Enhancing Video Anomaly Detection Using Spatio-Temporal Autoencoders and Convolutional LSTM Networks

Original Research Published: 11 January 2024

Volume 5, article number 190, (2024) Cite this article

Ghayth Almahadin

Department of Networks and Cybersecurity, Faculty of Information Technology, Al Ahliyya Amman University Country, Amman, Jordan

View author publications

You can also search for this author in PubMed | Google Scholar

Ghayth Almahadin, Maheswari Subburaj ✉, Mohammad Hiari, Saranya Sathasivam Singaram, Bhanu Prakash Kolla, Pankaj Dadheech, Amol D. Vibhute & Sudhakar Sengan ✉

Abstract

Identifying suspicious activities or behaviors is essential in the domain of Anomaly Detection (AD). In crowded scenes, the presence of inter-object occlusions often complicates the detection of such behaviors. Therefore, developing a robust method capable of accurately detecting and locating anomalous activities within video sequences becomes crucial, especially in densely populated environments. This research initiative

aims to address this challenge by proposing a novel approach focusing on AD behaviors in crowded settings. By leveraging a spatio-temporal method, the proposed approach harnesses the power of both spatial and temporal dimensions. This enables the method to effectively capture and analyze the intricate motion patterns and spatial information embedded within the continuous frames of video data. The objective is to create a comprehensive model that can efficiently detect and precisely locate anomalies within complex video sequences, specifically those featuring human crowds. The efficacy of the proposed model will be rigorously evaluated using a benchmark dataset encompassing diverse scenarios involving crowded environments. The dataset is designed to simulate real-world conditions where millions of video footage need to be continuously monitored in real time. The focus is on identifying anomalies, which might occur within short time frames, sometimes as brief as five minutes or even less. Given the challenges posed by the massive volume of data and the requirement for rapid AD, the research emphasizes the limitations of traditional Supervised Learning (SL) methods in this context.

Introduction

With technological advancements and the proliferation of surveillance cameras in various public settings, ensuring the safety and well-being of individuals has become a primary

concern. In this context, AD behaviors have emerged as a critical area of research within surveillance systems and Computer Vision (CV) [1,2,3,4,5]. While Anomaly Detection (AD) has undergone extensive investigation over the past decade, there remains ample scope for further refinement, particularly in extended video surveillance. The complexity of distinguishing anomalies is compounded by the influence of contextual factors, where normal behavior in one setting may appear aberrant in another. Given the challenge of annotating diverse events, constructing a comprehensive model encompassing all potential anomalies proves impractical [6,7,8,9,10]. To address these challenges, this research implements an automated framework for detecting and segmenting pertinent sequences, leveraging a Deep Learning (DL) approach for efficient feature extraction and representation from video data [11,12,13,14,15]. By harnessing the capabilities of convolutional autoencoders and temporal autoencoders, this study seeks to improve AD efficiency, enabling the identification of irregular patterns within vast volumes of video footage in a timely and accurate manner. Any dataset of videos will be significant; the videos will have noise associated with them and hold massive events and exchanges.

Additionally, context plays a significant role in the interpretation of anomaly. For instance, running around freely in a park is perfectly normal behaviour, whereas doing the same thing in a restaurant is likely to be regarded as highly unusual [16,17,18,19]. The concept of an anomaly is notoriously tricky to pin down and define precisely. When the scope is delineated, AD functions most effectively.

Regarding successful AD recognition in the past, the primary factor was labelled video footage with clearly defined events of interest. This footage mainly focuses on defining the events; hence, sequences, including crowded locations, were scarce. It would be prohibitively expensive to label every possible event type. In short, learning a model constituting all the abnormal and irregular events is impossible [20,21,22,23,24,25].

In this work, using an automated framework for detecting and segmenting sequences of interest, we aim to improve upon earlier developed labelling methods while reducing the labor-intensive effort required. The video data are represented using a set of standard features implied automatically from extended video footage with the help of a Deep Learning (DL) approach. Processing video frames unsupervised can be made easier with a Deep Neural Network (DNN) built on convolutional autoencoders. They identify spatial structures in data. These structures are then grouped to compose the proposed video representation. After confirming spatial structures, the representation is sent to the

temporal autoencoders to learn regular temporal patterns. Feature Extraction (FE) [26,27,28] representation of video is combined with learning those feature models. The ability to automatically fine-tune the model according to the video allows us to be domain-free and reduce computational time.

AD helps us to find data points that are chaotic and unexpected. The primary focus is to distinguish regular patterns from anomalous ones [29,30,31]. The solution to this problem might come quickly as Binary Classification (BC), and many might think that they can solve it using any Supervised Learning (SL) algorithm, but the classes can be highly imbalanced [32].

Millions of minutes of footage need to be reviewed in real-time, out of which there could be an anomaly for even 5 min or fewer. SL would suffer in this scenario. However, Autoencoders can be perfect as we can train them on normal parts and not use annotated data. The output of the autoencoder, when compared to the input, will give us a difference based on FE. The more distinct and significant the difference, the higher the chances of the input holding an anomaly.

The paper follows a structured organization comprising of "Introduction" describing an introduction, "Related works" discussing the related works, "Proposed methodology" proposing a methodology encompassing dataset details and various techniques, "Results and discussion" discusses the results and discussions, and "Conclusion and future work" concludes with future research directions. It maintains a well-defined format for clarity and comprehensibility.

Related Works

The article "Robust Real-Time Unusual Event Detection Using Multiple Fixed-Location Monitors" presented a method for identifying specific types of events that are not typical [33,34,35]. Their algorithm relies on a collection of multiple local monitors that each collect low-level data analysis as its starting point. If any one of the local monitors detects something out of the ordinary with their measurements, an alarm will sound [36,37,38]. "Abnormal Event Detection at 150 FPS in MATLAB" provided a method for detecting abnormal events based on using sparse-combination learning [39, 40] so that we can accelerate the testing process without compromising the accuracy [41,42,43,44].

In contrast to modern CNN models, this paper uses multiple layers of cells without a shared weight for unsupervised visual pattern recognition. Motivated by the animal visual cortex 50 years ago, CNN excels at image and video spatial feature extraction [45].

An approach to learning periodic patterns with autoencoders and minimal supervision. The research paper "Learning temporal regularity in video sequences" [4,5,6] implemented a fully convolutional autoencoder to learn local features and classifiers within the context of a single learning framework [46,47,48,49].

The purpose of this article by [50], "AD in crowded scenes," is to present a framework for AD in crowded scenes. They used an MDT-based model in their implementation. "Abnormal crowd behaviour detection using social force model" is the title of the author's paper in which they developed a model to detect and localize abnormal behaviors in crowd scenes using the social force model [51]. Using a bag of words methodology, they classified frames as normal or abnormal [52].

The author researched CV and Pattern Recognition (PR) for their "Real-Time AD and Localization in Crowded Scenes" paper. In crowded scenes, they suggested a technique for real-time AD and localization. [53] We defined each video as a collection of non-overlapping cubic patches and used local and global descriptors to describe them.

The objective of the research titled "Enhancing Video AD Using Spatiotemporal Autoencoders and Convolutional LSTM Networks" is to improve the efficiency and accuracy of AD within video data by integrating the capabilities of Spatiotemporal Autoencoders and Convolutional Long Short-Term Memory (LSTM) networks. The aim is to develop a robust and sophisticated framework that can effectively capture spatial and temporal information within video sequences, enabling the identification and localization of anomalous activities with higher precision in various surveillance and security applications [54].

Table 1 provides an overview comparison of the numerous methodologies discussed in this chapter.

Table 1 Analyses of the differences between various methods of feature processing

