, and arms lifted overhead are often found in scenes containing both behaviors.

iii. Violence and loitering: Holding an object, lifting of arms, touching nearby objects or walls, crouching down, bending down, and lying on the floor are common in both cases.

Nevertheless, it should be noted that the above similarities are not accidental. Indeed, they are inherent to the nature of the problem.

4. The Proposed Approach

This work proposes an approach that accurately detects three abnormal behaviors in videos. The overall design of this approach has three main characteristics. First, based on the intuition of each studied behavior presented in the previous section, a video dataset is carefully prepared by selecting videos from public datasets. These videos are then preprocessed to be converted into a suitable shape for the input to a CNN. Second, the approach uses a two-stream CNN architecture that allows us to use a combination of video (RGB) and optical flow streams to enhance the recognition performance. Third, due to the low number of examples found in public video sets related to abnormal behavior detection, transfer learning is applied that benefits from huge video datasets to specialize in abnormal behavior detection. So, the pretrained network is finetuned on the newly developed dataset to identify and classify an abnormal behavior. Specifically, each stream of the network provides a classification score which is finally fused to determine the final prediction. The overall architecture of the approach is shown in Figure 1. The following subsections provide a detailed discussion of the various components and steps involved in the development of the system.
4.1. Dataset Preparation

This study prepared a specialized dataset named UFLV-DS (Uncrowded scene Falling, Loitering, Violence Dataset), based on videos from six public datasets. More precisely, it used (URFDS: http://fenix.univ.rzeszow.pl/~mkepski/ds/uf.html (accessed on 15 January 2021)) UR Fall Dataset (URF-DS) and the (MulticamDS: http://www.iro.umontreal.ca/~labimage/Dataset/ (accessed on 15 January 2021)) Multiple Camera Fall Dataset (Multicam-DS) for fall detection, the dataset of (http://home-pages.inf.ed.ac.uk/rbf/CAVIAR (accessed on 15 January 2021)) CAVIAR Project (Caviar-DS) and UMN (http://mha.cs.umn.edu/proj_events.shtml (accessed on 15 January 2021)) (UMN-DS) for loitering, and Hockey Fight Dataset (Hockey-DS) [18] and Movies Dataset (Movies-DS) [18] for violence detection. The study intended to create a specialized dataset more aligned with its objectives. Therefore, the choice of datasets was driven by some requirements such as: (i) the videos should mainly contain uncrowded scenes with minimum variation in lighting conditions to enable focus on the behavior only, (ii) the main actions (or their negatives) pertaining to abnormal behaviors to be classified in the study must clearly be displayed, (iii) videos should contain as many as possible examples of the specifics of each behavior described in Section 3, and (iv) the dataset should be publicly available and other works in the literature should have used it for recognition of similar behaviors.

The UFLV-DS was finalized in two stages. Initially, a basic setup was created by randomly selecting positive and negative examples of each class given in the original dataset. In this stage, no specific criteria were applied for the selection of videos. Later, each example was manually labeled into one of four classes: fall, loitering, violence, and none. Note that the “none” class corresponded to negative examples taken from each class. A sample of frames taken from videos of positive and negative examples of each class is shown in Figure 2. The total number of positive and negative examples of each action class is as follows. The falling class has 40 positive and 27 negative examples. The loitering class has 35 positive and 23 negative examples. Similarly, there are 42 examples of positive and 28 examples of negative actions for violence. In this way, the total of all negative examples gives us the number of positive examples of the “none” class i.e., 78. Further, the examples were divided into training, validation, and testing datasets by specifying a split property for each example. Also, video frames were preprocessed to obtain a size of $224 \times 224$ for input to the model as detailed in the later sections.
The UFLV-DS was finalized in two stages. Initially, a basic setup was created by randomly selecting positive and negative examples of each class given in the original dataset. In this stage, no specific criteria were applied for the selection of videos. Later, each example was manually labeled into one of four classes: fall, loitering, violence, and none. Note that the "none" class corresponded to negative examples taken from each class. A sample of frames taken from videos of positive and negative examples of each class is shown in Figure 2. The total number of positive and negative examples of each action class is as follows. The falling class has 40 positive and 27 negative examples. The loitering class has 35 positive and 23 negative examples. Similarly, there are 42 examples of positive and 28 examples of negative actions for violence.

Further, the examples were divided into training, validation, and testing datasets by specifying a split property for each example. Also, video frames were preprocessed to obtain a size of $224 \times 224$ for input to the model as detailed in the later sections.

Figure 2. Sample of frame sequences taken from videos containing negative and positive examples of each action class.

4.2. Optical Stream Construction

As mentioned previously, the model uses the optical stream to detect abnormal behavior because many works in the literature (e.g., [26,31,51]) have suggested that the information from the optical stream is so indispensable that it alone can distinguish between most actions in large datasets. The optical stream is created by stacking $2L$ optical flow images from the datasets (each image $I \in \mathbb{R}^{w \times h \times l}$ being the correlation between two consecutive images) representing the motion pattern across the stacked frames. The approach stacks the horizontal $d^x_t$ and vertical $d^y_t$ components of the displacement vector fields to create stacks $O \in \mathbb{R}^{224 \times 224 \times 2 \times L}$ where $O = \{d^x_1, d^y_1, d^x_{1+1}, d^y_{1+1}, \ldots, d^x_{L+1}, d^y_{L+1}\}$.

The optical flow images particularly work well in abnormal behavior detection. These are appropriate to model short events and only take into account the motion information while discarding other irrelevant static information such as the background. As the lighting conditions in our datasets are stable, the problem of obtaining undesirable displacement vectors (due to lighting changes) is alleviated to some extent. To address this issue further, the TVL-1 optical flow algorithm [52] was chosen due to its strength in changing light conditions. Specifically, the publicly available software tool (https://github.com/feichtenhofer/gpu_flow (accessed on 19 January 2021)) provided by Feichtenhofer et al. [26] was used to obtain optical flow. A sample of sequential frames from the URFD dataset pertaining to falling action is shown in the first two rows of Figure 3, whereas their corresponding optical flow is shown in the following rows.
4.3. System Architecture and Flow

When it comes to choosing the architecture for video representation, there is no clear front-runner. Some of the major current video architectures can be classified based on types of kernels (2D or 3D), input (RGB video or optical flow), and the information propagation across frames (LSTMs or feature aggregation over time). Different network architectures have been compared thoroughly in the literature [23,31]. A brief comparison of a subset of major models based on these classification schemes is shown in Table 1.

