mathematics

MDPI

*Article*

# EADN: An Efficient Deep Learning Model for Anomaly Detection in Videos

**Sareer Ul Amin [1], Mohib Ullah [2], Muhammad Sajjad [1,2,*], Faouzi Alaya Cheikh [2], Mohammad Hijji [3], Abdulrahman Hijji [4] and Khan Muhammad [5,*]**

[1] Digital Image Processing Laboratory, Department of Computer Science, Islamia College Peshawar, Peshawar 25000, Pakistan; sareerulamin320@gmail.com
[2] Software, Data and Digital Ecosystems, Department of Computer Science, Norwegian University for Science and Technology (NTNU), 2815 Gjøvik, Norway; mohib.ullah@ntnu.no (M.U.); faouzi.cheikh@ntnu.no (F.A.C.)
[3] Industrial Innovation and Robotic Center (IIRC), University of Tabuk, Tabuk 47711, Saudi Arabia; m.a.hijji@gmail.com
[4] Department of Civil and Environmental Engineering, King Fahd University of Petroleum and Minerals (KFUPM), Dharan 31261, Saudi Arabia; a.hijji@hotmail.com
[5] Visual Analytics for Knowledge Laboratory (VIS2KNOW Lab), Department of Applied Artificial Intelligence, School of Convergence, College of Computing and Informatics, Sungkyunkwan University, Seoul 03063, Korea
* Correspondence: muhammad.sajjad@ntnu.no (M.S.); khanmuhammad@g.skku.edu (K.M.)

**Abstract:** Surveillance systems regularly create massive video data in the modern technological era, making their analysis challenging for security specialists. Finding anomalous activities manually in these enormous video recordings is a tedious task, as they infrequently occur in the real world. We proposed a minimal complex deep learning-based model named EADN for anomaly detection that can operate in a surveillance system. At the model's input, the video is segmented into salient shots using a shot boundary detection algorithm. Next, the selected sequence of frames is given to a Convolutional Neural Network (CNN) that consists of time-distributed 2D layers for extracting salient spatiotemporal features. The extracted features are enriched with valuable information that is very helpful in capturing abnormal events. Lastly, Long Short-Term Memory (LSTM) cells are employed to learn spatiotemporal features from a sequence of frames per sample of each abnormal event for anomaly detection. Comprehensive experiments are performed on benchmark datasets. Additionally, the quantitative results are compared with state-of-the-art methods, and a substantial improvement is achieved, showing our model's effectiveness.

**Keywords:** anomaly detection; shots segmentation; computer vision; deep learning; histogram difference; keyframe extraction; intelligent surveillance networks; crime detection

**MSC:** 68T45

**Citation:** Ul Amin, S.; Ullah, M.; Sajjad, M.; Cheikh, F.A.; Hijji, M.; Hijji, A.; Muhammad, K. EADN: An Efficient Deep Learning Model for Anomaly Detection in Videos. *Mathematics* **2022**, *10*, 1555. https://doi.org/10.3390/math10091555

Academic Editors: Yue-Jiao Gong and Ting Huang

Received: 25 March 2022
Accepted: 18 April 2022
Published: 5 May 2022

## 1. Introduction

In the twenty-first century, the rise in the crime rate is one of the prime reasons for lost lives [1]. A smart video surveillance system is one possible solution for rapidly detecting unexpected criminal events. Recently, enormous quantities of surveillance cameras have been put globally in various areas for public safety. Due to the limitations of manual surveillance, law enforcement organizations perform poorly in detecting or avoiding anomalous activities. A smart computer vision system is required to detect unusual behavior, one that can effectively recognize normal and abnormal events without human intervention. Such an automated system is beneficial for monitoring and decreases the human work required to maintain 24-h manual observation.

In the current literature, anomaly detection approaches based on sparse coding have shown promising results to date [2–5]. These techniques are assumed to be standard for anomaly identification. These approaches are trained so that the initial frames of a video are utilized to construct a dictionary of usual events. Despite computational efficiency, this is a poor technique for accurately detecting abnormal events. Weakly supervised multi-instance learning (MIL)-based techniques are also explored for anomaly detection in [3,5,6]. In such techniques, the training phase divides the videos into a predefined number of segments. These segments construct a bag of instances for positive and negative samples.

Due to the dynamic nature of surveillance environments, sparse coding techniques have specific drawbacks. For example, converting the dictionary from usual to anomalous events leads to a high false-positive and false-negative. Additionally, recognizing anomalous events in low resolution noisy videos is exceedingly difficult. While humans can recognize regular or uncommon events based on their intuition, machines must rely on visual features.

Predominantly, the existing methods suffer from a high proportion of false alarms, leading to lower performance. Furthermore, these approaches perform well on small datasets, but their performance is limited when applied to real-world circumstances. In this paper, to address these concerns, we proposed a simple and efficient lightweight model to detect anomalies in surveillance videos. EADN uses a windowing approach and processes five frames per sample, chronologically ordered to detect movements and actions in the surveillance videos. Our model learns visual features from a sequence of five frames per sample by integrating them into spatiotemporal information from surveillance videos. The key contributions of our work are as follows:

- Pre-Processing: Surveillance camera generates huge amount of data on a regular basis in the current technological era, needing a considerable amount of computational complexity and time for exploration. Existing techniques in anomaly detection in surveillance video literature lack the focus on pre-processing steps. In this paper, we present a novel and efficient pre-processing strategy of video shots segmentation, where shots boundaries are segmented based on underlying activity. Further, the segmented shots of the video can be processed for advanced analysis such as anomaly activities without any transmission delay. Thus, our pre-processing strategy plays a prominent role in the overall anomaly detection in the surveillance video framework.
- A simple, light-weight, and novel CNN model consisting of time-distributed layers for learning spatiotemporal features from a series of frames per sample is proposed.
- A comprehensive model evaluation on standard performance metrics using a challenging benchmark dataset and achieving promising results compared to the state-of-the-art with a model size of only 53.9 MBs.

The rest of the paper is organized in the following order. In Section 2, the background and related work is explained. The EADN framework is elaborated in Section 3. Details about the dataset, the quantitative evaluation, and discussion are given in Section 4. The final remarks and future directions are given in Section 5, which concludes the paper.

## 2. Related Work

The literature of anomaly detection methods is discussed in two main categories: Traditional handcrafted feature-based methods and deep feature-based methods for anomalous event recognition. Previously, anomaly detection was highly dependent on low-level hand-crafted feature-based methods. These methods are primarily based on three stages: (1) Feature extraction, in which low-level patterns from the training set are extracted; (2) feature learning is distinguished by the distribution of encoding regularity or normal events; and (3) outlier detection, separated clusters or outliers are identified as anomalous events. For example, Zhang et al. [7] employed the Markov random field to represent common occurrences by using spatiotemporal features. Similarly, Mehran et al. [8] developed a social interaction model in which cooperation forces were computed, and normal and abnormal activities were detected using optical flow. Furthermore, Nor et al. [9] proposed an explainable anomaly detection framework that assists in prognostic and health management PHM. Their framework is based on a Bayesian deep learning model with predefined prior and likelihood distributions. It provides additive explanations to come up with the local and global explanations for the PHM tasks. Similarly, Ullah et al. [10] come up with an attention-based LSTM for the action recognition in sport videos. They used convolution block attention to refine the spatial features. Similarly, a fully connected neural network with a softmax classifies the refine feature maps into different sports actions. Compared to visual data, Selicato et al. [11] worked on detecting anomalies in the gene data. For example, they come up with an ensemble-based approach for detecting the normal and abnormal gene expression matrices using hierarchical clustering and principal component analysis (PCA). Riaz et al. [12] proposed an ensemble of deep model for detecting anomalies in complex scenes. In the first step, a human pose estimation model is incorporated that detects the human joints. In the second step, the detected joints are treated as features and are given to a densely connected fully CNN for the anomaly detection. More recently, Zhao et al. [13] proposed an unsupervised technique that used a time varying sparse coding style to detect anomalies in videos using online query signals and sparse reconstruction ability acquired from a learned vocabulary of all events. However, developing the ability to identify anomalies in a timely way has remained a challenge, attracting the interest of several researchers. For example, Lu et al. [4] utilized Sparse Combination Learning (SCL) and analyzed their technique with local and cloud servers.