Proposed Methodology

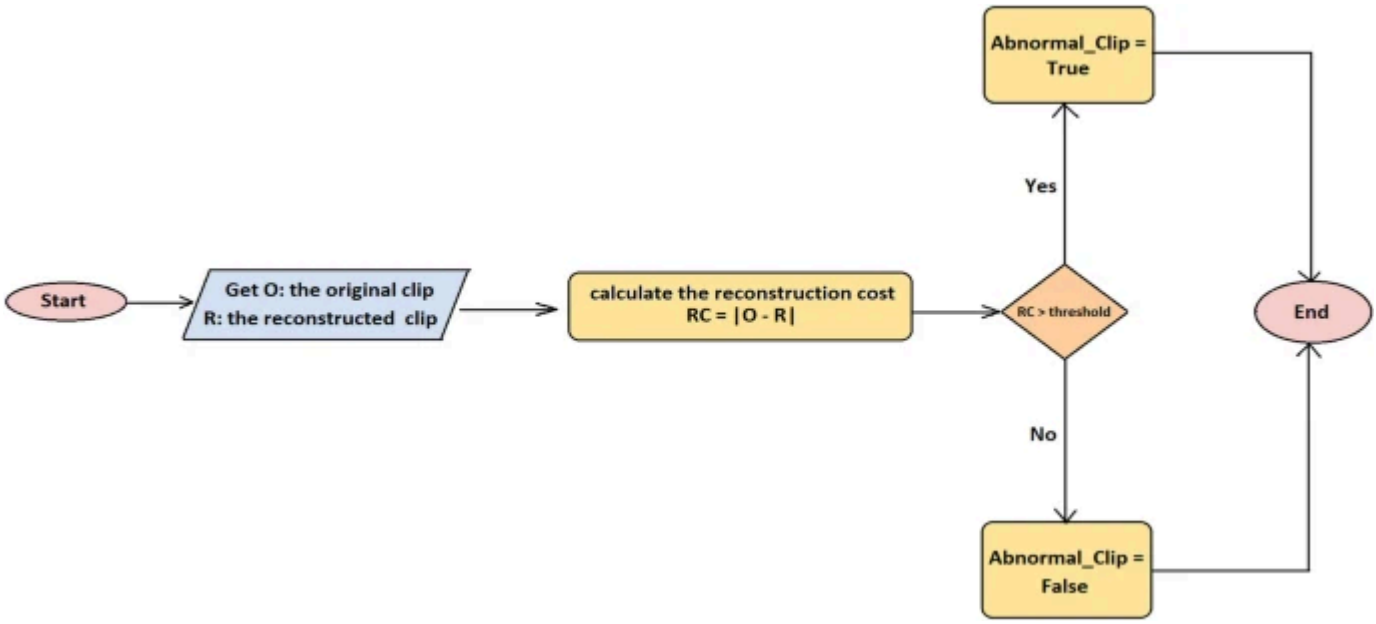
Dataset

This work used the USCD-AD dataset to train and test this proposed model. The videos included in this dataset were recorded using a fixed camera placed at a height that gave it a view that extended over walkways used by pedestrians. There are times when these walkways are not very crowded at all, and there are also times when the crowd density is at an all-time high. Only pedestrians are permitted at normal events. Abnormal occurrences can be attributed to either:

- Golf carts, bicycles, and skateboards are examples of non-pedestrian entities permitted to use the walkways.
- Certain motion patterns of pedestrians that were not observed Dataset link—UCSDped1 [https://drive.google.com/drive/folders/1JJY8FzXjVgysOlaTqHvmoSEPdUhu_VUZ?usp=sharing].

The following flow chart in Fig. 1 defines this model pipeline for AD.

Fig. 1



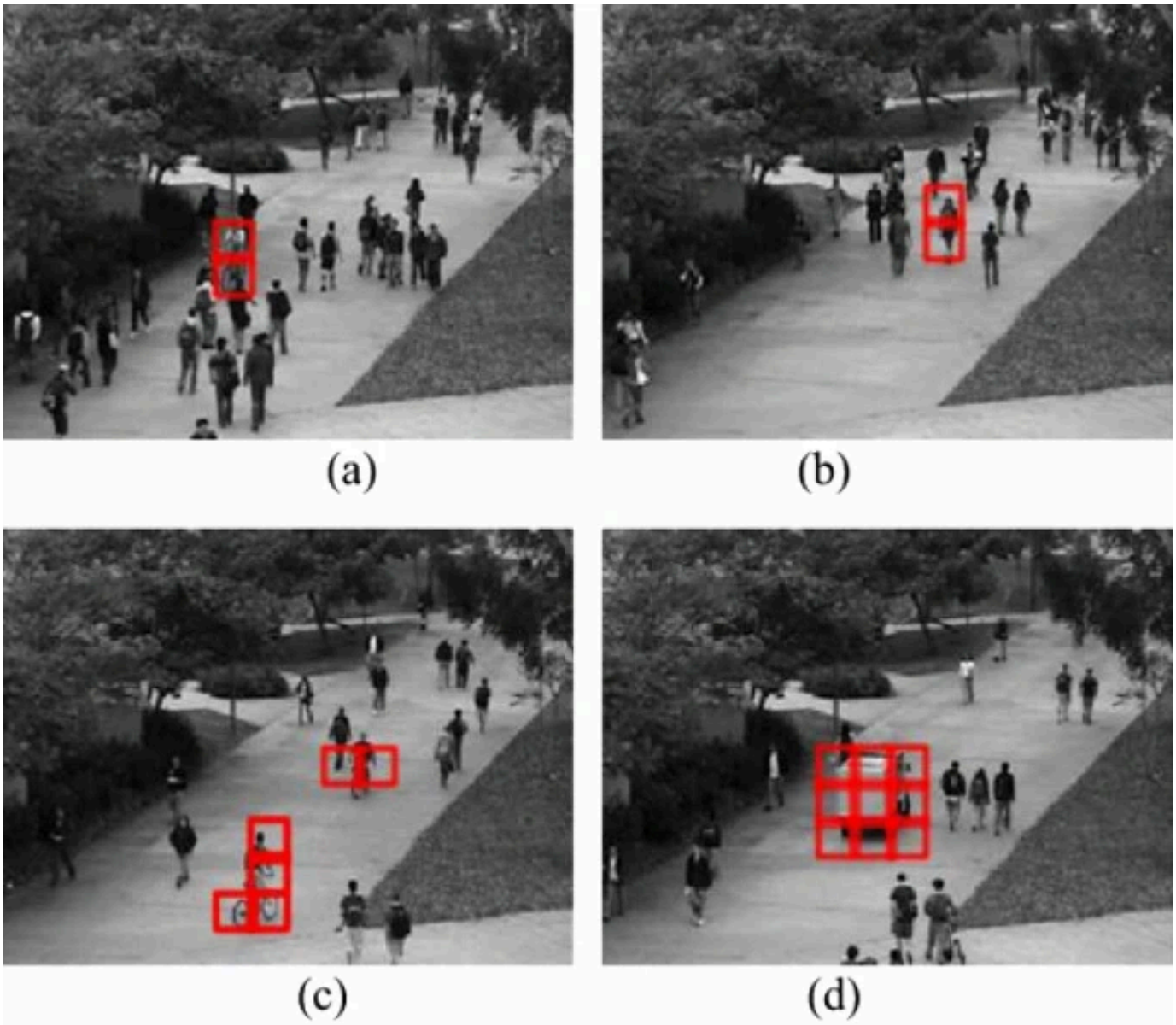
Proposed architecture diagram

The problem of AD in videos can be considered a Binary Classification (BC) issue. It requires labelled data, which is difficult to collect for the reasons below.

1. Because of their scarcity, abnormal occurrences are notoriously difficult to document.
2. There are a considerable number of unusual events. The detection and labelling of such events manually are a mammoth task that requires many resources to complete successfully.

In Fig. 2, unsupervised methods, such as autoencoders and spatio-temporal features, were favored. These approaches outshine their supervised counterparts as they solely rely on unlabelled video evidence containing few or no abnormal activities, readily available in real-life scenarios.

Fig. 2



Samples from the dataset. The red boxes define the anomaly

The preprocessing steps for the UCSD AD dataset were paramount in optimizing raw video data for model input and enhancing overall model performance. First, videos were uniformly resized to a consistent spatial resolution, typically 227 × 227 pixels or other specified sizes, ensuring dataset-wide uniformity. This step maintains compatibility with fixed input dimensions. Next, temporal subsampling of video frames created coherent input sequences to capture critical temporal dependencies, ultimately improving model convergence and generalization. Data augmentation techniques, including random horizontal flipping, random cropping, and minor adjustments in brightness and contrast, were employed to diversify the training dataset and bolster model robustness. Lastly, pixel values were normalized to a specific range, stabilizing training and facilitating

model convergence. These pre-processing steps prove indispensable in enhancing the model's performance in video AD tasks.

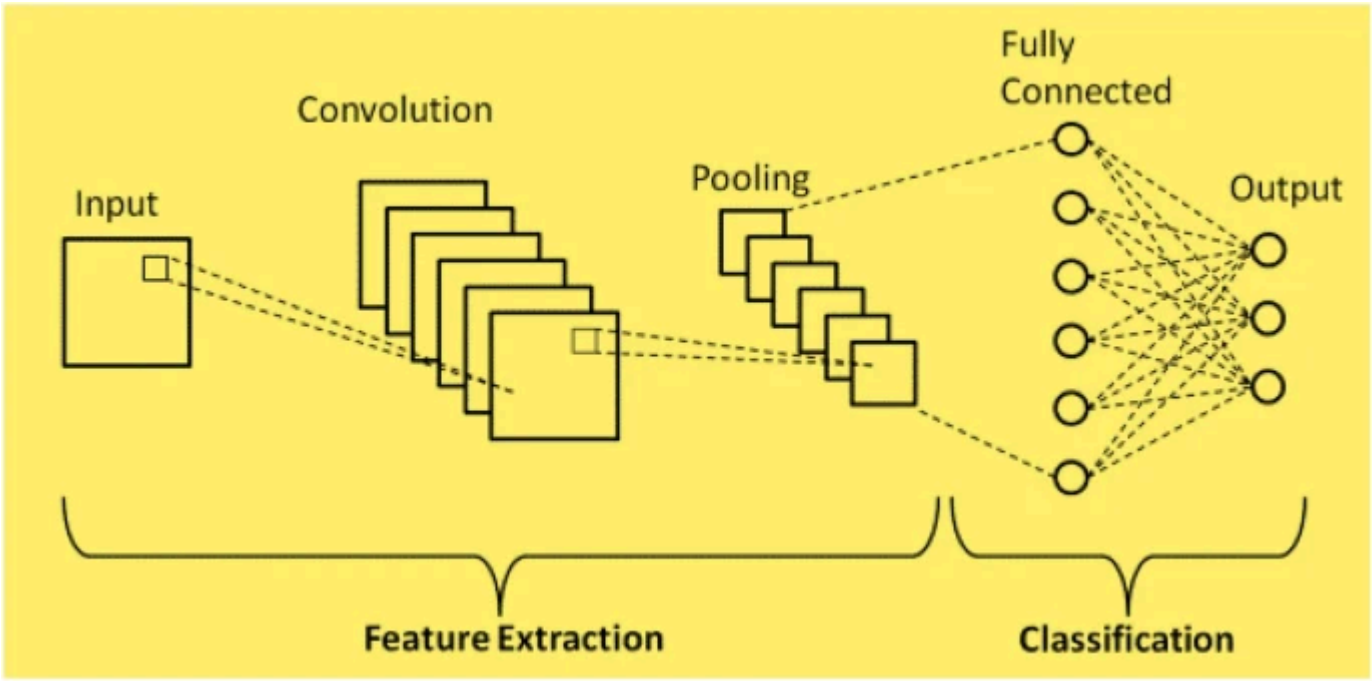
Convolution Neural Network (CNN)

A CNN is a Deep Learning (DL) algorithm that takes its input in the form of an image and assigns importance to technical approaches or objects in the image by updating its weights and biases in such a case that the objects that are present can differ significantly from one another and then outputs the results of this method.