This work adopted the two-stream Inflated 3D architecture (i3D) [31] due to its outstanding performance in video datasets i.e., UCF-101, HMDB-51 [53], and Kinetics [54]. This architecture has outperformed the other models in training and testing on Kinetics with and without ImageNet pretraining [31]. The i3D architecture essentially inflates Inception v1 [35] to use 3D filters and pooling kernels. In this regard, the Inception architecture has two main strengths. First, the model architecture size is much smaller (despite increasing the depth) compared to others such as AlexNet and VGGNet due to the use of global average pooling instead of fully connected layers. The resulting massive savings in memory are important for small real-time applications such as abnormal behavior detection. Second, it uses a concept of a microarchitecture (network in the network) when constructing the macro architecture. More specifically, the Inception module of this architecture serves as a building block that fits into a CNN enabling it to learn convolutional layers with multiple filter sizes that are computed in parallel, and the resulting feature maps are concatenated along the channel dimension. So, the network can effectively learn the local features via smaller convolutions and more abstract features with larger convolutions. This makes the module a multi-level feature extractor.

Furthermore, to enhance recognition performance, the model used a combination of video (RGB) and optical flow streams. Two-stream methods of behavior recognition traditionally use the RGB stream and optical stream to extract spatial and temporal data, respectively and then combine the information from both streams to predict behavior. Within each stream, i3D architecture was used that essentially converts the (2D) Inception module and the architecture into a 3D CNN. It inflates all filters and pooling kernels by adding a temporal dimension. So, the $N \times N \times N$ filters become $N \times N \times N$. The resulting inflated Inception module and the Inflated Inception v1 architecture are shown in Figure 4. In our implementation, the first i3D trains the video stream whereas the second is used to train the optical flow stream.

---

Figure 3. (a) Sample frames sequence pertaining to falling from URFDS; (b) their corresponding optical flow horizontal displacement; (c) optical flow vertical displacement.
Two-Stream Nets \[27,61\] Use of optical flow along with RGB frames gives high performance, low no. of parameters

Less test time augmentation due to snapshot sampling

Model Representative Works | Pros | Cons
--- | --- | ---
2D CNN/ CNN + LSTM \[55–58\] | Reuse image classification networks, Low no. of parameters | Tend to ignore temporal structure, large training footprint (expensive), may not capture low-level variations

3D CNNs \[28,29,59,60\] | Directly represent spatiotemporal data | High no. of parameters

Original Two-Stream Nets \[27,61\] | Enhanced use of both RGB and flow inputs, relatively low no. of parameters | Optical flow frames add computational load, high memory requirement

3D-Fused Two-Stream Nets \[26,37,62,63\] | Less test time augmentation due to snapshot sampling | Computation load, high memory requirement

Two-Stream Inflated 3D Nets \[31\] | | Enhanced use of both RGB and flow inputs, relatively low no. of parameters

Table 1. Comparison of Video Representation Architecture Models.

The model was implemented by using the Keras framework (precisely tf.keras with TensorFlow 2.0). Figure 5 shows a detailed view of our implementation of the architecture. Note that in our implementation, the batch normalization \[33\] and activation functions follow each 3D convolutional layer except for the last convolutional layer (classification block), as proposed in the original architecture \[31\]. Similarly, the concatenation of layers is carried out followed by each inception module.
block), as proposed in the original architecture [31]. Similarly, the concatenation of layers is carried out followed by each inception module.

Figure 5. Implementation details of the model. Note that the architecture of each Conv3d layer shows the number of filters, size of the convolutional kernel, and stride. The architecture of each pooling layer includes the kernel size and stride.

As evident from the dataset details discussed previously, the UFLV-DS is a small-sized dataset. Even though the study adopted several methods during experiments to avoid over-fitting due to the small dataset, the initial results for training and testing on these datasets alone were still not satisfactory. Therefore, to make up for the disadvantage of an insufficient dataset, the training model was initialized with the i3D model pre-trained on the Kinetics dataset, which is considered as a de-facto standard for video action recognition. Indeed, a 2-step approach to applying transfer learning was followed. The approach is described in the following section.

4.4. Model Training and Testing

In this section, the details of the training and testing of the proposed model on the dataset are discussed.

4.4.1. Weights Initialization

As part of the pretraining step, our network was initialized with the publicly available pre-trained weights provided by Carreira and Zisserman [31] for the i3D network trained in ImageNet and Kinetics datasets for both RGB and optical flow. The weights were initialized until the last set of 3D convolutional (i.e., Conv3d_5c_3b, see Figure 5) layer.

4.4.2. Finetuning on UFLV-DS Dataset

For finetuning the model, the classification block comprising the last set of average pooling, dropout, 3D convolutional, reshape, and softmax layers was removed from i3D and the newly initialized classifier layer was retrained on UFLV-DS dataset. During this step, the weights for all previous convolutional layers were kept frozen during iterations 1–30. After the 30th iteration, the stalled layers were unfrozen to enable the network to
make weight adjustments to the trained layers for UFLV-DS. Next, the softmax classifier was trained for the video and optical flow streams separately and then the results from the two streams were fused. Specifically, for the video stream, the model was trained using frame sequences of 64 frames each obtained after the segmentation of videos of the dataset. The size of each clip of the frame sequence was $224 \times 224$. These clips were used as input of the i3D network with 3 channels to learn video features related to abnormal behaviors addressed in this study. At this stage, data augmentation techniques were also applied to the input of the RGB stream as discussed later in this section. The second stream used the optical graphs (based on $L = 10$ horizontal and vertical optical flow overlay frames in a stack of size 20, following [27]) from the UFLV-DS dataset as input in order to learn motion features specific to falling, loitering, and violence. So, within this stream, i3D with 2 channels takes optical flow frames of size $224 \times 224$ as input.

The output layer contains a softmax with four possible outputs representing each of the classes i.e., fall, loitering, violence, and none. Specifically, the new classification block consists of the 3D average pooling layer (i.e., global_avg_pool of (2,7,7) with a stride (1,1,1)) followed by a dropout layer, a 3D convolutional layer ($1 \times 1 \times 1$), and reshape layers. Finally, a softmax layer provides the normalized class probabilities. As regards the fusion technique, there are two main approaches. In the first approach (referred to as early fusion or feature fusion), feature vectors obtained from different streams are merged and passed to one classifier to obtain the final score. In the second approach (referred to as late fusion or score fusion), output probabilities from separate classifiers of each stream are combined to get the final score. Further in this regard, different possibilities have been comprehensively investigated by Feichtenhofer et al. [53] in terms of “where to fuse” and “how to fuse”. They found that late fusion simply by applying a summation of the features was an effective fusion technique. This work applied the same method and calculated the sum of the weighted score from each stream to get the final prediction.