Compared to traditional methods, deep feature-based models have gained significant success in a variety of nonlinear high-dimensional data domains in the modern age, including activity identification and video summarization, among others. Liu et al. [14] developed a system in which video frames were encoded using a CNN, and anomalous events were detected using ConvLSTM. Their encoder encodes motion variation to detect abnormalities in a surveillance environment. Hasan et al. [15] proposed a convolution autoencoder and a RNN. Luo et al. [14] came up with a convolutional LSTM model with an autoencoder for video anomaly detection. Moreover, they expanded this work by detecting anomalies using a stacked RNN with an autoencoder. Liu et al. [16] proposed a technique for detecting video abnormalities by integrating a temporal and spatial detector. The discriminant saliency detector and a collection of dynamic texture features were treated in this model as normal training data events. Cheng et al. [17] developed a clustering-based deep autoencoder to extract information from normal events efficiently. Two modules are used to learn spatial-temporal feature regularity, and the spatial autoencoder in the first module handles the last individual video frame. In contrast, the temporal autoencoder in the second module operates and produces the RGB difference between the frames. Additionally, generative models are used for detecting anomalies in videos. For instance, Sabokroul et al. [18] came up with generative adversarial networks (GANs) for detecting abnormalities in a videos. This model teaches the normal distribution using GANs with discriminator and generator techniques. More recently, weakly supervised approaches for video labeling have been proposed where anomalous events are detected using C3D and MIL [19,20]. For example, Sultani et al. [5] developed a framework for detecting anomalous events based on weak video labels, and the MIL method. This model was trained on both normal and unusual videos by creating two distinct bags for common and uncommon events and then using the MIL technique to detect anomalous activity scores in the videos. Landi et al. [21] proposed a tube extraction technique that uses coordinates to build a regression model for abnormalities. Briefly, the average pooling layer combines the spatial features from the inception block with the temporal features of optimal flow before sending it to the regression model. Zhong et al. [22] developed a technique for detecting weakly supervised anomalies and a supervised system for action categorization with noisy labels. The novelty was keeping only the anomaly video labels noisy due to the unpredictable nature of the anomalous events. Additionally, a graph CNN was trained to clean up these noisy labels, and the activities were classified using an action classifier. Compared to the existing approaches, this paper proposed an efficient time-distributed 2D CNN with LSTM for anomaly detection in videos. The prominent features of the proposed method are discussed in Section 3.

## 3. Proposed Framework

This section discusses the EADN framework and its essential constituent structure in detail. The pictorial representation of our EADN framework is presented in Figure 1. In a nutshell, the anomaly detection and classification task consist of three parts: (i) Shot segmentation, (ii) feature extraction, and (iii) sequence learning and anomaly classification. In the first step, salient frames are segmented using a shot boundary detection algorithm. The segmented frames are given to the Lightweight CNN ($LW_{CNN}$) model to extract spatiotemporal features from the intermediate layer. After that, LSTM cells are used to learn spatiotemporal features from a sequence of frames per sample of each anomalous event. The proposed $LW_{CNN}$ took frames that are chronologically ordered to detect movements and action. Finally, the proposed trained $LW_{CNN}$ LSTM network is used to recognize anomalous events in the segmented shot of the video. A comprehensive list of model parameters is given in Table 1. The model is trained end-to-end and the categorical cross entropy loss function is used to optimize its parameters. Mathematically, the loss function is defined as:

$$Loss = -\sum_{i=1}^{N} y_i \cdot \log \hat{y}_i \tag{1}$$

where $N$ is the total number of anomalies, $y_i$ is the true class label, and $\hat{y}_i$ is the estimated probability given by the model. Using this configuration, the model takes the key frames of a given video and returns the score for the corresponding anomaly accoridngly. It is important to mention that the key frames are representative of the input video. Therefore, the anomaly detected in the key frames can be generalized as an anomaly occurring in the input video.

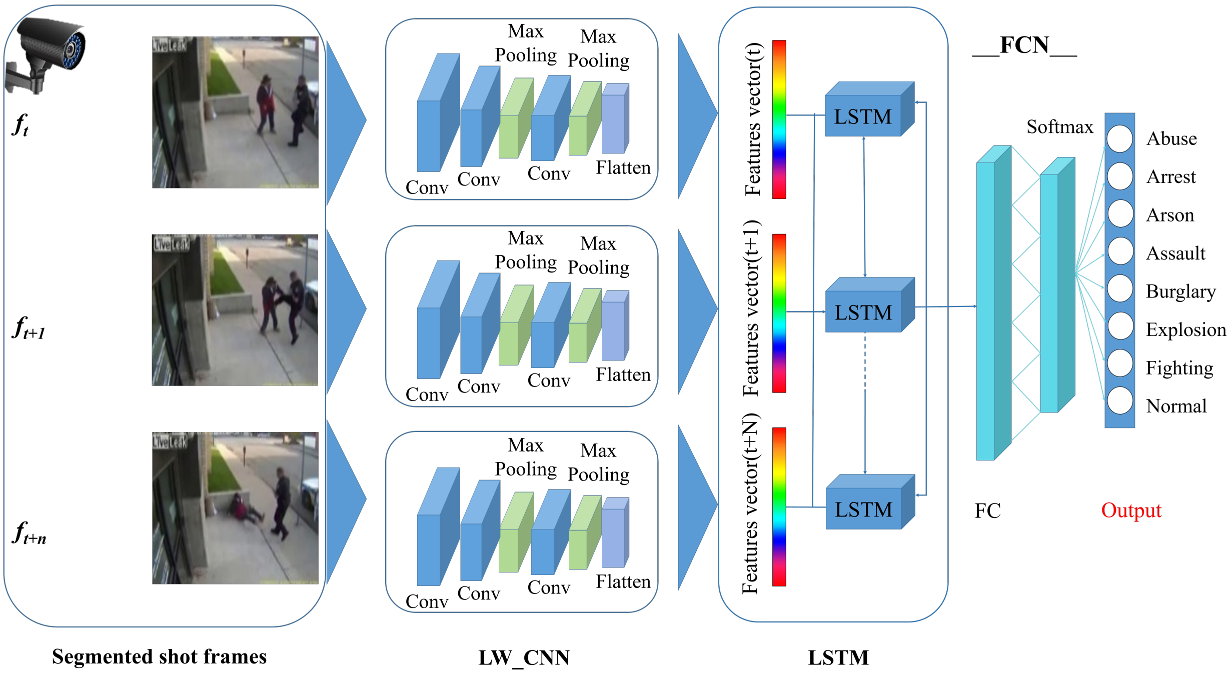


**Figure 1.** EADN framework for anomaly detection in surveillance videos.

**Table 1.** Summary of the input and output parameters in the proposed framework.

| Symbol | Description |
|---|---|
| m | Rows. |
| n | Columns. |
| $\mathcal{BD}(f_t, f_{t+1})$ | Block Difference between the corresponding blocks of the consecutive frames. |
| $\mathcal{H}(f_t, j, c, b)$ | Histogram value for the $j$th level, in the channel $c$ at the $b$th block in the frame $t$. |
| $\mathcal{H}(f_{t+1}, j, c, b)$ | Histogram value for the $j$th level, in the channel $c$ at the $b$th block in the frame $t+1$. |
| $\mathcal{N}_B$ | Total number of blocks. |
| $\mathcal{N}_H$ | Total number of possible gray levels. |
| $\mathcal{D}(f_t, f_{t+1})$ | Histogram difference between two consecutive frames. |
| $\mathcal{W}_k$ | Block's weight at $(k)$. |
| $\mathcal{MD}$ | Mean of the histogram difference. |
| $\mathcal{STD}$ | Standard variance of the histogram difference. |
| $\tau$ | Threshold. |
| $x$ | Reference frame. |
| $\mathcal{T}_v$ | Total number of video sequences. |
| $max(i)$ | Searching for the maximum difference within a shot. |
| $LW_{CNN}$ | Light-Weight Convolutional Neural Network. |
| $\mathcal{F}_t$ | Spatial feature representation series. |
| $h_{t+n}$ | Temporal feature representation series. |
| $h_t$ | Hidden state of the LSTM layer at current time step $t$. |
| $h_{t-1}$ | Hidden state of the LSTM layer at previous time step $t-1$. |
| $i_t$ | Input gate at time $t$. |
| $f_t$ | Forget gate at time $t$. |
| $\mathcal{O}_t$ | Output gate at time $t$. |
| $g$ | Recurrent unit. |
| $\mathcal{C}_t$ | Memory cell. |