Convolutional Neural Network (CNN) is constructed to resemble the connection patterns of the Neurons present in the human brain. Because it allows for a reduction in the total range of parameters and the reusability of weights, this layout is the one that works best for an image dataset. The network can understand the minute details of any image better. CNN condenses images into a format that is simpler to process while simultaneously preserving the details with the highest possible degree of accuracy. By employing the appropriate FE methods, a CNN can successfully capture the Spatial–Temporal dependencies that exist in an image.

The convolutional layer is the preliminary step in constructing CNN in Fig. 3, which comprises several layers in total. It is the core building block, and most of the computations are carried out by it. This work uses filters or kernels to convolve data or images. Filters are applied across the data. It is implemented through a sliding window. For each sliding action, the element-wise product of the image's filters is computed, and the result is added together. The result of a convolution employing a 3-D filter with color will be a 2-D matrix in its representation.

Fig. 3



Architecture of a CNN

Rectified Linear Unit (ReLU) is the name of the function that the subsequent Activation Layer (AL) uses. Applying the rectifier function at this stage will increase the network's level of non-linearity. This is necessary because images are constructed from features that are frequently not linear to one another.

The Pooling Layer (PL) comes after the AL and involves a down-sampling of features. This layer follows the AL. The 3-D volume performs down-sampling on each layer. This layer includes the hyperparameters for the dimensions of spatial extent and stride, and they can be found here. The dimension of spatial extent is the value of '*n*', which can take '*N*' representations of crosses and features and map them to a single value. The sliding window will skip a certain number of features along the width and the height depending on the "stroke," which is the number of features it will skip. The Fully Connected (FC) layer is the final layer, and it is the one that produces the output.

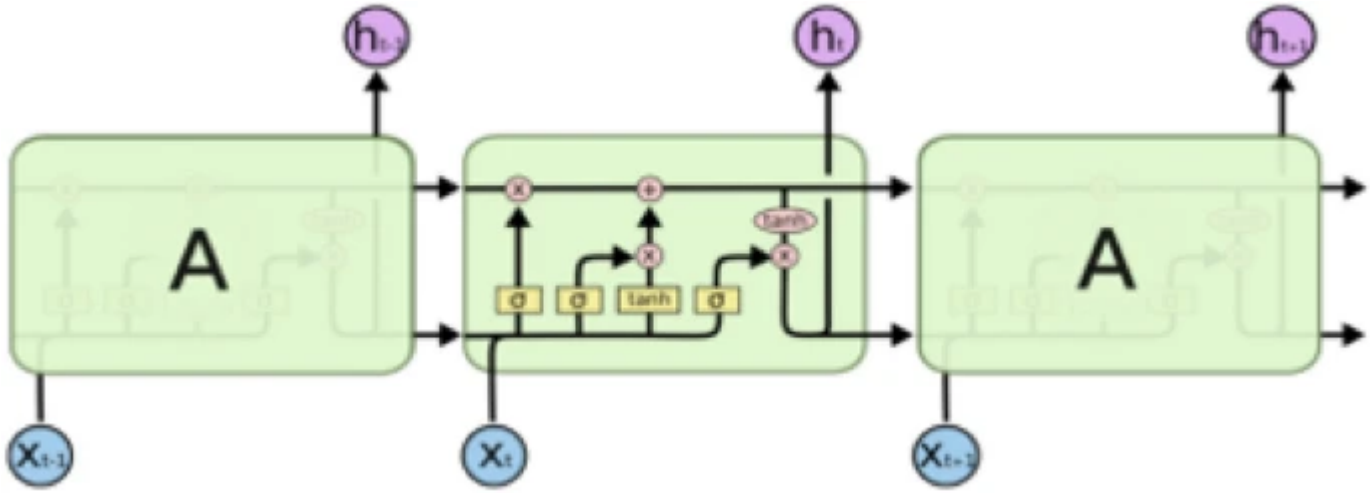
The input has been transformed into a single column because of this work, which flattens the input, which is the pooled feature map matrix in its entirety. Following that, the Neural Network (NN) will begin processing this information. This research combines all these features to create a model. This work uses the Sigmoid or SoftMax activation function to classify the output. The deconvolutional layers densify the sparse signals. Multiple learned filters are used to perform operations that are like convolution. By

operating in reverse convolution, they could link a single input activation to patch outputs.

Long Short-Term Memory (LSTM)

In DL, a Recurrent Neural Network (RNN) known as an LSTM in Fig. 4 has been developed specifically to address issues related to sequential prediction. In contrast to simple recurrent networks, LSTMs are not hindered by the optimization challenges that prevent them from capturing long-term temporal dependencies. The structure of an LSTM is like that of a chain, consisting of four layers that interact with one another.

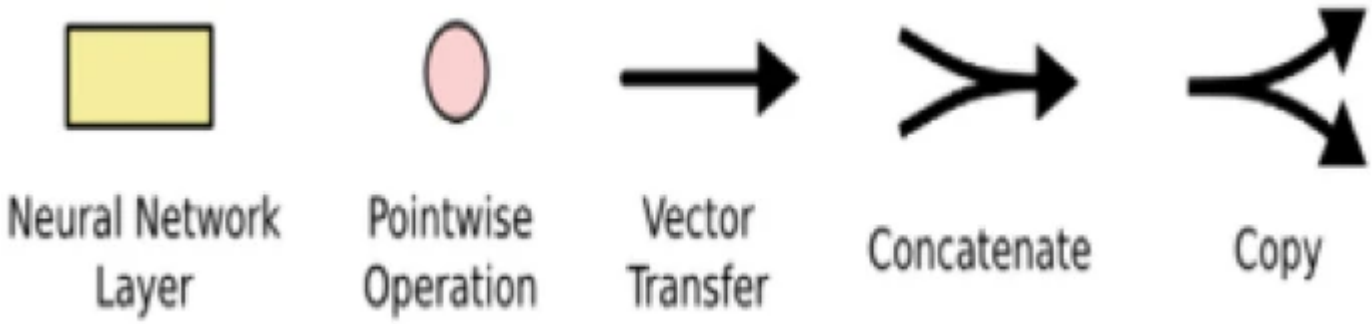
Fig. 4



LSTM architecture

A full vector is transmitted along each line from the node's output to the inputs of other nodes. A cell state with only a single insignificant interaction is held by the LSTM, the horizontal line that runs through the top of the diagram. Unaltered information can travel along it. Using only these regulated gates, LSTM could either remove information from the cell state or add information to it. Each cell state is protected and controlled by three of these gates in the LSTM. Figure 5 shows the legends for the LSTM model.

Fig. 5



Legend for the LSTM model

Working of LSTM

A. Stage 1: Forget Gate (FG) Layer: It is made by a sigmoid layer. It decides what information should be removed from the cell state. Considerations of h_{t-1} and x_t are made, and a number between 0 and 1 are returned. It is done for each number in the cell state C_{t-1}. 1 stands for "total retention," while a 0 represents "no retention", Eq. (1):

$$f_{t} = \sigma (W_{f} \cdot [h_{t - 1} ,x_{t}] + b_{f}).$$

(1)

B. Stage 2: Input Gate (IG) Layer: It is made by a sigmoid and a *tanh* layer. It determines what new data should be saved as part of the cell's current state. The sigmoid layer determines the values we will update. A set of candidate values, C_t, for the state is produced by the *tanh* layer. To generate an update to the step [Eq. (2)], the two decisions are consolidated into one.

$$\begin{gathered} i_{t} = \sigma (W_{i} \cdot [h_{t - 1} ,x_{1}] + b_{i} \hfill \\ \widehat{{C_{t} }} = tanh(Wc \cdot [h_{t - 1} ,x_{1}] + b_{c} \hfill \\ \end{gathered}$$

(2)

Following the input gate layer, we must transition from the previous cell state C_{t1} to the subsequent C_t.