Moreover, while fine-tuning in the UFLV-DS, some data augmentation techniques were applied. First, during training, our system used random cropping both spatially and temporally similar to [31]. Precisely, for the spatial cropping, the system resized the smaller video to 256 pixels and then randomly cropped a $224 \times 224$ patch. In contrast, random temporal cropping was applied while picking the starting frame. On the other hand, during testing, the model took center crops of size $224 \times 224$ and applied the models convolutionally over the entire length of the video.

Furthermore, this study tried to deal with the problem of an imbalanced dataset. The main challenge here was to handle a situation of conflicting requirements. On the one hand, since a judicious balance between the positive and negative examples of each class resulted in about 3 times more examples for the “none” class (resulting in high bias towards none), it was appropriate to reduce the number of negative examples. On the other hand, sufficient negative examples for each class were required to ensure that system is learning the class features accurately. Therefore, our approach tried to reduce the negative examples to a feasible extent by keeping the examples with more elaborate (negative) features (as detailed in Section 4.1). However, even after balancing the dataset, learning of the abnormal behavior seemed to be difficult. This was mainly due to the inherent problem described above. Thus, as expected, the model still seemed more inclined towards predicting everything as the majority class i.e., none. It is noteworthy that this problem underlines the main purpose of including the “none” class. It allowed us to monitor and improve the performance of the model against the specific negative examples of each of the abnormal behavior classes.

Therefore, we used the cross-entropy loss function $\mathcal{L}$ for the softmax defined as:

$$\mathcal{L}(y, t) = - \sum_{c=1}^{C} t_c \log(y_c)$$  \hspace{1cm} (1)$$

where C-dimensional vector $y$ is a vector of real values between 0 and 1 representing the prediction of the network and $t$ is a C-dimensional vector of ground truth values. In this
equation, the importance of one class can be increased by introducing a weight within the loss function. Therefore, our updated loss function $L$ is given by:

$$L(y, t) = - \sum_{c=1}^{C} w_c t_c \log(y_c)$$  

(2)

where $w$ contains the weights associated with each class, $y$ is the prediction and $t$ is the ground truth.

In this way, a class weight enables us to use different weights for each class. This means that a class weight of 1.0 can be used if no change in the weighting of the class is required. Whereas, by using a higher weight for a class, the loss function will be penalized for every mistake on that class more than a mistake on the other classes. In other words, by increasing their weight, the network prioritizes the learning of certain classes (abnormal behavior in this case) possibly at the expense of worsening the learning for the other class (i.e., none). However, the resulting bias can be considered acceptable because of the importance of detecting an abnormal event even at the expense of some false alarms. The current implementation estimated the class weights by using `compute_class_weight` in scikit_learn and used the parameter `class_weight` while training the model.

5. Experimental Works

Several experiments were designed and performed to evaluate and prove the performance of the proposed solution for abnormal behavior recognition and classification system. During experimentation, a sequence of 64 frames (each prepared as $224 \times 224$ pixel sequence without changing the aspect ratio) obtained from the videos were used as input to the video stream. In the case of shorter clips, the video was looped as many times as required to fulfill the requirements. Therefore, for example, to meet the definition of loitering, the shorter clips in the dataset were looped several times. Later, optical graphs were extracted from the UFLV-DS videos and used as input to the optical stream as described previously.

The proposed model is implemented in Python (keras with TensorFlow 2.0). The training and experiments were conducted using the same environment on Ubuntu 20.04 operating system, with NVIDIA GTX 1080Ti (11 GB) GPU.

5.1. Evaluation Methodology

As accuracy is not generally considered the best measure of the performance of a model, the study employed two main metrics i.e., precision and recall, which are not biased by imbalanced class distributions. However, accuracy was also calculated for the sake of comparison with existing approaches. It is important to note here that the metrics were applied for each class separately. In this way, the case of abnormal behavior (three classes) versus none was not considered as it will be inconsequential to mark a case as true positive where, for example, fall has been predicted as loitering (note both are abnormal events).

The metrics are briefly described in the following.

**Precision:** Precision is the measure of the proportion of positive identifications of a class that was actually correct. In other words, it measures what proportion of the predictions made for a class (fall, loitering, violence, none) was correct. Precision $P$ is defined as follows:

$$P = \frac{TP}{TP + FP}$$  

(3)

where $TP$ and $FP$ refer to true positives and false positives, respectively.

**Recall:** It is also known as sensitivity. It is a measure of the proportion of actual positives that were identified correctly. In our case, recall measures what proportion of the behavior labeled as one of the four classes was correctly predicted. Recall $R$ is defined as:

$$R = \frac{TP}{TP + FN}$$  

(4)
where TP and FN refer to true positives and false negatives, respectively.

**Accuracy:** The accuracy $A$ of the model can be calculated as follows:

$$A = \frac{TP + TN}{TP + TN + FP + FN}$$  \hspace{1cm} (5)

where $TP$, $TN$, $FP$, and $FN$ are true positives, true negatives, false positives, and false negatives, respectively.

**$F_1$ Score:** The $F_1$ score is the most used metric of a more general parametric class known as the F-measures. It is defined as the harmonic mean of precision and recall and can be calculated as given below:

$$F_1 = \frac{2 \cdot TP}{2 \cdot TP + FP + FN} = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$  \hspace{1cm} (6)

where $TP$, $FP$, and $FN$ are true positives, false positives, and false negatives, respectively, and precision and recall are the metrics defined above.

### 5.2. Determining the Best Network Configuration and Setup

The first set of experiments was carried out to find the best network configuration. Specifically, the study analyzed the model performance for different optimizers as well as various values of the learning rate, momentum, and class weight. As recommended by the authors of the Inception network [35], the experiments started off by using the stochastic gradient descent (SGD) with a learning rate of $10^{-2}$, the momentum of 0.9, and an L2 weight decay of 0.0002. After experimenting with a few choices for the learning rate with the SGD optimizer, it was replaced for Adam with an initial learning rate of $10^{-3}$. In this experiment, a scheduled learning rate decay was applied by using learning rates of $10^{-3}$, $10^{-4}$, and $10^{-5}$ in epochs 1–30, 31–50, and 50–70, respectively. As this setting yielded the best results, the approach used the same model on the testing set.

So far as the class weights are concerned, the weights $w_1$, $w_2$, $w_3$ are important to increment the importance of the three classes of abnormal behavior, because the none class (as reflected in the results of precision and recall values) was found to be learned better than the other classes. However, if one considers all three classes of abnormal behavior versus the none class, the recall of the model is particularly important. Specifically, some false positives are more acceptable than false negatives. Hence, by adjusting the class weights, the system tried to reach a situation where the model is performing better for the three classes of abnormal behavior even at cost of having some false alarms. In this case, a value higher than 1.0 for $w_1$, $w_2$, $w_3$ improved results. Therefore, the base values for these weights were set to 2.0 and incremented or decremented based on results. In this way, the most promising results for one class (i.e., loitering) were achieved with a value slightly higher than 2.0 but if a standard value for all three was to be selected, a value of 2.0 is deemed adequate in general.