### *3.1. Shot Segmentation Using Boundary Detection*

Our shot segmentation algorithm is inspired by [23], where histogram differences are used to detect shot boundary and consequently extracted keyframes. The key steps of the algorithm are the following:

Step 1: First, divide a frame into blocks with $m$ rows and $n$ columns. Second, compute a $x^2$ histogram that matches the difference in corresponding blocks between consecutive frames in the video sequence. Here, the histograms of $b$th blocks in the $f_t$ and $f_{t+1}$ frames are $\mathcal{H}(f_t, j, c, b)$ and $\mathcal{H}(f_{t+1}, j, c, b)$, respectively. Finally, use the following equation to calculate the difference between blocks:

$$\mathcal{BD}(f_t, f_{t+1}) = \sum_{c=1}^{3} \sum_{b=1}^{N_{\mathcal{B}}} \sum_{j=1}^{N_{\mathcal{H}}} \frac{\mathcal{H}(f_t, j, c, b) - \mathcal{H}(f_{t+1}, j, c, b)}{x} \tag{2}$$

$\mathcal{BD}$ represents block difference, and $N_{\mathcal{B}}$ is the total number of blocks, while $N_{\mathcal{H}}$ is the total number of possible gray levels and $\mathcal{H}(f_t, j, c, b)$ is the histogram value for the $j$th level in the channel $c$ at the $b$th block in the frame $t$.

Step 2: Calculate the difference through $x^2$ histograms between two consecutive frames:

$$\mathcal{D}(f_t, f_{t+1}) = \sum_{k=1}^{\mathcal{N}} W_k \mathcal{BD}_k(f_t, f_{t+1}) \tag{3}$$

where $W_k$ denotes the block's weight at $k$ and $\mathcal{N}$ represents the total number of blocks, while $\mathcal{BD}_k(f_t, f_{t+1})$ is the block difference at $k$ between $f_t$ and $f_{t+1}$ frames.

Step 3: Calculate the threshold: For the entire video sequence, compute the difference of the $x^2$ histogram through mean and standard deviations as follows:

$$\mathcal{MD} = \frac{\sum_{f_{t=1}}^{\mathcal{T}_{v-1}} \mathcal{D}(f_t, f_{t+1})}{\mathcal{T}_{v-1}} \tag{4}$$

$$\mathcal{STD} = \sqrt{\frac{\sum_{f_{t=1}}^{\mathcal{T}_{v-1}} (\mathcal{D}(f_t, f_{t+1}))^2}{\mathcal{T}_{v-1}}} \tag{5}$$

where $\mathcal{T}_v$ is the total number of video sequence.

Step 4: Detection of shot boundaries: Let $\mathcal{T} = \mathcal{MD} + \alpha \times \mathcal{STD}$ be the threshold. $\alpha$ is the constant. It weights the standard deviation for the overall threshold $\mathcal{T}$. If $\mathcal{D}(f_t, f_{t+1}) \geq \mathcal{T}$, the frame $f_t$ represents the end of the previous shot, and the frame $f_{t+1}$ represents the end of the following shot. Generally, the shortest shot should last between 1 and 2.5 s. For the sake of fluency, the frame rate must be at least 25 frames per second (in most situations, it is 30 frames per second), or a flash may appear. As a result, a shot must have at least 30 to 45 frames. Thus, a shot merging principle created that state: If a detected shot has fewer than 38 frames, it will be merged into the preceding shot or considered independent. The pseudo-code is given in Algorithm 1.

**Algorithm 1** Pseudocode of shot boundary detection.

```
Require: Total number of videos = $\mathcal{T}_v \in \mathbb{R}^3$
Require: Distribute Each frame $f$ in sixteen block i.e., $\forall f \in \mathbb{B}^{16}$
  for $\mathcal{T} \leftarrow 1\ \mathcal{T}_v$ do
    for $f \leftarrow 1\ \mathcal{T}$ do
      $\mathcal{BD}(f_t, f_{t+1}) \leftarrow \sum_{c=1}^{3} \sum_{b=1}^{N_{\mathcal{B}}} \sum_{j=1}^{N_{\mathcal{H}}} \frac{\mathcal{H}(f_t,j,c,b) - \mathcal{H}(f_{t+1},j,c,b)}{x}$
      $\mathcal{D}(f_t, f_{t+1}) \leftarrow \sum_{k=1}^{\mathcal{N}} W_k \mathcal{BD}_k(f_t, f_{t+1})$
      $\mathcal{MD} \leftarrow \frac{\sum_{f_{t=1}}^{\mathcal{T}_{v-1}} \mathcal{D}(f_t, f_{t+1})}{\mathcal{T}_{v-1}}$
      $\mathcal{STD} \leftarrow \sqrt{\frac{\sum_{f_{t=1}}^{\mathcal{T}_{v-1}} (\mathcal{D}(f_t, f_{t+1}))^2}{\mathcal{T}_{v-1}}}$
      $\tau = \mathcal{MD} + \alpha \times \mathcal{STD}$
      if $\mathcal{D}(f_t, f_{t+1}) \geq \tau$ then
        Previous shot last frame $f_t$
        Next shot last frame $f_{t+1}$.
      else if then
        Print "Shot not detected"
      end if
    end for
    return Shot boundary detection for the given video.
    Repeat loop until last video frame
  end for
```

### 3.2. Extraction of Keyframes

Keyframes are essential in the abstraction of video. The term "keyframes" refers to a collection of prominent frames taken from video sequences. The following describes the algorithm for keyframe extraction:

Step 1: Compute the difference between the general frames (all frames except the reference frame) and the reference frame (first frame of each shot):

$$\mathcal{D}(x, y) = \sum_{k=1}^{\mathcal{N}} W_k \mathcal{BD}_{\|}(x, y_{t+1}) \tag{6}$$

where $W_k$ denotes the block's weight at $k$ and $\mathcal{N}$ represents the total number of blocks, while $\mathcal{BD}_{\|}(x, y_{t+1})$ is the block difference at $k$ between the $x$ reference frame and $y$ general frames.

Step 2: Within a shot, look for the maximum difference:

$$\max(i) = \mathcal{D}(x,y)_{max} \quad y = 2,3,4,\ldots,\mathcal{FC} \tag{7}$$

where $\max(i)$ represents the maximum $x^2$ histogram within shot $i$, and $\mathcal{D}(x,y)$ is the difference between the $x$ reference frame and $y$ general frames, while $\mathcal{FC}$ is the total number of the current shots.

Step 3: Use the relationship between $\max(i)$ and $\mathcal{MD}$ to determine "Shot Type":

$$TypeShot = \begin{matrix} 1 & if \max(i) \geq \mathcal{MD} \\ 0 & \text{otherwise} \end{matrix} \tag{8}$$

A shot will be declared as a dynamic shot if its $\max(i)$ is bigger than $\mathcal{MD}$; otherwise it is a static shot.

Step 4: Determine the keyframe's location: If $TypeShot = 0$ and the number of frames in the shot is odd, the frame in the center of the shot is chosen as a keyframe; if the number of frames is even, any frame between the two frames in the middle of the shot can be chosen as the keyframe. When $TypeShot$ equals 1, the frame with the greatest difference is designated as the keyframe [23]. The graphical depiction of the four steps for extracting the keyframes is given in Figure 2.