We multiply the previous state by f_t but forget what we decided to let go of. After that, we multiply by $(i_t * C_t)$. These are the new candidate values, and their magnitude is determined by the degree to which we would like each state value to be updated, Eq. (3):

$$C_{t} = f_{t} \times C_{t - 1} + i_{t} \times \widetilde{{C_{t} }}.$$

(3)

C. Stage 3: ***Final Filtered Output:*** A sigmoid layer is responsible for its generation. This layer determines which aspects of the cell state will be included in the output. After the sigmoid gate's output is settled, the cell state is multiplied by the result of *tanh* to move the values to the range [− 1, 1]. As a result, we only output the portion that we choose to output, Eq. (4):

$$\begin{gathered} o_{t} = \alpha (W_{0} [h_{t - 1} ,x_{1}] + b_{0} , \hfill \\ h_{t} = o_{t} \times tanh(C_{t}). \hfill \\ \end{gathered}$$

(4)

D. Autoencoder: One device that combines the functions of an encoder and a decoder into a single unit is known as an autoencoder. After receiving the input, the encoder will encode the data utilizing a reduced representation. On the other hand, the decoder works to recreate the initial input using the encoded version of the reduced representation. The Autoencoder is required to learn a sparse representation of the training data due to the constraints imposed by the network. Because it is a UL method, the autoencoder is the best algorithm for dealing with this AD issue.

E. Working of the Encoder–Decoder

(i) **The Encoder:** Learning representations of the input data (x), known as the encoding f, is where it shines (x). The name for the encoder's final layer, known as the bottleneck, comes from its function. When f is the final input representation, this holds (x).

(ii) **The Decoder:** By introducing the use of the encoding that is presented in the bottleneck, it creates a reconstruction of the input data denoted by $r = g(f(x))$.

The autoencoder was used in this study to learn patterns of regularity in video sequences. The trained autoencoder is thought to reconstruct regular video sequences with low error but fails to accurately reconstruct motions in irregular video sequences. This distinction will aid in the identification of anomalies.

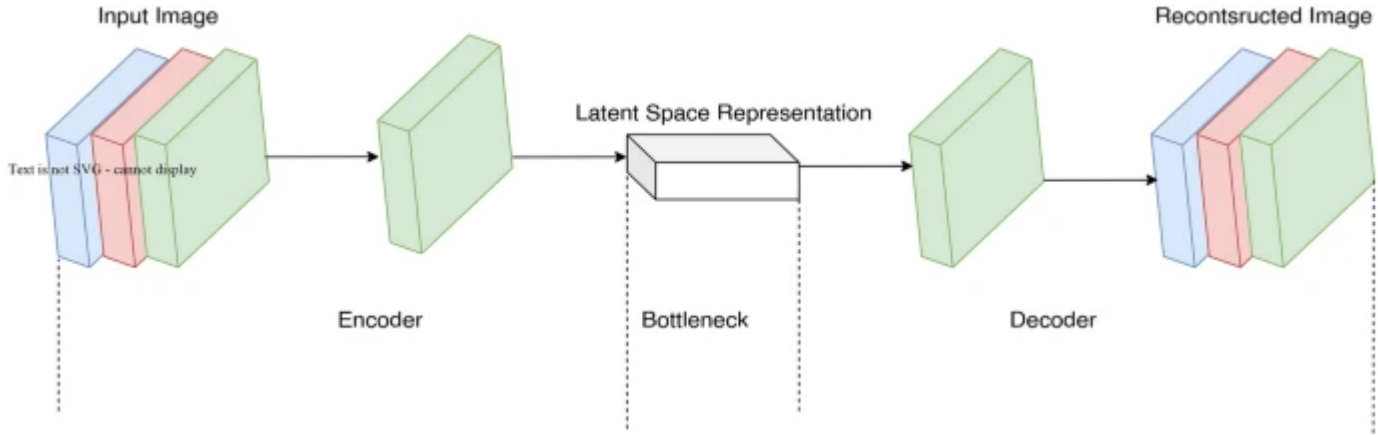
F. Convolutional Autoencoder—CAE

The train dataset contains images that are in the 158 × 238-pixel format. These pictures have had their dimensions normalized and rescaled to be exactly 100 × 100. This dataset, along with the batch size and the shuffle factor, is loaded into the function known as the data loader.

The encoder comprises two convolutional layers, each with a kernel size 32 and 2-MaxPooling2D layers. The encoder and the decoder are connected by a dense layer of FC, containing 2000 neurons that are not visible to the naked eye. The encoder and decoder are connected via this layer. The greater the size of this bottleneck, the more information can be reconstructed, which opens the possibility of recognizing minute details. The decoder produces the final signal using a 1-neuron output layer, 32-kernel deconvolution layers, and 2-up sampling layers. In the output layer of the algorithm is where the implementation of the Sigmoid function can be found.

In Fig. 6, the network is initialized using the Xavier algorithm. Adam optimizer is used, and the learning rates are defined. The batch size for this autoencoder is 32, and it has been trained for 30 epochs. The trained model is kept for future use and is subsequently called upon for testing. The test dataset is also resized equivalently before testing. Resizing data is not mandatory; the autoencoder can handle variable input sizes, but having a uniform dataset makes the computation relatively easy.

Fig. 6



Architecture of convolutional autoencoder

Test images are iterated over while the difference between input and output is calculated. The differences motivate the development of a 4 × 4 convolution kernel pixel map. It is an anomaly if the pixel value is higher than 1020, equal to 255 times four. The highest possible value for each pixel equals 4 × 4 times 255. They will be marked only when the pixels' neighbouring pixels are also abnormal.

Autoencoder has moderately improved pedestrian reconstruction but has trouble with novel objects. The value of information transmitted between the encoder and the decoder will be determined by the dimension of the bottleneck layer. Quite good image reconstruction occurs if we make it too big or remove it altogether.

The same model was defined in this work; however, the dense layer was eliminated and trained once more. It is much easier for the network to reconstruct pedestrians and other objects without a bottleneck layer, but it is now more difficult to spot anomalies.

In the training process for the model, the choice of optimizer was Adam, a widely used optimization algorithm with a fixed learning rate of 0.001. A learning rate schedule was incorporated, reducing the learning rate by a factor of 0.1 if the validation loss plateaued for a predefined number of epochs, allowing dynamic adjustment as training progressed. Convergence criteria were set based on monitoring the validation loss; training was terminated early if the loss did not exhibit significant improvement for a specified number of epochs to prevent overfitting. These training specifics are crucial for replicating the results and laying the groundwork for further advancements in video AD.

G. Spatio-Temporal Stacked Frame Autoencoder—STAE

The typical CAE ignores the timeliness of images in a sequence. The motion of a person walking or running behind other people, in addition to the motion of a bicycle or golf cart, is challenging to detect. Instead of one image at a time, researchers now consider '*n*' images. Difference between CAE and STAE input format: [batch size, 1, width, height] vs. [batch size, n, width, height].

The original input dataset must be updated to include '*n*' channels rather than just 1. Since this model uses many parameters, it needs a good deal of training data. This research uses temporal data augmentation to create many images for training. Concatenating frames with multiple skipping strides allow us to generate more training sequences. A succession of frames can be fed into the encoder as its input source. Encoder architecture consists of the spatio-temporal encoders working together. They are called in the above order, one after the other. After being encoded, the sequence features are sent to the temporal encoder so that motion can be encoded. It happens after the sequence has been output from the spatial encoder.