As previously mentioned in Section 4.1, some initial trials of the approach were based on a basic setup that included randomly selected videos from the datasets. This means that specific movement patterns of each behavior were not taken into account. Also, videos from only one dataset were used in training for loitering (Caviar-DS) and violence (Hockey-DS). Moreover, this basic experimental setup tried to solve the problem of unbalanced datasets by applying Synthetic Minority Over-sampling Technique (SMOTE) [64] to generate synthetic samples from the minority class. Later, the study enhanced the dataset as previously described and obtained the best configuration setup results.

### 5.3. Analysis of Detection Errors

Before presenting the results of overall system performance (next section), this section will discuss the cases of detection errors and the underlying causes. The analysis of detection errors was performed to ascertain that insight into specifics of each abnormal behavior and the resulting continuous improvement has led the system to achieve the
required level of performance with the final setup and that most remaining errors are due to essential rather than accidental factors. In other words, this analysis was performed to demonstrate that in nearly all cases of detection errors, the activity in question was generally indistinguishable from the predicted abnormal behavior. For this purpose, a sample set was created comprising 90 sequences from all the sequences with errors; 30 samples for each of the false alarms, missed detections, and misclassifications discussed in the following.

5.3.1. False Alarms

In the specific case of abnormal activity recognition, one can consider it a false alarm (false positive) when a stack of frames that has been labeled as “none” is predicted as one of the abnormal behaviors i.e., falling, loitering, or violence. Note that it is different from a misclassification in general, which in this case would mean, for instance, a falling predicted as violence. Misclassifications are discussed later in this section. A careful analysis of samples found the following common sources of confusion leading to false alarms in a majority of cases (see Table 2). It is important to note that the percentages given in the table only show the percentage of each case relative to the total number of false alarms made, and they do not reflect the number of false alarms made for a class.

Table 2. Percentage of different cases of false alarms.

| False Alarm Cases                | % Cases Relative to Total False Alarms |
|----------------------------------|---------------------------------------|
| None predicted as Fall           | 57.40                                 |
| nf1                              | 23.40                                 |
| nf2                              | 16.00                                 |
| nf3                              | 11.40                                 |
| nf4                              | 6.60                                  |
| None predicted as Violence       | 22.90                                 |
| nv1                              | 10.90                                 |
| nv2                              | 6.40                                  |
| nv3                              | 5.60                                  |
| None predicted as Loitering      | 17.80                                 |
| nl1                              | 7.50                                  |
| nl2                              | 5.60                                  |
| nl3                              | 4.70                                  |
| Misc. cases                      | 1.90                                  |

It was noticed that most of the false alarms (57.40%) were generated for falling. One possible reason for this may be that falling contains most of the patterns that are in some way similar to those used in videos for negative examples of each class. Specifically, the following cases belonging to “none” were predicted as falling.

- **nf1**: the person in the scene is slowly bending down until he reaches the floor and then lies down.
- **nf2**: the person in the scene bends down while sitting to pick an object from the floor.
- **nf3**: the actor enters or exits the scene walking slowly but unsteadily.
- **nf4**: the person in the scene is lying down.

A total of 22.90% false positives were generated for violence. Specifically, the following cases containing no abnormal activity were predicted as fighting.

- **nv1**: two or more players are seen close to each other while at least one is lifting an arm.
- **nv2**: two or more players are close to each other and at least one bends down.
nv3: several actors in the scene are performing normal acts involved in playing hockey such as running.

Loitering received the minimum number of false alarms (17.80%). All cases in this category are related to the system erroneously ignoring the time perspective. Precisely, the following cases containing no abnormal activity were predicted as loitering.

nl1: the person enters the scene walking steadily towards the exit, returns towards the entrance, and leaves the scene.
nl2: the actor is working out.
nl3: the actor is standing idly.

In addition to the above, some other rare cases of false alarm (various cases making a total of 1.90%) were also noted but they were insignificant and appeared only in a few stacks for each abnormal behavior.

5.3.2. Missed Detections

In the context of this study, a missed detection (false negative) refers to cases in which a stack of frames labeled as one of the abnormal behaviors i.e., falling, loitering, or violence is predicted as “none”. As stated above, it is different from misclassification which is discussed in the next subsection. On analysis of samples, the following common sources of confusions leading to missed detections were found (see Table 3). It is important to note that the percentages given in the table only show the percentage of each case relative to the total number of missed detections, and they do not reflect the number of detections missed for a class.

| Missed Detection Cases       | % Cases Relative to Total Missed Detections |
|------------------------------|---------------------------------------------|
| Fall predicted as None       | 14.71                                       |
| fn1                          | 7.14                                        |
| fn2                          | 4.62                                        |
| fn3                          | 2.94                                        |
| Loitering predicted as None  | 62.18                                       |
| ln1                          | 31.09                                       |
| ln2                          | 21.43                                       |
| ln3                          | 9.66                                        |
| Violence predicted as None   | 19.33                                       |
| vn1                          | 8.82                                        |
| vn2                          | 5.88                                        |
| vn3                          | 4.62                                        |
| Misc. cases                  | 3.78                                        |

Fall detection received the minimum false negatives (14.71%). Specifically, the following cases containing a fall were predicted as “none”.

fn1: the person tries not to fall while swinging during the walk before falling.

fn2: 4.62% of the cases do not contain anything special. The system is possibly not learning a specific feature correctly to detect just a normal fall.

fn3: the actor falls while trying to grab a nearby object (chair or table).

Most of the missed detection cases were related to loitering (62.18%). This may be due to the fact that loitering contains the most indistinguishable features among the three abnormal behaviors which are mostly found in examples of none. The specific sources of errors observed for the missed detection of loitering are given in the following.
ln1: 31.09% of cases do not contain anything special during loitering. These are just the normal cases of loitering that the system is unable to detect possibly because of learning some features incorrectly.

ln2: the scene contains many people, and one person is walking forward and backward repeatedly in the corridor. It can be hypothesized that the presence of several people (crowded scene) is resulting in the error.

ln3: the actor is sitting cross-legged next to the wall outside a shop.

A total of 19.33% of cases of violence predicted as “none” contained the following common sources of errors.

vn1: two players are fighting in a relatively distant scene; thus, the error may be related to the unclarity of features.

vn2: several players are engaging in the fight.

vn3: one of the two fighting players falls down.

In addition to the above, some other rare cases of false negatives (various cases making a total of 3.78%) were also noted but they were insignificant and appeared only in a few stacks for each abnormal behavior.