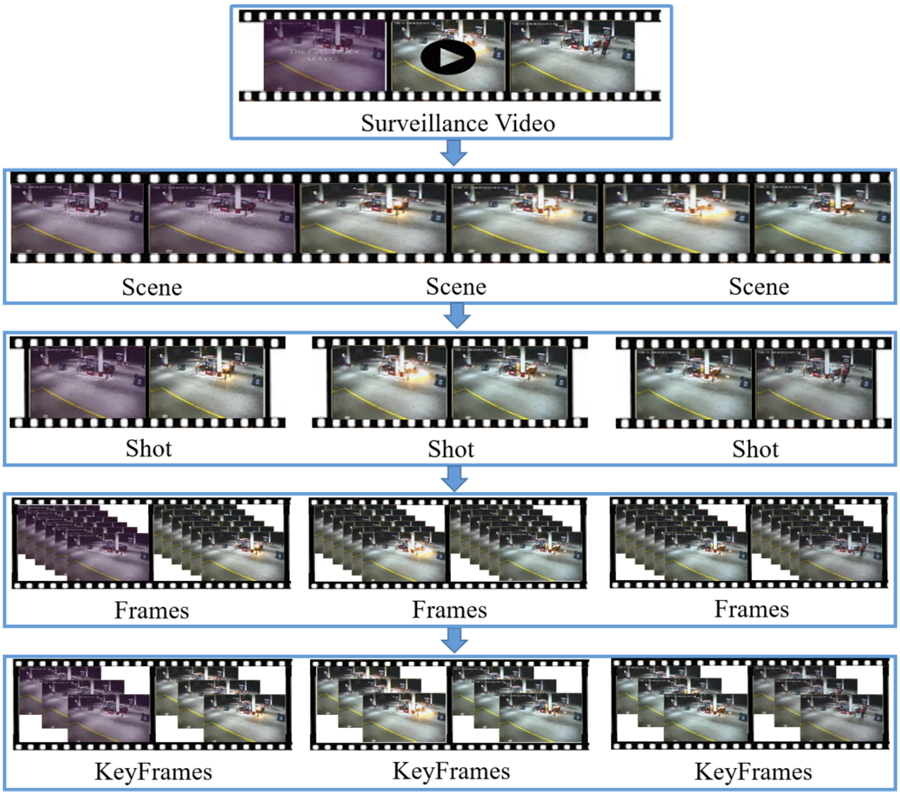


**Figure 2.** Steps of keyframe extraction from a given input video.

### 3.3. Proposed CNN Architecture

Our proposed CNN architecture ($LW_{CNN}$) consists of time distributed 2D convolutional layers (CL) and two time distributed 2D max-pooling layers, whose number of filters, kernel sizes, and strides are specified in Table 2. Each time distributed 2D CL is followed by the Rectified Linear Unit (ReLU) activation function with a kernel size of 3 × 3 and stride size of 2 × 2. After the second and third CL, it has a time distributed 2D max pooling layer with a stride size of 2 × 2 to reduce the network's size. We employ same-padding strategies for each convolutional operation to prevent skipping information at the input frame's border. As a result, the system produces feature maps that are the same size as the raw frame input of h × w. We begin with 64 feature maps in the first CL and then the same feature maps in the second. The final CL contains 128 feature maps. The proposed model receives a pre-processed frame of segmented shot as input. First, our system extracts spatial features from each frame by using ($LW_{CNN}$) and then feeds the sequence of spatial features into the LSTM to capture temporal features. For final prediction, the fully connected layer receives the output of the last time step of the LSTM layer and feeds it to the softmax layer [24], as seen in Figure 1. To capture the spatial features of each frame, we used a frame-wise $LW_{CNN}$, as seen in Figure 1. Furthermore, the input frames $(f_t, f_{t+1}, f_{t+n})$ are feed into an $LW_{CNN}$ individually, which transforms each frame into a sequence of spatial feature representation series as seen in Equation (9); $\mathcal{F}_t$ is a spatial feature representation series. However, the temporal features $\langle_{t+n}$ are computed using the spatial feature representation series as input to the LSTM network, as shown in Equation (10). The $\mathcal{H}_t$ is the hidden state of the LSTM layer at current time step $t$, and the hidden state of the previous time step $t-1$ is denoted by $\mathcal{H}_{t-1}$ [24]. The previous time step's information is fed as an input to the current time step. The output of the last time step of the hidden state $\mathcal{H}_{t+n}$ is passed into the next fully connected layer as an input [24]. The softmax layer receives input from the fully connected layer and calculates the final probability estimate for each class, as shown in Equation (11).

$$\mathcal{F}_t = LW_{CNN}(f_t, f_{t+1}, f_{t+n}) \tag{9}$$

$$\mathcal{H}_{t+n} = LSTM(\mathcal{F}_t) \tag{10}$$

$$Predication : y_j = softmax(\mathcal{H}_{t+n}) \tag{11}$$

**Table 2.** Description of $LW_{CNN}$ architecture used in EADN framework.

| Layer Type | Number of Filters | Size | Padding Value | Stride | Activation | Output Shape |
|---|---|---|---|---|---|---|
| Time Distributed Conv2D$_1$ | 64 | 3 × 3 | same | 2 × 2 | Relu | 5, 112, 112, 64 |
| Time Distributed Conv2D$_2$ | 64 | 3 × 3 | same | 2 × 2 | Relu | 5, 56, 56, 64 |
| Time Distributed MaxPooling2D$_1$ | 1 | 2 × 2 | - | 2 × 2 | - | 5, 28, 28, 64 |
| Time Distributed Conv2D$_3$ | 128 | 3 × 3 | same | 2 × 2 | Relu | 5, 14, 14, 128 |
| Time Distributed MaxPooling2D$_2$ | 1 | 2 × 2 | - | 2 × 2 | - | 5, 7, 7, 128 |
| Time Distributed Flatten$_1$ | - | - | - | - | - | 56272 |

## 4. Experimental Results

Three benchmark datasets, CUHK Avenue [4], UCSD Pedestrian [25], and UCF-Crime [5], are used for the evaluation of the EADN framework. Details about the number of videos, training–testing split, average frame, and types of anomalies are given in Tables 3 and 4. Sample frames of normal and anomalous events are given in Figure 3. Furthermore, the 3D visualizations of datasets are given in Figure 4. Two quantitative metrics, namely frame-based area under the curve (AUC) and receiver operating characteristic (ROC) [26], are used for the quantitative analysis. The algorithm is implemented in Keras backed Tensor Flow in Python version 3.7.4 programming environment. Experiments were done using a personal computer (PC) equipped with an NVidia GeForce GTX TITAN 1080 graphics-processing unit (GPU) with 8GB of RAM, a Windows 10 operating system, and the CUDA toolkit 9.0 with cuDNN v7.0. The quantitative results demonstrated the effectiveness of our EADN framework and showed a sustainable improvement in the state-of-the-art.

**Table 3.** The UCSD Pedestrian and Avenue dataset's statistical information

| Dataset | No. of Videos | Training Set | Test Set | Average Frames | Dataset Length | Example of Anomalies |
|---|---|---|---|---|---|---|
| UCSDPet1 | 70 | 34 | 36 | 201 | 5 min | Bikers, small carts, walking across walkways |
| UCSDPet2 | 28 | 16 | 12 | 163 | 5 min | Bikers, small carts, walking across walkways |
| Avenue | 37 | 16 | 21 | 839 | 5 min | Run, throw, new object |

**Table 4.** The UCF-Crime dataset's statistical information.

| Anomaly's Types | No. of Videos | Training Set | Test Set |
|---|---|---|---|
| Abuse | 50 | 48 | 2 |
| Arrest | 50 | 45 | 12 |
| Arson | 50 | 41 | 21 |
| Assault | 50 | 47 | 21 |
| Explosion | 50 | 29 | 21 |
| Fighting | 50 | 45 | 21 |
| Shooting | 50 | 27 | 21 |
| Shoplifting | 50 | 29 | 21 |
| Vandalism | 50 | 45 | 21 |
| Burglary | 100 | 87 | 21 |
| Stealing | 100 | 95 | 21 |
| Accident | 150 | 127 | 21 |
| Robbery | 150 | 145 | 5 |
| Total | 950 | 810 | 140 |