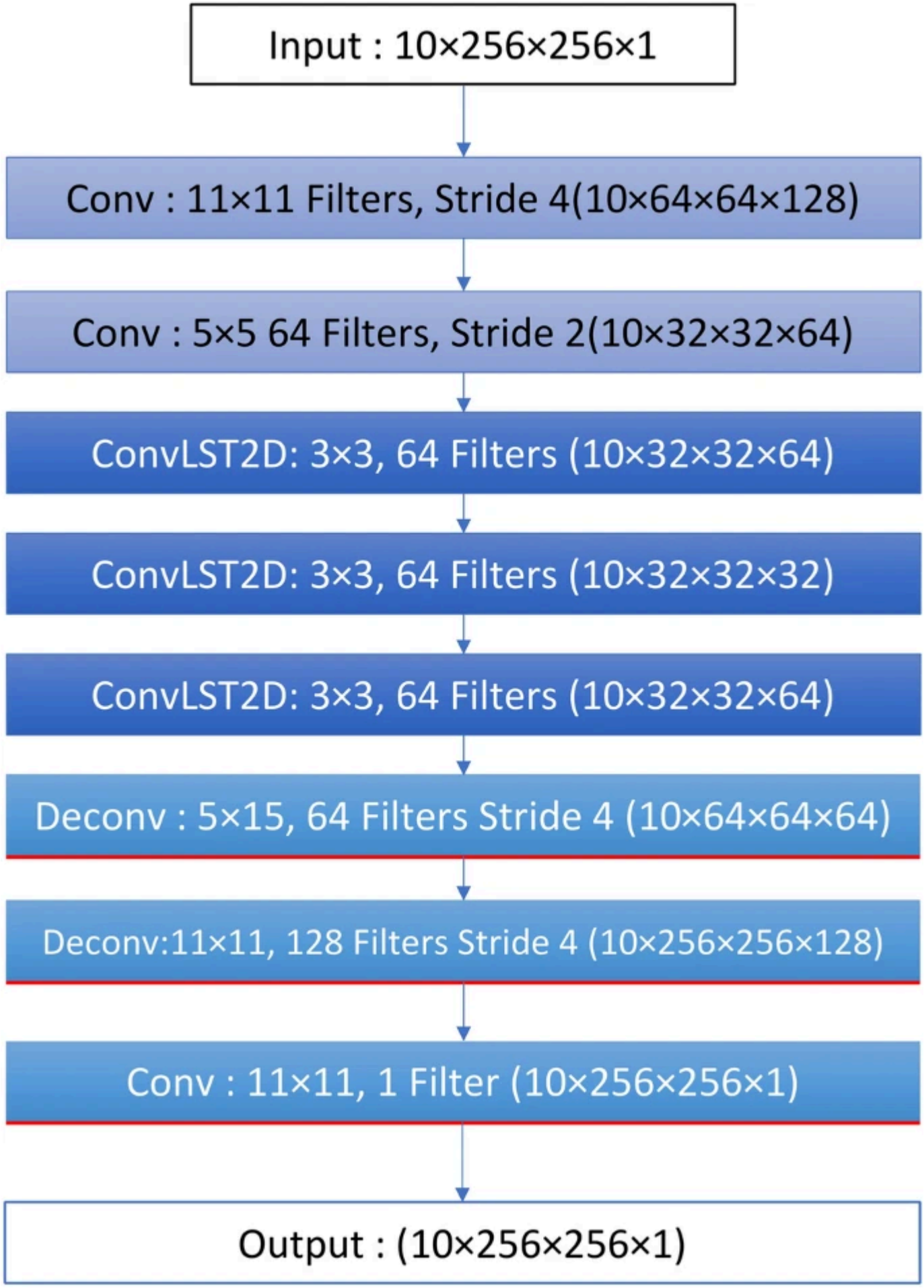
The video sequence can be reconstructed with the help of the decoder, which acts similarly to the encoder. Training Deep Neural Networks (DNN) requires a significant financial investment. The batch normalization procedure is used to bring the activities of the neurons up to a consistent level to cut down on the amount of time needed for training. The encoder is constructed using two MaxPooling layers and three convolutional layers. After each layer comes a batch normalization layer to adjust the values. The decoder consists of two sampling layers and three deconvolutions. In the decoder, too, we follow each layer with a batch normalization layer. The final output layer has 10 channels with the sigmoid activation. The model is compiled with previously mentioned hyperparameters and trained for 30 epochs. After training, we load the saved model and test it on the test dataset. This model can now detect motion correctly, which the earlier failed to achieve.

H. Spatio-Temporal Autoencoder Convolutional LSTM

Through the application of convolutional LSTM cells, this work can potentially advance the previously developed model. Convolutional LSTM layers were used in place of fully connected LSTM layers. The inability of FC-LSTMs to store spatial data exceptionally well is caused by total connections in input-to-state and state-to-state transitions. During these transitions, no spatial information is encoded.

The spatio-encoder has two convolutional layers for its computations. The temporal encoder and decoder consist of one convolutional LSTM layer each. This model bottleneck layer is a convolutional LSTM layer. The spatio-decoder consists of two deconvolutional layers, and lastly, there is the sigmoid-based output layer. Each layer is now normalized by layer normalization instead of batch normalization, as we use RNNs in Fig. 7.

Fig. 7



Architecture of spatio-temporal convolutional LSTM autoencoder

Results and Discussion

The L2 norm is used to calculate the reconstruction error of the intensity value I of a pixel at any point (x, y) in the frame 't' of the video. The distance between the vector coordinates and the vector space's origin is what the L2 norm attempts to compute. The Euclidean distance is another name for this concept, Eq. (5):

$$e(x,y,t) = \left\| {I(x,y,t) - f_{{\text{w}}} I(x,y,t)} \right\|_{2} ,$$

(5)

f_{w}, represents the trained model that has learned the training dataset using the LSTM convolutional autoencoder.

The following equation explains the reconstruction error.

$$e(t) = \sum\limits_{x,y} {e^{x,y,t} }$$

(6)

The errors in the reconstruction are determined by adding all these pixels while making mistakes. If the sequence starts at 't', then the following equation can be used to determine the cost of reconstructing a 10-frame sequence:

$${\text{Seq\_Reconstruction\_Cost}}(t) = \sum\limits_{t^{\prime} = t}^{t + 10} {e(t{\prime})} .$$

(7)

The anomaly or abnormality score (denoted by $Sa(t)$) is computed by scaling the cost between 0 and 1. It is computed according to the following equation:

$$s_{{\text{a}}} (t) = \frac{{{\text{Seq\_Reconstrcution\_Cost}}(t) - {\text{Seq\_Reconstruction\_Cost}}(t)_{{{\text{Min}}}} }} {{{\text{Seq\_Reconstruction\_Cost}}(t)_{{{\text{Max}}}} }}.$$

(8)

The regularity score (denoted by $S_r(t)$) is then calculated by subtracting the AD scores from 1, Eq. (9):

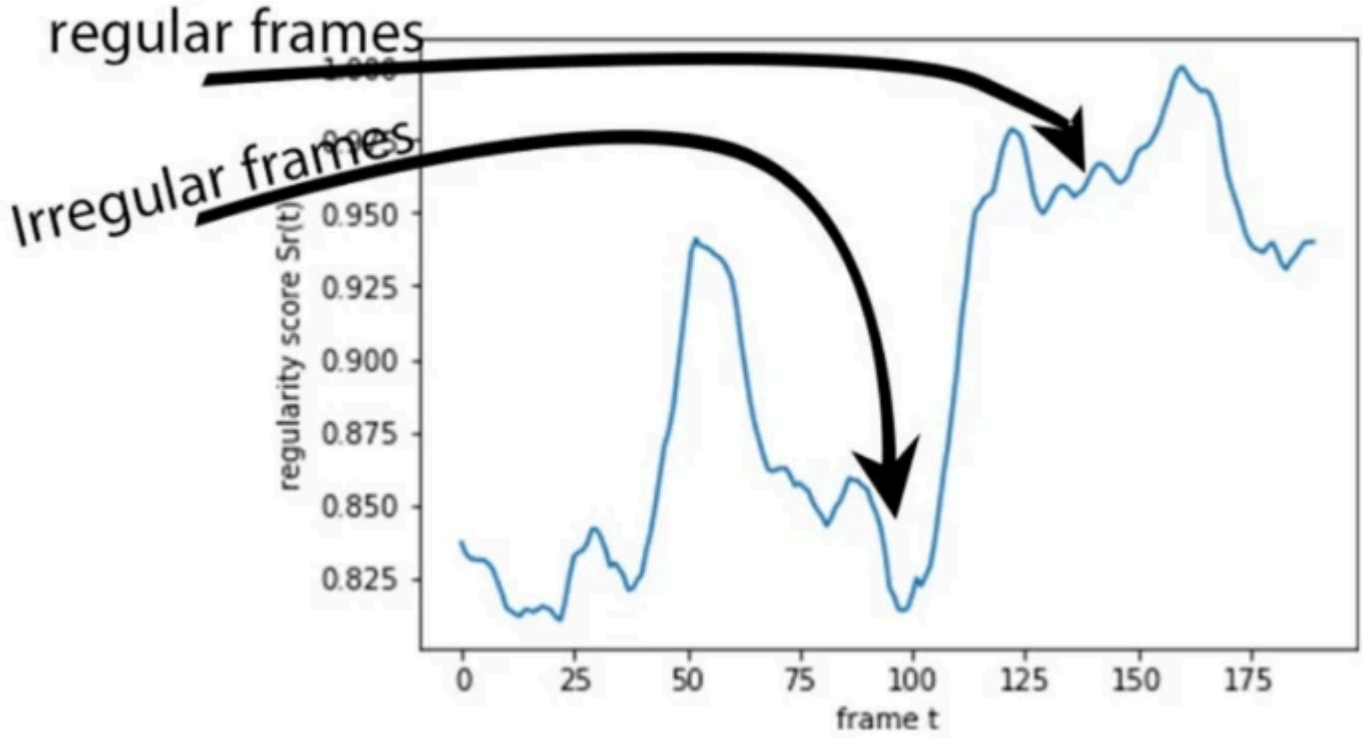
$$s_{{\text{r}}} (t) = 1 - s_{{\text{a}}} (t).$$

(9)

The regularity scores are plotted as shown.

In Fig. 8, the minima or the dips in the graph are the frames where anomalies are detected.