5.3.3. Misclassifications

This section discusses the cases where the confusion due to the inherent problem of joint detection discussed in Section 3.4 resulted in classification errors. On analysis of samples, for a majority of cases, it was possible to identify the common sources of confusion as follows (see Table 4). It is important to mention that the percentages given in the table only show the percentage of each case relative to the total number of misclassifications and not the percentage of misclassifications done by the system. A total of 2.24% cases of fall predicted as loitering involved the following common sources of errors.

Table 4. Percentage of different cases of misclassifications.

| Misclassification Cases          | % Cases Relative to Total Misclassifications |
|---------------------------------|---------------------------------------------|
| Fall predicted as Loitering     | 2.24                                        |
| fl1                             | 1.40                                        |
| fl2                             | 0.84                                        |
| Fall predicted as Violence      | 19.33                                       |
| fv1                             | 9.80                                        |
| fv2                             | 6.16                                        |
| fv3                             | 3.36                                        |
| Loitering predicted as Fall     | 6.72                                        |
| lf1                             | 4.20                                        |
| lf2                             | 2.52                                        |
| Loitering predicted as Violence | 36.69                                       |
| lv1                             | 18.21                                       |
| lv2                             | 12.04                                       |
| lv3                             | 6.44                                        |
Table 4. Cont.

| Misclassification Cases       | % Cases Relative to Total Misclassifications |
|-------------------------------|---------------------------------------------|
| Violence predicted as Fall    | 20.45                                       |
| vf1                           | 8.68                                        |
| vf2                           | 7.28                                        |
| vf3                           | 4.48                                        |
| Violence predicted as Loitering| 12.89                                       |
| vl1                           | 8.12                                        |
| vl2                           | 4.76                                        |
| Misc. cases                   | 1.68                                        |

fl1: 1.40% of the cases do not contain anything special about them.
fl2: the actor spends at least 2 s walking normally before swinging and falling.
A total of 19.33% of cases of fall predicted as violence involved the following common sources of errors.
fl3: the actor tries to balance himself by raising arms sideways before the fall.
fl4: the actor walking unsteadily and falls sideward.
fl5: the actor falls while raising a hand in which he is holding a ball.
A total of 6.72% of cases of loitering predicted as fall involved the following common sources of errors.
lf1: the actor stretches arms overhead while walking straight.
lf2: the actor walks unsteadily for a minimum of 5 s before bending down to pick an object from the floor.
A total of 36.69% of cases of loitering predicted as violence involved the following common sources of errors.
lv1: the actor is walking fast while moving arms swiftly.
lv2: the actor is walking normally, holding an object (stick, handbag) in hand.
lv3: the actor walks for a minimum of 5 s, crouches down, and continues walking normally.
A total of 20.45% of cases of violence predicted as fall involved the following common sources of errors. Overall, with respect to actions or movement patterns, these cases are similar to those of fall predicted as violence. Similarly, a total of 12.89% cases of violence predicted as loitering involved the following common sources of errors. A majority of movement patterns in this class resemble those listed above for loitering predicted as violence. In addition to the above, other random cases of misclassifications (making a total of 1.68%) were also noted but they were insignificant and appeared only in a few stacks for each abnormal behavior.

5.3.4. Discussion of Detection Errors

A thorough insight into different patterns found in each of the abnormal acts led to accurately addressing the training needs of the system with regards to each pattern. This has resulted in a continuous improvement in performance during the development of the recognition model. By providing appropriate data related to each pattern, the system has been able to mitigate the problems related to variance pertaining to a single behavior. However, considering the overlapping between acts of three classes, most of the cases of misclassification are found to be intrinsic rather than accidental. So far as the other sources of errors described above are concerned, they could be classified into two main groups after a detailed analysis.

Rare appearance of certain patterns in the datasets: the network essentially finds it difficult to learn the patterns belonging to each class that do not appear in many examples. Some examples of such rare occurrences include falling after swinging for some time, falling while grabbing an object, and detecting a behavior in a distant scene. Insufficient examples
of specific patterns can be observed as the main cause of a vast majority of the existing false
negatives, as in some cases no other explanation could be found except that the system is
unable to learn the features required to detect that pattern.

Limitations related to quality of videos: the quality of video and resultant frames is an
inherent feature of the underlying datasets. It was observed that in some cases (e.g., hockey
fights shot from a long distance), the optical flow algorithm was unable to capture the
movement of players correctly.

5.4. Performance Evaluation

The study measured the performance of the model in terms of the number of correct
predictions made against each class. The results are shown in Table 5. One can see from
the results that the model is more likely to misclassify other activities (none) as one of
the abnormal behaviors (loitering). However, as previously explained in Section 4.4.2,
the model was deemed acceptable to have some false positives rather than misclassifying one
of the abnormal conditions as “none”. Another case of a higher misclassification rate can
be seen in loitering being predicted as “none”. As described in the previous section, this
is mainly due to the essential difficulty of distinguishing between the two. Similarly,
misclassifications are slightly high in the case of falling versus violence and vice versa. One
possible reason for this could be that the most confounding activities to falling such as
bending and lying are frequently found in the videos of violence.

Table 5. Confusion Matrix (%) for the proposed approach.

| Actual  | Predicted |
|---------|-----------|
|         | Falling   | Loitering | Violence | None     |
| Falling | 98.03     | 0.05      | 1.36     | 0.56     |
| Loitering | 0.01     | 93.34     | 0.11     | 6.54     |
| Violence | 1.04     | 2.00      | 96.91    | 0.05     |
| None    | 0.92      | 4.61      | 1.62     | 92.85    |

Table 6 shows the classification results obtained using the basic setup described in
Section 5.2. Table 7 shows the results obtained after considering the distinguishing move-
ment patterns and increasing the number of examples for each case. A comparison
of results in these two tables shows the significant improvement achieved using the enhanced
setup. Further, the results in Table 7 show that the model generalizes well and yields ade-
quate classification performance for all abnormal behaviors. The $F_1$ score (a combination
of recall and precision) for falling, loitering, and violence i.e., 98%, 93%, and 97% respectively,
is an indicator that the model can accurately distinguish between different events involving
abnormal behavior.