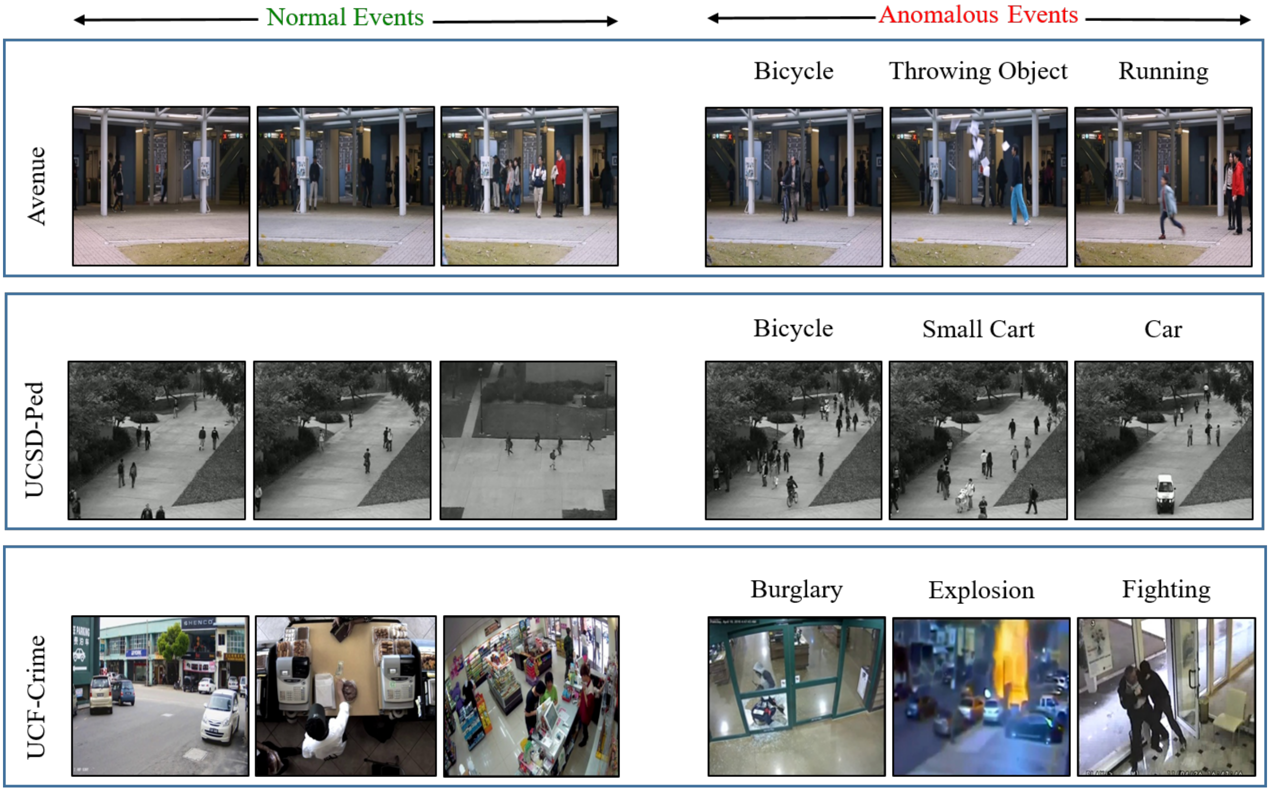


**Figure 3.** Sample of the 'Normal' and 'Abnormal' event frames from three datasets: Avenue, UCSD (Ped1 and Ped2), and UCF-Crime.

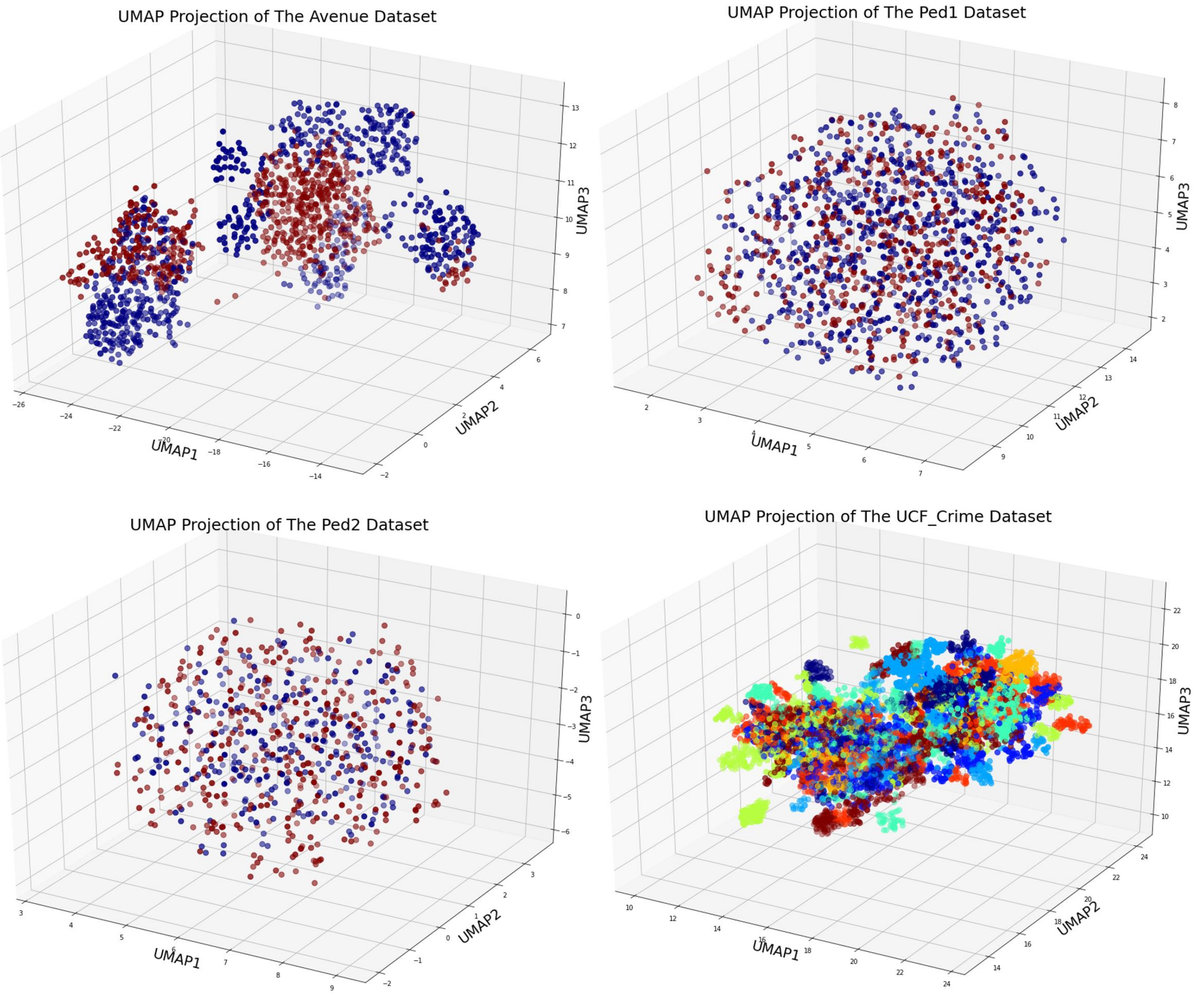


**Figure 4.** UMap data visualization of UCSD, CUHK, and UCF Crime datasets.

### *4.1. Quantitative Results*

UCSD pedestrian is a widely used dataset for anomaly detection in surveillance videos and is considered as a benchmark dataset. Similarly, the CUHK Avenue and UCF-Crime datasets are freely accessible and frequently used to evaluate video anomaly detection algorithms. We compare our EADN framework with both the earlier techniques [4,8,25,27] and the contemporary ones [15,16,28–31], including supervised and unsupervised strategies. Table 5 shows the quantitative comparison of the EADN against state-of-the-art methods in frame-based AUC values. EADN achieved a frame-based AUC of 93% and 97% for the UCSDped1 and UCSDped2 dataset, surpassing all the earlier methods. Similarly, on the CUHK Avenue dataset, EADN achieved 97% AUC compared to the best state-of-the-art of only 86.1%. Last but not the least, on the UCF-Crime dataset, EADN obtained 98% AUC, improving the state-of-the-art substantially. In Figure 5, we compared the frame-based ROC curve of our model with the state-of-the-art methods. Figure 6 plots the EADN's frame-based anomaly detection performance against the anomaly scores for test video sequences (a) Arrest048-x264, (b) Fighting047-x264, (c) Assault051-x264, and (d) Normal-

Videos-027-x264. Figures 7 and 8 illustrate the EADN's training accuracy and loss against the number of epochs.