Fig. 8

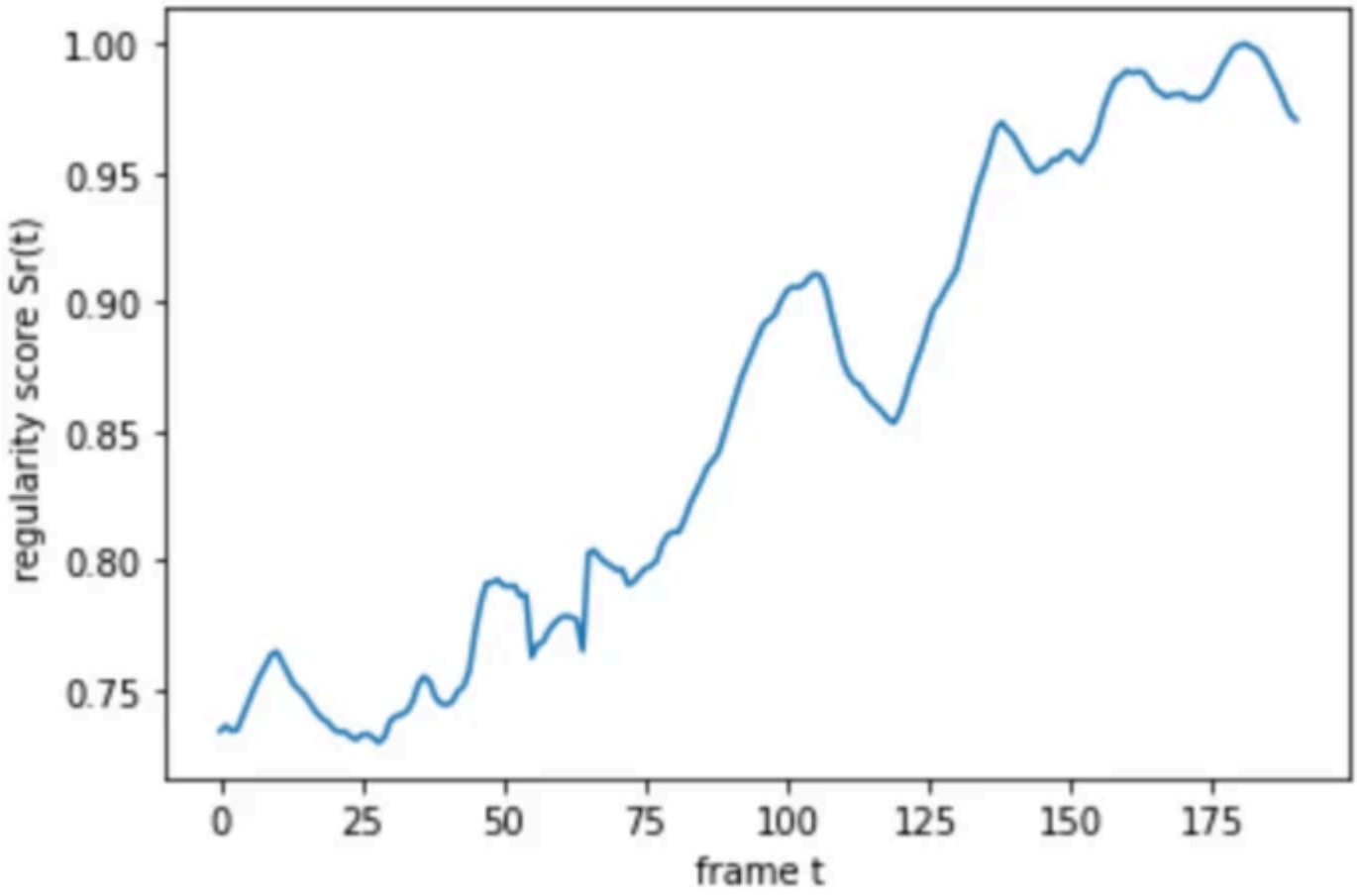


Plot for regularity score

The determination of the AD threshold in this research involved a manual approach guided by domain-specific considerations. This manual threshold selection allowed for

adaptation to the dataset's characteristics and the operational requirements of the application. Incorporating domain expertise, the definition of anomalies within this context was carefully outlined to ensure interpretability and alignment with real-world scenarios. However, it is important to note that manual threshold setting is subject to sensitivity and potential subjectivity. To address this, sensitivity analyses were conducted to evaluate the threshold's impact on model performance. This tailored approach aimed to balance FP and FN, ensuring optimal AD results within this unique application (Fig. 9).

Fig. 9



Regularity score for test dataset—Test 010

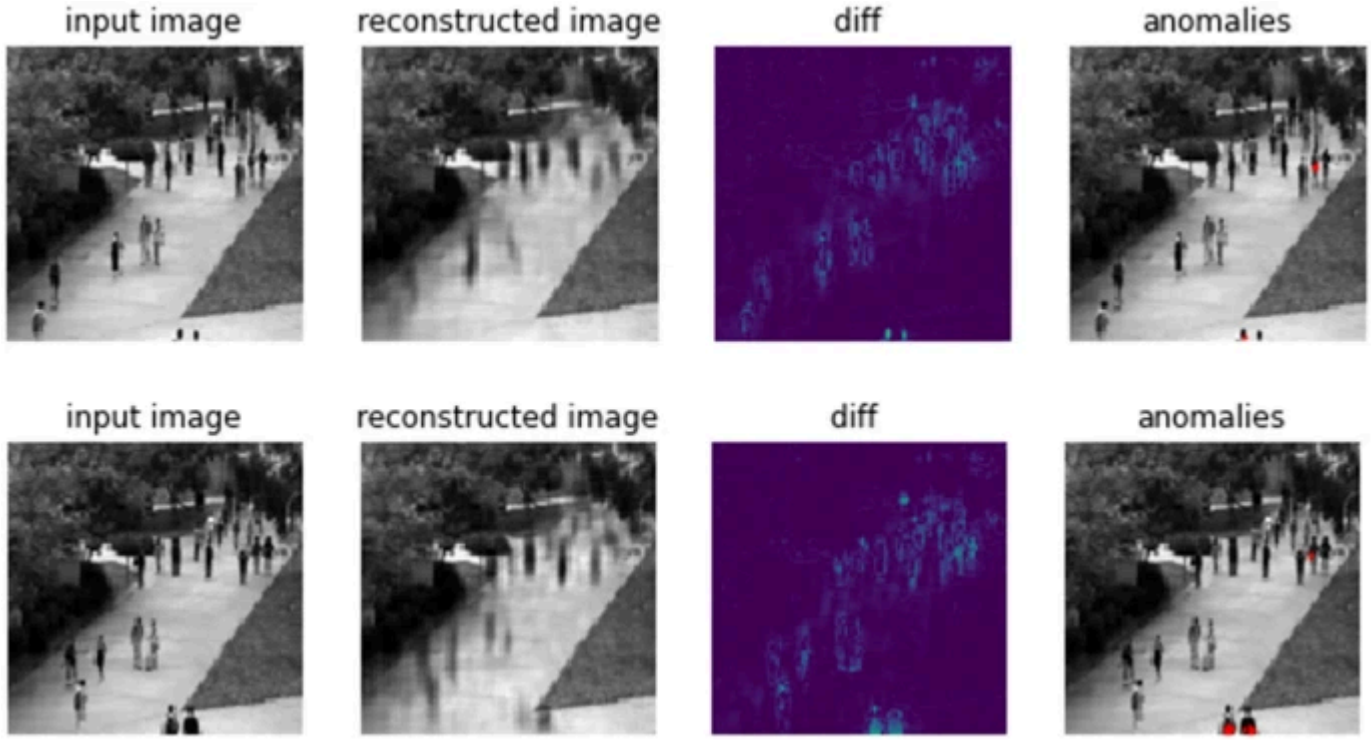
Future researchers are encouraged to assess whether manual or automated threshold selection is most suitable for their specific use cases.

Predictions

a. Convolutional Autoencoder (CAE)

The model barely recognizes the object from the video (Fig. 10).

Fig. 10

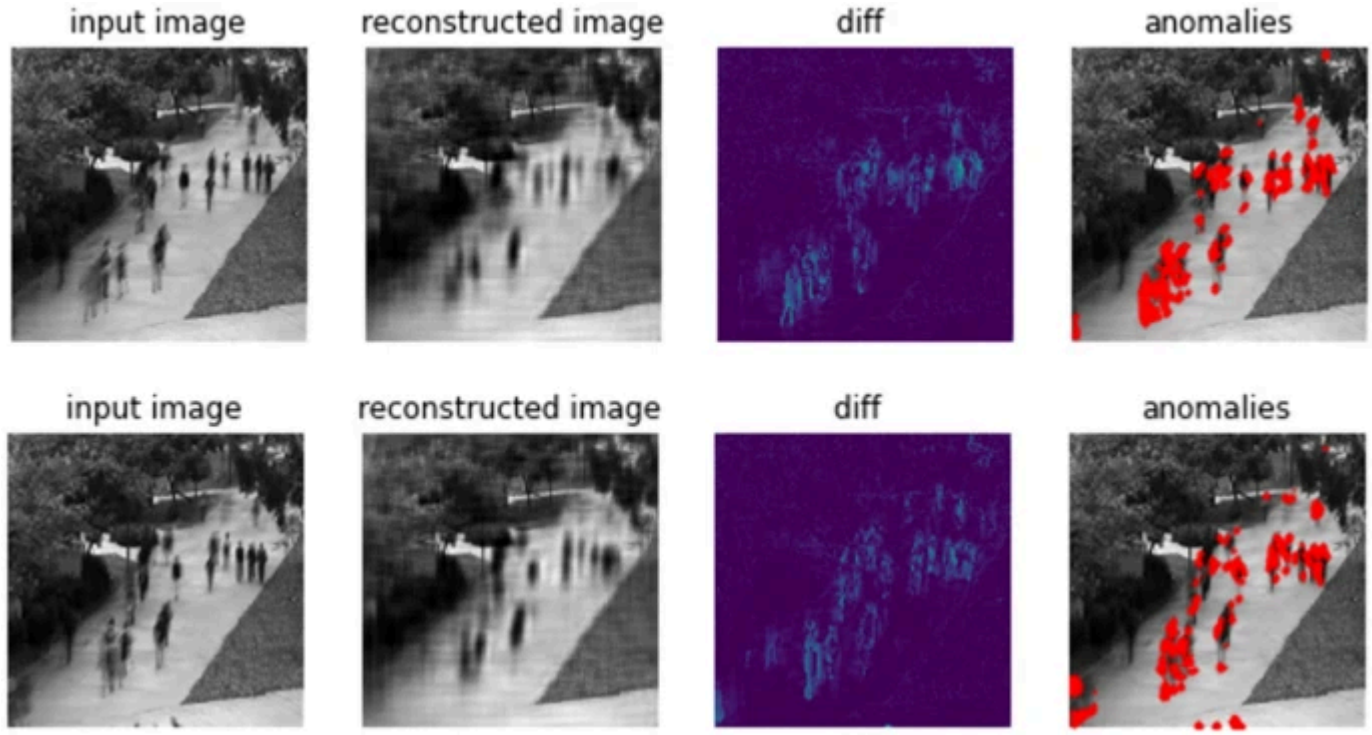


Predictions from the convolutional autoencoder model

b. Spatio-Temporal Stacked Frame Autoencoder-STAE

The model recognizes the temporal dependencies, and motion is detected accurately. As soon as pedestrians stop walking, they are no longer highlighted and vice-versa (Fig. 11).