Table 6. Overall classification test results for 4 classes with the basic setup.

|         | Falling | Loitering | Violence | None |
|---------|---------|-----------|----------|------|
| Recall  | 0.92    | 0.73      | 0.88     | 0.87 |
| FP Rate | 0.04    | 0.02      | 0.02     | 0.11 |
| Precision | 0.88   | 0.91      | 0.92     | 0.72 |
| Accuracy | 0.95    | 0.91      | 0.95     | 0.88 |
| $F_1$   | 0.90    | 0.81      | 0.90     | 0.79 |
Table 7. Final classification test results for 4 classes.

|               | Falling | Loitering | Violence | None |
|---------------|---------|-----------|----------|------|
| Recall        | 0.98    | 0.93      | 0.97     | 0.93 |
| FP Rate       | 0.01    | 0.02      | 0.01     | 0.02 |
| Precision     | 0.98    | 0.93      | 0.97     | 0.93 |
| Accuracy      | 0.99    | 0.97      | 0.98     | 0.96 |
| $F_1$         | 0.98    | 0.93      | 0.97     | 0.93 |

5.5. Comparison with the State-of-the-Art

To further evaluate the proposed approach, it was compared with state-of-the-art works reported in the literature for the detection of falling, loitering, and violence. To make a fair comparison, the study selected the existing approaches that work on RGB data (instead of other inputs such as depth maps). So far as the comparison criteria are concerned, this work has not been compared with a common baseline as there is no existing work in the literature addressing all three abnormal behaviors within the same study. This means that the proposed approach must be compared with other works that were trained on one or more datasets corresponding to only one behavior and hence evaluated on those specific datasets. On the other hand, since one of the main drivers of the idea of this work was generality, that is, to enable detection of three behaviors in different scenarios, the experiments in the current work were conducted with the system trained on a combined dataset. However, despite this difference in training and evaluation conditions, the dataset employed in this work is essentially a combination of the same datasets that were used in the other works. Therefore, it is expected that the comparison should still give a reasonable picture of the performance level of our model. So, instead of only presenting the results of metrics, this section will also provide brief details of the datasets and training conditions of each work. Furthermore, it should be noted that since researchers have used diverse performance measures, the results will be presented based on the metrics originally employed in each paper. A majority of the approaches have reported results of accuracy. However, as previously stated, since higher accuracy does not always imply a higher predictive power, precision and recall values of our method are also reported for all classes.

The results of comparison with approaches for falling, loitering, and violence detection are tabulated in Tables 8–10, respectively. Our comparison includes both handcrafted based as well as deep learning approaches. Note that all methods included in this section used a binary classification-based approach for a single abnormal behavior detection, that is, to detect the presence of one of the fall, loitering, or violent behaviors in videos. Zerrouki and Houacine [65] use curvelet transforms and area ratios features to characterize the human body. They use an SVM classifier to identify the posture and then to distinguish fall events from other activities, they apply a Hidden Markov Model (HMM). They reported an accuracy of 97% on URF-DS. Kun et al. [48] develop an augmented set of features named HLC (a combination of Histograms of Oriented Gradients (HOG), Local Binary Pattern (LBP), and feature extracted by the Deep Learning Framework Caffe) to represent human motion. Then by using two SVMs to classify the fall events, they have achieved a recall value of 93% on Multicam-DS. Yao et al. [12] have combined the use of geometric features and CNNs. Specifically, they use Gaussian mixture model (GMM) to represent the features of each pixel pertaining to the foreground and proposed a novel method of head segmentation (locating, tracking, ellipse fitting of head). Later, a 2D-CNN is used to learn the correlation of the head and torso ellipses during the fall. For training and testing, they used a self-collected dataset of 102 videos containing 30 falls and 28 other activities and achieved an accuracy of 90% on this dataset. Our method is closely related to the work of Adrian et al. [8] where instead of using any feature engineering, a CNN model pretrained in Imagenet and UCF-101 has been used to detect falls based on RGB data and optical flow images. However, instead of using two parallel streams for RGB and optical flow, they train the model first on RGB and then on optical flow. They have reported results obtained
for each dataset separately as well as for a combination of three datasets. In this way, their combined dataset setup is similar to our subset related to falling behavior on which they have reported a recall of 94% compared to the same of 98% in our case. As evident from the results in Table 8, our approach performs better than the state-of-the-art in terms of accuracy. Adrian et al.’s method performs better in terms of recall when tested on URF-DS and Multicam-DS individually. The higher value for precision, which refers to the fraction of predictions of falls that actually fell, suggests that the prediction performance is better than the state-of-the-art.

Table 8. Comparison with the state-of-the-art for falling.

| Paper                  | Pretraining         | Dataset        | Data Used | Method/Classification Approach                                      | Precision | Recall | Accuracy |
|------------------------|---------------------|----------------|-----------|---------------------------------------------------------------------|-----------|--------|----------|
| Zerrouki and Houacine  | -                   | URF-DS         | RGB       | Curvelet transforms and area ratios features+SVM-HMM                | -         | -      | 0.97     |
| Kun et al. [48]        | -                   | Multicam-DS    | RGB       | Histograms of Oriented Gradients (HOG), LBP, Caffe features+SVM     | -         | 0.93   | -        |
| Yao et. al. [12]       | -                   | Self-collected | RGB       | GMM+2D-CNN                                                          | -         | -      | 0.90     |
| Adrian et al. [8]      | Imagenet + UCF-101  | URF-DS         | RGB, OF   | 2D-CNN                                                             | 1.00      | 0.99   | 0.95     |
|                        |                     | Multicam-DS    | RGB, OF   | 2D-CNN                                                             |           |        |          |
| Proposed Method        | Imagenet + Kinetics | UFLV-DS        | RGB, OF   | 2-Stream 3D-CNN                                                     | 0.98      | 0.98   | 0.99     |

Table 9. Comparison with the state-of-the-art for loitering.

| Paper                  | Pretraining         | Dataset        | Data Used | Method/Classification Approach                                      | Precision | Recall | Accuracy |
|------------------------|---------------------|----------------|-----------|---------------------------------------------------------------------|-----------|--------|----------|
| Gomez et al. [15]      | -                   | Caviar-DS      | RGB       | Sequential micro-patterns + Generalized Sequential Patterns (GSP) algorithm | 0.87      | 0.53   | -        |
| Huang et al. [14]      | -                   | PETS2007+ Self-collected | RGB       | GMMs fusion, MeanShift+ activity area-based trajectory detection algorithm | 0.96      | 1.00   | 0.97     |
| Proposed Method        | Imagenet + Kinetics | UFLV-DS        | RGB, OF   | 2-Stream 3D-CNN                                                     | 0.93      | 0.93   | 0.97     |
Table 10. Comparison with the state-of-the-art for violence.

| Paper          | Pretraining | Dataset     | Data Used | Method/Classification Approach | Precision | Recall | Accuracy |
|----------------|-------------|-------------|-----------|---------------------------------|-----------|--------|----------|
| Ding et al. [17] | -           | Hockey-DS  | RGB       | 3D-CNN                          | -         | -      | 0.89     |
| Nievas et al. [18] | -           | Hockey-DS  | RGB       | STIP+MoSIFT+HIK                 | -         | -      | 0.91     |
| Movies-DS      |             |             |           |                                 |           |        |          |
| Asad et al. [20] | -           | Hockey-DS  | RGB       | Feature fusion+2D-CNN+LSTM       | -         | -      | 0.99     |
|                 |             | Movies-DS  |           |                                 |           |        |          |
|                 |             | Crowd Violence |        |                                 |           |        | 0.97     |
|                 |             | BEHAVE     |           |                                 |           |        | 0.95     |
| Proposed Method | Imagenet +  | UFLV-DS     | RGB, OF   | 2-Stream 3D-CNN                 | 0.97      | 0.97   | 0.98     |