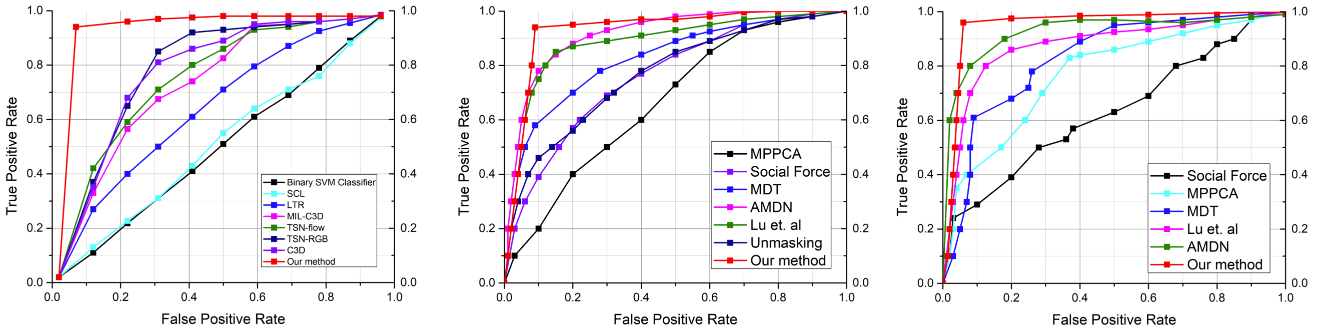
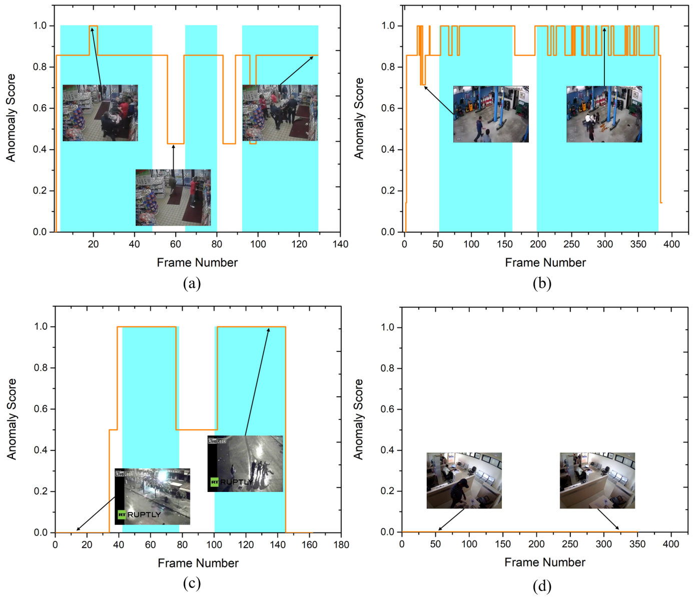


**Figure 5.** The proposed approach is compared to other existing approaches using the UCFcrime, UCSDped1, and UCSDped2 dataset. The ROC curves for state-of-the-art methods and our method are represented in the color codes.



(a)

(b)

(c)

(d)

**Figure 6.** The curves of anomaly score, for four test videos from the test set of the UCF-Crime dataset (**a**) Arrest048-x264, (**b**) Fighting047-x264, (**c**) Assault051-x264, and (**d**) Normal-Videos-027-x264. Cyan regions denote the ground truth abnormal frames. For better representation, we normalized anomaly scores for every video into the range of [0, 1]. This demonstrates that, as anomalies arise, the anomaly scores increase. (For best results, view in color).

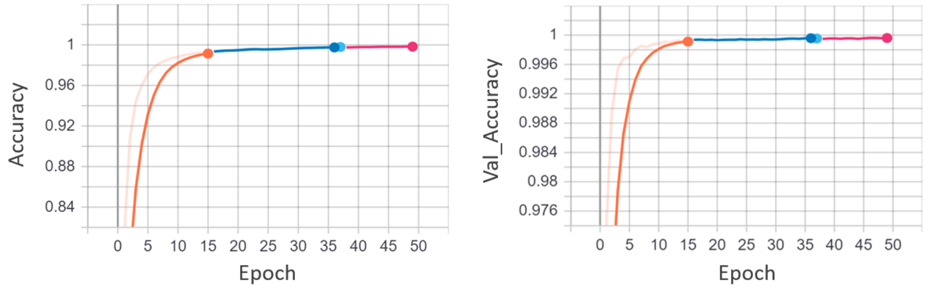


**Figure 7.** EADN training and validation accuracy graphs utilizing the UCF-Crime dataset. The training accuracy and the validation accuracy are plotted against the number of epochs. It is apparent from the graph that the model gets an optimal accuracy after 35 epochs. After 35 epochs, the model tends to overfit.

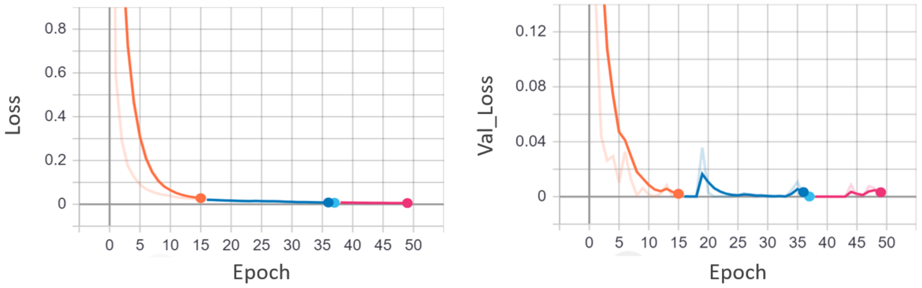


**Figure 8.** EADN training and validation loss graphs utilizing the UCF-Crime dataset. The training loss and the validation loss are plotted against the number of epochs. It is apparent from the graph that the model has a minimum loss after 35 epochs. After 35 epochs, the model tends to overfit.

**Table 5.** A frame-based AUC comparison of accuracies of the EADN framework against existing approaches based on the UCSDped1, UCSDped2, CUHK Avenue, and UCF crime datasets.

| Publication Year | Method | UCSDped1 [25] | UCSDped2 [25] | CUHK Avenue Dataset [4] | UCF-Crime Dataset [5] |
|---|---|---|---|---|---|
| 2009 | Kim and Grauman [27] | 59 | 69.3 | | |
| 2009 | Mehran et al. [8] | 67.5 | 55.6 | | |
| 2010 | Mahadevan et al. [25] | 81.1 | 82.9 | | |
| 2013 | Lu et al. [4] | 91.8 | - | | 65.51 |
| 2016 | Hasan et al. [15] | 81.0 | 90.0 | 70.2 | 50.6 |
| 2017 | Ionescu et al. [32] | 68.4 | 82.2 | 80.6 | |
| 2017 | Hinami et al. [28] | - | 92.2 | | |
| 2017 | Luo et al. [29] | - | 92.2 | 81.7 | |
| 2017 | Chong & Tay [33] | | | 80.3 | |
| 2017 | Xu et al. [31] | 92.1 | 90.8 | | |
| 2018 | Binary SVM classifier. [5] | | | | 50.0 |
| 2018 | MIL-C3D without constraints [5] | | | | 74.44 |
| 2018 | MIL-C3D with constraints [5] | | | | 75.41 |
| 2018 | Liu et al. [16] | 83.1 | 95.4 | 84.9 | |
| 2019 | Zhong et al. [22] | - | 93.2 | | 81.08 |
| 2019 | TSN-Optical Flow [22] | | 92.8 | | 78.08 |
| 2019 | Zhou et al. [34] | | | 86.1 | |
| 2019 | Spatiotemporal [35] | | | | 63.0 |
| 2019 | Zhou et al. [34] | 83.3 | 94.9 | 86.1 | |
| 2019 | Lee et al. [36] | | 96.6 | 90.0 | |
| 2020 | Zaheer et al. [37] | | 94.47 | | 79.54 |
| 2020 | Singh et al. [38] | 94.6 | 95.9 | 92.7 | |
| 2020 | Tang et al. [39] | 82.6 | 96.2 | 83.7 | |
| 2020 | Ganokratanaa et al. [40] | 98.5 | 95.5 | 87.9 | |
| 2021 | Maqsood et al. [41] | | | | 45.0 |
| 2021 | Ullah et al. [42] | | | | 85.53 |
| 2021 | Wu et al. [43] | 85.9 | 92.4 | | |
| 2021 | Qiang et al. [44] | 85.2 | 97.1 | 85.8 | |
| 2021 | Madan et al. [45] | | | 86.9 | |
| 2021 | Tian et al. [46] | | 96.5 | | 84.30 |
| 2022 | EADN (Ours) | 93.0 | 97.0 | 97.0 | 98.0 |