Fig. 11

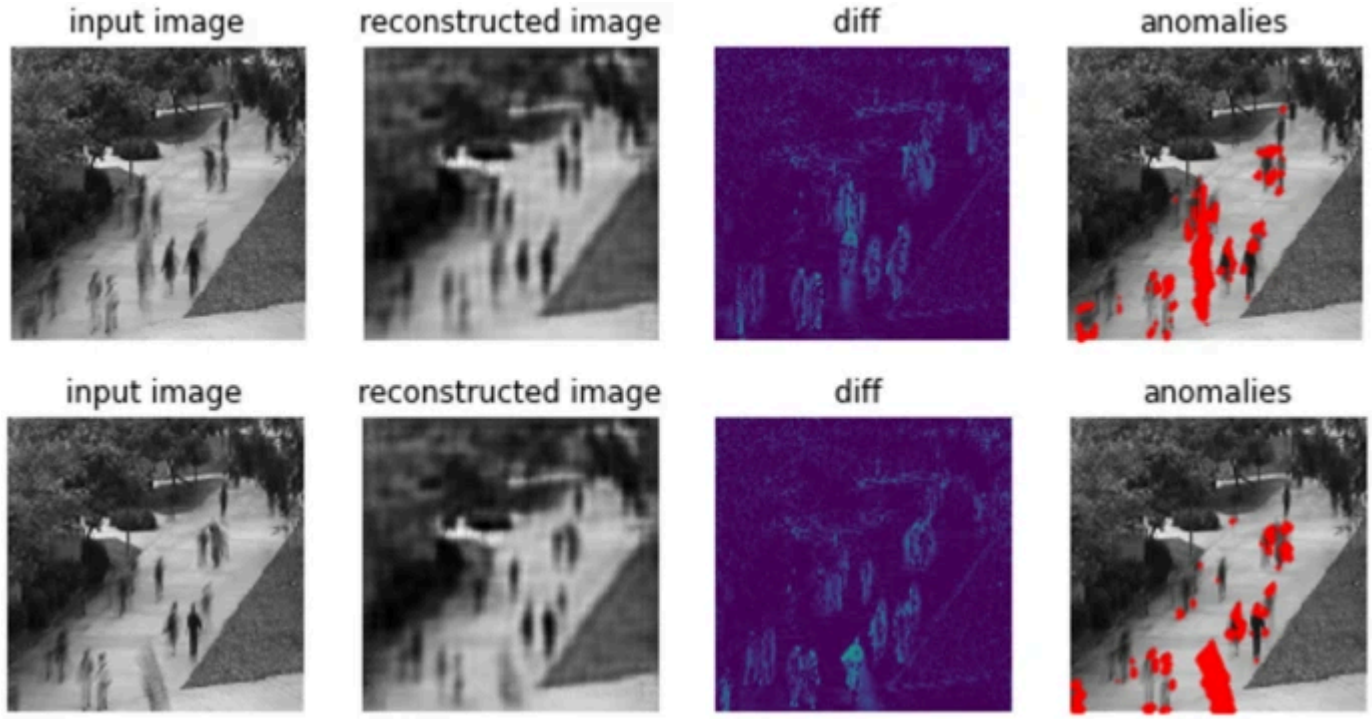


Predictions from the spatio-temporal autoencoder model

c. Spatio-Temporal Autoencoder Convolutional LSTM

The straight-line highlight in the image showcases how the model can follow the anomaly. It tracks the path of the cyclist (Fig. 12).

Fig. 12



Predictions from the LSTM autoencoder model

Understanding the limitations of the proposed video AD models is crucial for practical applications. These models may produce False Positives (FP) in scenarios involving sudden environmental changes, crowded scenes, camera disturbances, or partial occlusions, where normal variations or abrupt transitions are falsely identified as anomalies. Conversely, False Negatives (FN) may occur in gradual anomalies, adaptive behaviors, high variability scenes, or entirely novel anomalies that the models have not been trained to recognize. These limitations emphasize the need for further research to refine model architectures, incorporate contextual information, and establish adaptive thresholding mechanisms to reduce both FP and FN rates, ultimately enhancing the reliability of AD models for real-world surveillance and AD applications.

Conclusion and Future Work

Video Anomaly Detection (AD) presents an enduring challenge in computer vision and surveillance, driving continuous efforts to refine existing methodologies. As showcased in this study, Deep Learning (DL) has emerged as a potent tool for tackling this challenge. The research introduces an innovative model that combines spatial Feature Extraction (FE) with a temporal sequencer conv-LSTM, effectively redefining AD as a spatio-temporal sequence AD problem. The conv-LSTM layer, notable for its convolutional architecture, inherits the merits of Fully Connected LSTMs (FC-LSTMs) and

demonstrates suitability for analyzing spatio-temporal data. The model excels when applied to videos capturing routine events from a fixed viewpoint, although its performance may exhibit variability depending on scene complexity, occasionally yielding False-Positive (FP) outcomes.

Contemplating the future of AD research unveils several promising directions. Notably, a strategic plan is to integrate spatio-temporal autoencoder convolutional LSTM and employ layer normalization techniques to streamline the training process. This research underscores the importance of addressing ethical and privacy considerations associated with surveillance technologies in tandem with technical advancements. A strong commitment to individual privacy will be upheld by incorporating anonymization methods and privacy safeguards into future AD systems.

Moreover, this work contributes to the drive for standardization by leveraging established datasets for model evaluation, promoting equitable comparisons across diverse approaches. Future research will delve into multimodal integration, enhancing the model's capacity to process diverse data sources effectively. Incorporating human feedback mechanisms and prioritizing privacy-preserving measures will be pivotal in advancing video AD technology while safeguarding individual rights and privacy. The challenges and research directions outlined collectively steer the ongoing evolution of video AD.

Availability of Data and Material

Not applicable.

Code Availability

Not Applicable.

References

1. Ruwali A, Kumar AJS, Prakash KB, Sivavaraprasad G, Ratnam DV. Implementation of hybrid deep learning model (LSTM-CNN) for ionospheric TEC forecasting using GPS data. IEEE Geosci Remote Sens Lett. 2021;18(6):1004–8.

Article Google Scholar

2. Kantipudi MVVP, Kumar S, Jha AK. Scene text recognition based on bidirectional LSTM and deep neural network. Comput Intell Neurosci. 2021;2021:1–11.

Article Google Scholar

3. Ratnam DV, Rao KN. Bi-LSTM based deep learning method for 5G signal detection and channel estimation. AIMS Electron Electr Eng. 2021;5(4):334–41.

Article Google Scholar

4. Reddybattula KD, et al. Ionospheric TEC forecasting over an Indian low latitude location using long short-term memory (LSTM) deep learning network. Universe. 2022;8(11):562.

Article Google Scholar

5. Enireddy V, Karthikeyan C, Babu DV. OneHotEncoding and LSTM-based deep learning models for protein secondary structure prediction. Soft Comput. 2022;26(8):3825–36.

Article Google Scholar

6. Fernandes, Mannepalli K. Speech emotion recognition using deep learning LSTM for Tamil language. Pertan J Sci Technol. 2021;29(3):1915–36.

Google Scholar

7. Fernandes JB, Mannepalli K. Enhanced deep hierarchal GRU & BILSTM using data augmentation and spatial features for tamil emotional speech recognition. Int J Mod

Educ Comput Sci. 2022;14(3):45–63.

Article Google Scholar

8. Dharani NP, Bojja P. Analysis and prediction of COVID-19 by using recurrent LSTM neural network model in machine learning. Int J Adv Comput Sci Appl. 2022;13(5):171–8.

Google Scholar

9. Divya TV, Banik BG. Detecting fake news over job posts via bi-directional long short-term memory (BIDLSTM). Int J Web-Based Learn Teach Technol. 2021;16(6):1–18.

Article Google Scholar

10. Bhimavarapu U. IRF-LSTM: enhanced regularization function in LSTM to predict the rainfall. Neural Comput Appl. 2022;34(22):20165–77.