Gomez et al. [15] have used the Generalized Sequential Patterns (GSP) algorithm to detect sequential micro-patterns in the input videos. This method maintains a database of frequently found loitering patterns and matches the newly found patterns with it for classification. They have reported the precision and recall values of 87% and 53%, respectively, on Caviar-DS. Huang et al. [14] have presented a method based on pedestrian activity area classification. They divided the loitering behaviors into three categories, namely rectangle, ellipse, and sector loitering. The proposed algorithms for each category calculate enclosing areas through curve fitting based on trajectory coordinates. Compared with our method, they have reported higher precision and recall results on a large dataset i.e., PETS2007 along with a self-collected set of videos. However, seeing that their model is trained for one action only and was trained on the PETS2007 dataset, which is significantly larger than our dataset of activities related to loitering, our results are satisfactory. Moreover, our detection approach avoids the work involved in trajectory calculations, analysis, and handcrafting.

Ding et al. [17] have used a 3D-CNN to achieve 89% accuracy on the Hockey-DS. Nievas et al. [18] have developed a histogram intersection kernel (HIK) framework using the Space-Time Interest Points (STIP) and Motion SIFT (MoSIFT) action descriptors. The method has been trained and tested on Hockey-DS and Movies-DS with an accuracy of 91% and 89%, respectively. Asad et al. [20] proposed a multi-level fusion approach that integrates local motion patterns using features from sequences of input frames. They use a 2D-CNN in combination with a Wide-Dense Residual Block (WDRB) to learn feature maps. The model was trained on four datasets i.e., Hockey-DS, Movies-DS, Crowd Violence, and BEHAVE separately to achieve high accuracy results. However, they have not reported the precision and recall values for their model. Hence, considering the overall results for the joint detection of three behaviors on the combined dataset, our approach yields relatively better results. The overall performance of our model has been positively affected due to pretraining on a fairly large activity dataset, i.e., Kinetics, as well as through the use of an enhanced dataset based on insight into specifics of each abnormal behavior.

6. Conclusions and Future Work

In this paper, a deep learning-based approach for the detection of falling, loitering, and violence in uncrowded scenes was presented. The paper proposed a two-stream CNN architecture that adopts an Inception 3D network for each of the spatial and temporal streams. The approach can extract spatiotemporal information and ensure the full use of motion information to achieve higher performance. Moreover, a successful application of transfer learning was presented that adopted the existing action recognition features and carried out specialized abnormal behavior detection. The experiments on the specialized
dataset have shown that the approach can detect falling, loitering, and violence with an accuracy of up to 99%, 97%, and 98%, respectively. Similarly, behavior classification performs well in terms of precision and recall as well. Precisely, it achieves $F_1$ scores of 98%, 93%, and 97% for falling, loitering, and violence detection.

In future research, the author intends to investigate the inclusion of more abnormal behaviors (e.g., person in the wrong place) as well as the possibility of training deeper convolutional layers on other datasets to improve the model performance. Similarly, the combined detection of multiple behaviors in a single scene (e.g., loitering may take place while someone is falling) is also an interesting direction for future research. Furthermore, although optical flow is a powerful technique to represent motion, it also results in expensive computations. Therefore, to avoid any preprocessing steps, the design of more complex architectures may be investigated that can learn motion representations from raw data.

Funding: The author acknowledges the Deanship of Scientific Research at King Faisal University for the financial support under Nahser Track (Grant No. 186330).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The datasets used in this study are publicly available from sources cited in the paper.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Ben Mabrouk, A.; Zagrouba, E. Abnormal behavior recognition for intelligent video surveillance systems: A review. Expert Syst. Appl. 2018, 91, 480–491. [CrossRef]

2. Ullah, H.; Islam, I.U.; Ullah, M.; Afq, M.; Khan, S.D.; Iqbal, J. Multi-feature-based crowd video modeling for visual event detection. Multimed. Syst. 2020, 1–9. [CrossRef]

3. Rossetto, L.; Gasser, R.; Lokoc, J.; Baier, W.; Schoeffmann, K.; Muenzer, B.; Soucek, T.; Nguyen, P.A.; Bolettiere, P.; Leibetsder, A.; et al. Interactive Video Retrieval in the Age of Deep Learning—Detailed Evaluation of VBS 2019. IEEE Trans. Multimed. 2021, 23, 243–256. [CrossRef]

4. Tsakanikas, V.; Dagjulkas, T. Video surveillance systems-current status and future trends. Comput. Electr. Eng. 2018, 70, 736–753. [CrossRef]

5. Wang, T.; Miao, Z.; Chen, Y.; Zhou, Y.; Shan, G.; Snoussi, H. AED-Net: An Abnormal Event Detection Network. Engineering 2019, 5, 930–939. [CrossRef]

6. Ullah, W.; Ullah, A.; Haq, I.U.; Muhammad, K.; Sahjed, M.; Baik, S.W. CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks. Multimed. Tools Appl. 2020, 1–17. [CrossRef]

7. Afiq, A.; Zakariya, M.; Saad, M.; Nurfarzana, A.; Khir, M.; Fadzil, A.; Jale, A.; Gunawan, W.; Izuddin, Z.; Faizari, M. A review on classifying abnormal behavior in crowd scene. I. Vis. Commun. Image Represent. 2019, 58, 285–303. [CrossRef]

8. Núñez-Marcos, A.; Azkune, G.; Arganda-Carreras, I. Vision-Based Fall Detection with Convolutional Neural Networks. Wirel. Commun. Mob. Comput. 2017, 2017, 1–16. [CrossRef]

9. Rezaee, K.; Haddadnia, J.; Delbari, A. Modeling abnormal walking of the elderly to predict risk of the falls using Kalman filter and motion estimation approach. Comput. Electr. Eng. 2015, 46, 471–486. [CrossRef]

10. Nguyen, V.D.; Le, M.T.; Do, A.D.; Duong, H.H.; Thai, T.D.; Tran, D.H. An efficient camera-based surveillance for fall detection of elderly people. In Proceedings of the 9th IEEE Conference on Industrial Electronics and Applications, Hangzhou, China, 9–11 June 2014; pp. 994–997.