### 4.2. *Comparison with the State-of-the-Art Techniques*

Our EADN framework is compared with the existing strategies using the benchmark datasets. The authors of [42] examined a variety of deep learning models with the integration of multi-layer BD-LSTM, including VGG-19+multi-layer BD-LSTM, InceptionV3+multi-layer BD-LSTM, and ResNet-50 + multi-layer BD-LSTM. ResNet-50 combined with bidirectional-LSTM has the smallest model size among these approaches. The EADN framework for anomaly detection has a lower storage size, fewer learned parameters, and a faster processing time than other existing approaches [22,42,47–49]. The model's efficiency is compared to the recent techniques in terms of model size, time complexity, and parameter count, as seen in Tables 6 and 7. The quantitative results reveal that EADN framework has the lowest false alarm rates. Additionally, it can process a 32-frame sequence in 0.194 s, which is considerably faster than previous approaches [22,42,47–49]. As seen in Table 7, the sizes of existing approaches are significantly larger than the EADN.

**Table 6.** False alarm rate of the proposed EADN framework against state-of-the-art methods. In addition to the UCF-Crime dataset, EADN gives 0.08, 0.06, and 0.04 false alarm rates on the UCSDped1, UCSDped2, and the CUHK Avenue datasets.

| Method | UCF-Crime Dataset [5] |
|---|---|
| MIL-C3D with constraints [5] | 1.9 |
| Hasan et al. [15] | 27.2 |
| Lu et al. SCL [4] | 3.1 |
| C3D [22] | 2.8 |
| TSN-RGB [22] | 0.1 |
| TSN-Optical flow [22] | 1.1 |
| EADN (Ours) | 0.03 |

**Table 7.** Evaluation of EADN in terms of parameters, model size, and time complexity against the state-of-the-art methods.

| Method | Parameter Count (in million) | Model Size (MB) | Latency/Per Sequence (s) |
|---|---|---|---|
| C3D [22] | - | 313 | - |
| VGG-19+multi-layer BD-LSTM [47] | 143 | 605.5 | 0.22 |
| Inception V3+ multi-layer BD-LSTM [48] | 23 | 148.5 | - |
| ResNet-50 + multi-layer BD-LSTM [42,49] | 25 | 143 | 0.20 |
| EADN (Ours) | 14.14 | 53.9 | 0.20 |

## 5. Conclusions

This paper proposed a lightweight and cost-effective model for anomaly detection in surveillance videos that achieves promising accuracy on several benchmark anomaly detection datasets. The model works in three main steps: (i) Shot segmentation, (ii) feature extraction, and (iii) sequence learning and anomaly classification. In shot segmentation, salient shots are segmented using a shot boundary detection algorithm from the surveillance videos. Afterwards, we use our EADN framework to propagate a sequence of frames per sample of the salient shots and forward it to a lightweight Convolutional Neural Network ($LW_{CNN}$) that gets the spatiotemporal features from the intermediate layer. After that, LSTM cells learn the spatiotemporal features from a sequence of frames per sample of each anomalous event, which enables the EADN to classify anomalous events in the segmented shot of the video. Our model is validated using a variety of evaluation parameters and proved to be more accurate than recently published techniques. The experimental results demonstrate that EADN improves accuracy by 10.9%, 0.9%, 1.6%, and 15.88% for the Avenue, UCSD Pedestrian1, UCSD Pedestrian2, and UCF-Crime datasets, respectively, and significantly reduces false alarm rates as compared to recent works. However, our EADN framework still has room for real-time accuracy and efficiency improvement. In the future, we aim to incorporate the attention-based deep learning (self and multi-headed attention) networks to improve the accuracy and efficiency of the current EADN framework.

**Author Contributions:** S.U.A., M.U., M.S., F.A.C., M.H., A.H. and K.M. contributed to the conceptualization, analysis, validation, manuscript writing, and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received no external funding.

**Institutional Review Board Statement:** Not applicable.

**Informed Consent Statement:** Not applicable.

**Data Availability Statement:** Not applicable.

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. Piza, E.L.; Welsh, B.C.; Farrington, D.P.; Thomas, A.L. CCTV surveillance for crime prevention: A 40-year systematic review with meta-analysis. *Criminol. Public Policy* **2019**, *18*, 135–159. [CrossRef]
2. Cheng, K.W.; Chen, Y.T.; Fang, W.H. An efficient subsequence search for video anomaly detection and localization. *Multimed. Tools Appl.* **2016**, *75*, 15101–15122. [CrossRef]
3. He, C.; Shao, J.; Sun, J. An anomaly-introduced learning method for abnormal event detection. *Multimed. Tools Appl.* **2018**, *77*, 29573–29588. [CrossRef]
4. Lu, C.; Shi, J.; Jia, J. Abnormal event detection at 150 fps in matlab. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Sydney, Australia, 1–8 December 2013; pp. 2720–2727.
5. Sultani, W.; Chen, C.; Shah, M. Real-world anomaly detection in surveillance videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–22 June 2018; pp. 6479–6488.
6. Huo, J.; Gao, Y.; Yang, W.; Yin, H. Abnormal event detection via multi-instance dictionary learning. In *International Conference on Intelligent Data Engineering and Automated Learning*; Springer: Berlin/Heidelberg, Germany, 2012; pp. 76–83.
7. Zhang, D.; Gatica-Perez, D.; Bengio, S.; McCowan, I. Semi-supervised adapted hmms for unusual event detection. In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 20–25 June 2005; Volume 1, pp. 611–618.
8. Mehran, R.; Oyama, A.; Shah, M. Abnormal crowd behavior detection using social force model. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 935–942.
9. Nor, A.K.M.; Pedapati, S.R.; Muhammad, M.; Leiva, V. Abnormality detection and failure prediction using explainable Bayesian deep learning: Methodology and case study with industrial data. *Mathematics* **2022**, *10*, 554. [CrossRef]
10. Ullah, M.; Mudassar Yamin, M.; Mohammed, A.; Daud Khan, S.; Ullah, H.; Alaya Cheikh, F. Attention-based LSTM network for action recognition in sports. *Electron. Imaging* **2021**, *2021*, 302-1–302-6. [CrossRef]
11. Selicato, L.; Esposito, F.; Gargano, G.; Vegliante, M.C.; Opinto, G.; Zaccaria, G.M.; Ciavarella, S.; Guarini, A.; Del Buono, N. A new ensemble method for detecting anomalies in gene expression matrices. *Mathematics* **2021**, *9*, 882. [CrossRef]
12. Riaz, H.; Uzair, M.; Ullah, H.; Ullah, M. Anomalous Human Action Detection Using a Cascade of Deep Learning Models. In Proceedings of the 2021 9th European Workshop on Visual Information Processing (EUVIP), Paris, France, 23–25 June 2021; pp. 1–5.
13. Zhao, B.; Fei-Fei, L.; Xing, E.P. Online detection of unusual events in videos via dynamic sparse coding. In Proceedings of the CVPR 2011, Colorado Springs, CO, USA, 20–25 June 2011; pp. 3313–3320.
14. Luo, W.; Liu, W.; Gao, S. Remembering history with convolutional lstm for anomaly detection. In Proceedings of the 2017 IEEE International Conference on Multimedia and Expo (ICME), Hong Kong, China, 10–14 July 2017; pp. 439–444.
15. Hasan, M.; Choi, J.; Neumann, J.; Roy-Chowdhury, A.K.; Davis, L.S. Learning temporal regularity in video sequences. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 733–742.
16. Liu, W.; Luo, W.; Lian, D.; Gao, S. Future frame prediction for anomaly detection–a new baseline. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 6536–6545.
17. Chang, Y.; Tu, Z.; Xie, W.; Yuan, J. Clustering driven deep autoencoder for video anomaly detection. In *European Conference on Computer Vision*; Springer: Berlin/Heidelberg, Germany, 2020; pp. 329–345.
18. Sabokrou, M.; Khalooei, M.; Fathy, M.; Adeli, E. Adversarially learned one-class classifier for novelty detection. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 3379–3388.
19. Ullah, W.; Ullah, A.; Hussain, T.; Khan, Z.A.; Baik, S.W. An Efficient Anomaly Recognition Framework Using an Attention Residual LSTM in Surveillance Videos. *Sensors* **2021**, *21*, 2811. [CrossRef] [PubMed]
20. Tomar, D.; Agarwal, S. Multiple instance learning based on twin support vector machine. In *Advances in Computer and Computational Sciences*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 497–507.
21. Landi, F.; Snoek, C.G.; Cucchiara, R. Anomaly locality in video surveillance. *arXiv* **2019**, arXiv:1901.10364.
22. Zhong, J.X.; Li, N.; Kong, W.; Liu, S.; Li, T.H.; Li, G. Graph convolutional label noise cleaner: Train a plug-and-play action classifier for anomaly detection. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1237–1246.
23. Rathod, G.I.; Nikam, D.A. An algorithm for shot boundary detection and key frame extraction using histogram difference. *Int. J. Emerg. Technol. Adv. Eng.* **2013**, *3*, 155–163.
24. Zhang, D.; Yao, L.; Zhang, X.; Wang, S.; Chen, W.; Boots, R.; Benatallah, B. Cascade and parallel convolutional recurrent neural networks on EEG-based intention recognition for brain computer interface. In Proceedings of the AAAI Conference on Artificial Intelligence, New Orleans, LA, USA, 2–7 February 2018; Volume 32.
25. Mahadevan, V.; Li, W.; Bhalodia, V.; Vasconcelos, N. Anomaly detection in crowded scenes. In Proceedings of the 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 13–18 June 2010; pp. 1975–1981.

26. Li, W.; Mahadevan, V.; Vasconcelos, N. Anomaly detection and localization in crowded scenes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *36*, 18–32.
27. Kim, J.; Grauman, K. Observe locally, infer globally: A space-time MRF for detecting abnormal activities with incremental updates. In Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition, Miami, FL, USA, 20–25 June 2009; pp. 2921–2928.
28. Hinami, R.; Mei, T.; Satoh, S. Joint detection and recounting of abnormal events by learning deep generic knowledge. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 3619–3627.
29. Luo, W.; Liu, W.; Gao, S. A revisit of sparse coding based anomaly detection in stacked rnn framework. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 341–349.
30. Ravanbakhsh, M.; Nabi, M.; Mousavi, H.; Sangineto, E.; Sebe, N. Plug-and-play cnn for crowd motion analysis: An application in abnormal event detection. In Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), Lake Tahoe, NV, USA, 12–15 March 2018; pp. 1689–1698.
31. Xu, D.; Yan, Y.; Ricci, E.; Sebe, N. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Comput. Vis. Image Underst.* **2017**, *156*, 117–127. [CrossRef]
32. Tudor Ionescu, R.; Smeureanu, S.; Alexe, B.; Popescu, M. Unmasking the abnormal events in video. In Proceedings of the IEEE International Conference on Computer Vision, Venice, Italy, 22–29 October 2017; pp. 2895–2903.
33. Chong, Y.S.; Tay, Y.H. Abnormal event detection in videos using spatiotemporal autoencoder. In *International Symposium on Neural Networks*; Springer: Berlin/Heidelberg, Germany, 2017; pp. 189–196.
34. Zhou, J.T.; Du, J.; Zhu, H.; Peng, X.; Liu, Y.; Goh, R.S.M. Anomalynet: An anomaly detection network for video surveillance. *IEEE Trans. Inf. Forensics Secur.* **2019**, *14*, 2537–2550. [CrossRef]
35. Gianchandani, U.; Tirupattur, P.; Shah, M. *Weakly-Supervised Spatiotemporal Anomaly Detection*; University of Central Florida Center for Research in Computer Vision REU: Orlando, FL, USA, 2019.
36. Lee, S.; Kim, H.G.; Ro, Y.M. BMAN: Bidirectional multi-scale aggregation networks for abnormal event detection. *IEEE Trans. Image Process.* **2019**, *29*, 2395–2408. [CrossRef] [PubMed]
37. Zaheer, M.Z.; Mahmood, A.; Shin, H.; Lee, S.I. A self-reasoning framework for anomaly detection using video-level labels. *IEEE Signal Process. Lett.* **2020**, *27*, 1705–1709. [CrossRef]
38. Singh, K.; Rajora, S.; Vishwakarma, D.K.; Tripathi, G.; Kumar, S.; Walia, G.S. Crowd anomaly detection using aggregation of ensembles of fine-tuned convnets. *Neurocomputing* **2020**, *371*, 188–198. [CrossRef]
39. Tang, Y.; Zhao, L.; Zhang, S.; Gong, C.; Li, G.; Yang, J. Integrating prediction and reconstruction for anomaly detection. *Pattern Recognit. Lett.* **2020**, *129*, 123–130. [CrossRef]
40. Ganokratanaa, T.; Aramvith, S.; Sebe, N. Unsupervised anomaly detection and localization based on deep spatiotemporal translation network. *IEEE Access* **2020**, *8*, 50312–50329. [CrossRef]
41. Maqsood, R.; Bajwa, U.I.; Saleem, G.; Raza, R.H.; Anwar, M.W. Anomaly recognition from surveillance videos using 3D convolution neural network. *Multimed. Tools Appl.* **2021**, *80*, 18693–18716. [CrossRef]
42. Ullah, W.; Ullah, A.; Haq, I.U.; Muhammad, K.; Sajjad, M.; Baik, S.W. CNN features with bi-directional LSTM for real-time anomaly detection in surveillance networks. *Multimed. Tools Appl.* **2021**, *80*, 16979–16995. [CrossRef]
43. Wu, C.; Shao, S.; Tunc, C.; Satam, P.; Hariri, S. An explainable and efficient deep learning framework for video anomaly detection. *Clust. Comput.* **2021**, 1–23. [CrossRef] [PubMed]
44. Qiang, Y.; Fei, S.; Jiao, Y. Anomaly detection based on latent feature training in surveillance scenarios. *IEEE Access* **2021**, *9*, 68108–68117. [CrossRef]
45. Madan, N.; Farkhondeh, A.; Nasrollahi, K.; Escalera, S.; Moeslund, T.B. Temporal Cues from Socially Unacceptable Trajectories for Anomaly Detection. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 2150–2158.
46. Tian, Y.; Pang, G.; Chen, Y.; Singh, R.; Verjans, J.W.; Carneiro, G. Weakly-supervised video anomaly detection with robust temporal feature magnitude learning. In Proceedings of the IEEE/CVF International Conference on Computer Vision, Montreal, BC, Canada, 11–17 October 2021; pp. 4975–4986.
47. Simonyan, K.; Zisserman, A. Very deep convolutional networks for large-scale image recognition. *arXiv* **2014**, arXiv:1409.1556.
48. Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; Wojna, Z. Rethinking the inception architecture for computer vision. In Proceedings of the IEEE Conference on Computer Vision and pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 2818–2826.
49. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.