11. Afsan, M.; Sengur, A.; Xiao, Y.; Wang, H.; Ince, M.C.; Ma, X. Shape feature encoding via Fisher Vector for efficient fall detection in depth-videos. Appl. Soft Comput. 2015, 37, 1023–1028. [CrossRef]

12. Yao, C.; Hu, J.; Min, W.; Deng, Z.; Zou, S.; Min, W. A novel real-time fall detection method based on head segmentation and convolutional neural network. J. Real-Time Image Process. 2020, 17, 1–11. [CrossRef]

13. Khrayef, C.; Benzarti, F.; Amiri, H. Elderly fall detection based on multi-stream deep convolutional networks. Multimed. Tools Appl. 2020, 79, 19537–19560. [CrossRef]

14. Huang, T.; Han, Q.; Min, W.; Li, X.; Yu, Y.; Zhang, Y. Loitering Detection Based on Pedestrian Activity Area Classification. Appl. Sci. 2019, 9, 1866. [CrossRef]
15. Tomás, R.M.; Tapia, S.A.; Caballero, A.F.; Ratté, S.; Eras, A.G.; González, P.L. Identification of Loitering Human Behaviour in Video Surveillance Environments; Springer: Berlin/Heidelberg, Germany, 2015.

16. Lim, M.K.; Tang, S.; Chan, C.S. Surveillance: Intelligent framework for multiple events detection in surveillance videos. Expert Syst. Appl. 2014, 41, 4704–4715. [CrossRef]

17. Ding, C.; Fan, S.; Zhu, M.; Feng, W.; Jia, B. Violence detection in video by using 3D convolutional neural networks. In International Symposium on Visual Computing; Springer: Berlin/Heidelberg, Germany, 2014; pp. 551–558.

18. Nievas, E.B.; Suaraz, O.D.; Garcia, G.B.; Sukthankar, R. Violence detection in video using computer vision techniques. In International Conference on Computer Analysis of Images and Patterns; Springer: Berlin/Heidelberg, Germany, 2011; pp. 332–339.

19. Song, W.; Zhang, D.; Zhao, X.; Yu, J.; Zheng, R.; Wang, A. A Novel Violent Video Detection Scheme Based on Modified 3D Convolutional Neural Networks. IEEE Access 2019, 7, 39172–39179. [CrossRef]

20. Asad, M.; Yang, J.; He, J.; Shamsolmoali, P.; He, X. Multi-frame feature-fusion-based model for violence detection. Vis. Comput. 2020, 1–17. [CrossRef]

21. Kim, J.-H.; Won, C.S. Action Recognition in Videos Using Pre-Trained 2D Convolutional Neural Networks. IEEE Access 2020, 8, 60179–60188. [CrossRef]

22. Lu, Z.; Qin, S.; Li, X.; Li, L.; Zhang, D. One-shot learning hand gesture recognition based on modified 3d convolutional neural networks. Mach. Vis. Appl. 2019, 30, 1157–1180. [CrossRef]

23. Hara, K.; Kataoka, H.; Satoh, Y. Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet? In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6546–6555.

24. Kataoka, H.; Wakamiya, T.; Hara, K.; Satoh, Y. Would mega-scale datasets further enhance spatiotemporal 3d cnns? arXiv 2020, arXiv:2004.04968.

25. Tripathi, G.; Singh, K.; Vishwakarma, D.K. Convolutional neural networks for crowd behaviour analysis: A survey. Vis. Comput. 2018, 35, 753–776. [CrossRef]

26. Feichtenhofer, C.; Pinz, A.; Zisserman, A. Convolutional Two-Stream Network Fusion for Video Action Recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 1933–1941.

27. Simonyan, K.; Zisserman, A. Two-stream convolutional networks for action recognition in videos. arXiv 2014, arXiv:1406.2199.

28. Hara, K.; Kataoka, H.; Satoh, Y. Learning Spatio-Temporal Features with 3D Residual Networks for Action Recognition. In Proceedings of the IEEE International Conference on Computer Vision Workshops (ICCVW), Venice, Italy, 22–29 October 2017; pp. 3154–3160.

29. Varol, G.; Laptev, I.; Schmid, C. Long-Term Temporal Convolutions for Action Recognition. IEEE Trans. Pattern Anal. Mach. Intell. 2018, 40, 1510–1517. [CrossRef]

30. Sha, L.; Zhiwen, Y.; Kan, X.; Jinli, Z.; Honggang, D. An improved two-stream CNN method for abnormal behavior detection. J. Physics: Conf. Ser. 2020, 1617, 012064. [CrossRef]

31. Carreira, J.; Zisserman, A. Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Honolulu, HI, USA, 21–26 July 2017; pp. 4724–4733.

32. Xie, S.; Sun, C.; Huang, J.; Tu, Z.; Murphy, K. Rethinking Spatiotemporal Feature Learning: Speed-Accuracy Trade-offs in Video Classification. In Medical Image Computing and Computer Assisted Intervention–MICCAI 2018; Apress: New York, NY, USA, 2018; pp. 318–335.

33. Ioffe, S.; Szegedy, C. Batch normalization: Accelerating deep network training by reducing internal covariate shift. arXiv 2015, arXiv:1502.03167.

34. Chen, M.Y.; Hauptmann, A. MoSIFT: Recognizing Human Actions in Surveillance Videos; Carnegie Mellon University: Pittsburgh, PA, USA, 2009.

35. Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; Rabinovich, A. Going deeper with convolutions. In Proceedings of the Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2015.

36. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. arXiv 2015, arXiv:1512.03385.

37. Dai, W.; Chen, Y.; Huang, C.; Gao, M.-K.; Zhang, X. Two-Stream Convolutional Neural Network with Video-stream for Action Recognition. In Proceedings of the International Joint Conference on Neural Networks (IJCNN), Budapest, Hungary, 14–19 July 2019; pp. 1–8.

38. Ramya, P.; Rajeswari, R. Human action recognition using distance transform and entropy based features. Multimedia Tools Appl. 2021, 80, 8147–8173. [CrossRef]

39. Chirki, A.; Touati, H.; Snoussi, H.; Kamoun, F. Deep learning and handcrafted features for one-class anomaly detection in UAV video. Multimed. Tools Appl. 2021, 80, 2599–2620. [CrossRef]

40. Castellano, G.; Castiello, C.; Cianciotta, M.; Mencar, C.; Vessio, G. Multi-view Convolutional Network for Crowd Counting in Drone-Captured Images. In HCI International 2020—Late Breaking Papers: Cognition, Learning and Games; Springer Science and Business Media LLC: Berlin/Heidelberg, Germany, 2020; pp. 588–603.

41. Castellano, G.; Castiello, C.; Mencar, C.; Vessio, G. Crowd Detection in Aerial Images Using Spatial Graphs and Fully-Convolutional Neural Networks. IEEE Access 2020, 8, 64534–64544. [CrossRef]