Article Google Scholar

11. Majji R, Prakash PGO, Cristin R, Parthasarathy G. Social bat Optimisation dependent deep stacked auto-encoder for skin cancer detection. IET Image Process. 2020;14(16):4122–31.

Article Google Scholar

12. Brahmane AV, Krishna CB. Rider chaotic biography optimization-driven deep stacked auto-encoder for big data classification using spark architecture: Rider chaotic biography optimization. Int J Web Serv Res. 2021;18(3):42–62.

Article Google Scholar

13. Panneerselvam IR. Transfer learning autoencoder used for compressing multimodal biosignal. Multimedia Tools Appl. 2022;81(13):17547–65.

Article Google Scholar

14. Mahanty M, Bhattacharyya D, Midhunchakkaravarthy D. SRGAN assisted encoder-decoder deep neural network for colorectal polyp semantic segmentation. Revue d'Intelligence Artificielle. 2021;35(5):395–401.

Article Google Scholar

15. Tilak VG, Ghali VS, Kumar AD, Sankar KBS, Sharanya VSNS. Deep autoencoder for automatic defect detection in thermal wave imaging. J Green Eng. 2020;10(12):13107–18.

Google Scholar

16. Kumar YP, Babu BV. Stabbing of intrusion with learning framework using auto encoder based intellectual enhanced linear support vector machine for feature dimensionality reduction. Revue d'Intelligence Artificielle. 2022;36(5):737–43.

Article Google Scholar

17. Brahmane AV, Krishna BC. DSAE-deep stack auto encoder and RCBO-rider chaotic biogeography optimization algorithm for big data classification. Adv Parallel Comput. 2021;39:213–27.

Google Scholar

18. Appathurai A, Sundarasekar R, Raja C, Alex EJ, Palagan CA, Nithya A. An efficient optimal neural network-based moving vehicle detection in traffic video surveillance system. Circuits Syst Signal Process. 2020;39(2):734–56.

Article Google Scholar

19. Raju K, et al. A robust and accurate video watermarking system based on SVD hybridation for performance assessment. Int J Eng Trends Technol. 2020;68(7):19–

24.

Article Google Scholar

20. Shaik AA, Mareedu VDP, Polurie VVK. Learning multiview deep features from skeletal sign language videos for recognition. Turk J Electr Eng Comput Sci. 2021;29(2):1061–76.

Article Google Scholar

21. Suneetha M, et al. Multi-view motion modelled deep attention networks (M2DA-Net) for video-based sign language recognition. J Vis Commun Image Represent. 2021;78:103161.

Article Google Scholar

22. Ghuge CA, Chandra Prakash V, Ruikar SD. Weighed query-specific distance and hybrid NARX neural network for video object retrieval. Comput J. 2020;63(7):1738–55.

Article Google Scholar

23. Mohan KK, Prasad CR, Kishore PVV. Yolo V2 with bifold skip: a deep learning model for video based real time train bogie part identification and defect detection. J Eng Sci Technol. 2021;16(3):2166–90.

Google Scholar

24. Kotkar VA, Sucharita V. Scalable anomaly detection framework in video surveillance using keyframe extraction and machine learning algorithms. J Adv Res Dyn Control Syst. 2020;12(7):395–408.

Article Google Scholar

25. Suneetha M, Prasad MVD, Kishore PVV. Sharable and unshareable within class multi view deep metric latent feature learning for video-based sign language recognition. Multimedia Tools Appl. 2022;81(19):27247–73.

Article Google Scholar

26. Ali SKA, Prasad MVD, Kumar PP, Kishore PVV. Deep multi view spatio temporal spectral feature embedding on skeletal sign language videos for recognition. Int J Adv Comput Sci Appl. 2022;13(4):810–9.

Google Scholar

27. Gullapelly A, Banik BG. Exploring the techniques for object detection, classification, and tracking in video surveillance for crowd analysis. Indian J Comput Sci Eng. 2020;11(4):321–6.

Article Google Scholar

28. Ghuge CA, Prakash VC, Ruikar SD. Systematic analysis and review of video object retrieval techniques. Control Cybern. 2020;49(4):471–98.

Google Scholar

29. Priyadharshini B, Gomathi T. Navie bayes classifier for wireless capsule endoscopy video to detect bleeding frames. Int J Sci Technol Res. 2020;9(1):3286–91.

Google Scholar

30. Ali SA, Prasad MVD, Kishore PVV. Ranked multi-view skeletal video-BASED sign language recognition with triplet loss embeddings. J Eng Sci Technol. 2022;17(6):4367–97.

Google Scholar

31. Krishnamohan K, Prasad CR, Kishore PVV. Train rolling stock video segmentation and classification for bogie part inspection automation: a deep learning approach. J Eng Appl Sci. 2022. https://doi.org/10.1186/s44147-022-00128-x.

Article Google Scholar

32. Li X, Manivannan P, Anand M. Task modelling of sports event for personalized video streaming data in augmentative and alternative communication. J Interconnect Netw. 2022. https://doi.org/10.1142/S0219265921410279.

Article Google Scholar

33. Wagdarikar AMU, Senapati RK. A secure communication approach in OFDM using optimized interesting region-based video watermarking. Int J Pervasive Comput Commun. 2022;18(2):171–94.

Article Google Scholar

34. Ghuge C, Prakash VC, Ruikar S. An integrated approach using optimized naive bayes classifier and optical flow orientation for video object retrieval. Int J Intell Eng Syst. 2021;14(3):210–21.

Google Scholar

35. Jadhav AD, Pellakuri V. Highly accurate and efficient two phase-intrusion detection system (TP-IDS) using distributed processing of HADOOP and machine learning techniques. J Big Data. 2021. https://doi.org/10.1186/s40537-021-00521-y.

Article Google Scholar

36. Adam A, Rivlin E, Shimshoni I, Reinitz D. Robust real-time unusual event detection using multiple fixed-location monitors. IEEE Trans Pattern Anal Mach Intell. 2008;30(3):555–60.

Article Google Scholar

37. Hasan M, Choi J, Neumann J, Roy-Chowdhury AK, Davis LS. Learning temporal regularity in video sequences. In: 2016 IEEE conference on computer vision and pattern recognition (CVPR). 2016. pp. 733–742.

38. Lu C, Shi J, Jia J. Abnormal event detection at 150 fps in Matlab. In: 2013 IEEE international conference on computer vision. 2013. pp. 2720–2727.

39. Mahadevan V, Li W, Bhalodia V, Vasconcelos N. Anomaly detection in crowded scenes. In: Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR). 2010. pp. 1975–1981.

40. Mehran R, Oyama A, Shah M. Abnormal crowd behavior detection using social force model. In: 2009 IEEE computer society conference on computer vision and pattern recognition workshops, CVPR workshops 2009. 2009. pp. 935–942.

41. Patraucean V, Handa A, Cipolla R. Spatio-temporal video autoencoder with differentiable memory. In: International conference on learning representations, 2015. 2016. pp. 1–10.

42. Sabokrou M, Fathy M, Hoseini M, Klette R. Real-time anomaly detection and localization in crowded scenes. In: 2015 IEEE conference on computer vision and pattern recognition workshops (CVPRW). 2015. pp. 56–62.

43. Shi X, Chen Z, Wang H, Yeung DY, Wong W, Woo W. Convolutional LSTM network: a machine learning approach for precipitation nowcasting. In: Proceedings of the 28th international conference on neural information processing systems, NIPS 2015. Cambridge, MA, USA: MIT Press; 2015. pp. 802–810.

44. Wang T, Snoussi H. Histograms of optical flow orientation for abnormal events detection. In: IEEE international workshop on performance evaluation of tracking and surveillance, PETS. 2013. pp. 45–52.

45. Yen SH, Wang CH. Abnormal event detection using HOSF. In: 2013 International conference on IT convergence and security, ICITCS 2013. 2013.

46. Zhao B, Fei-Fei L, Xing EP. Online detection of unusual events in videos via dynamic sparse coding. In: Proceedings of the IEEE computer society conference on computer vision and pattern recognition. 2011. pp. 3313–3320

47. Zhou S, Shen W, Zeng D, Fang M, Wei Y, Zhang Z. Spatial-temporal convolutional neural networks for anomaly detection and localization in crowded scenes. Sig Process Image Commun. 2016;47:358–68.

Article Google Scholar

48. Shakeela S, Shankar NS, Reddy PM, Tulasi TK, Koneru MM. Optimal ensemble learning based on distinctive feature selection by univariate ANOVA-F statistics for IDS. Int J Electron Telecommun. 2021;67(2):267–75.

Google Scholar

49. Jadhav AD, Pellakuri V. Accuracy based fault tolerant two phase—intrusion detection system (TP-IDS) using machine learning and HDFS. Revue d'Intelligence Artificielle. 2021;35(5):359–66.

Article Google Scholar

50. Hira S, Bai A, Hira S. An automatic approach based on CNN architecture to detect Covid-19 disease from chest X-ray images. Appl Intell. 2021;51(5):2864–89.

Article Google Scholar

51. Murthy MYB, Koteswararao A, Babu MS. Adaptive fuzzy deformable fusion and optimized CNN with ensemble classification for automated brain tumor diagnosis. Biomed Eng Lett. 2022;12(1):37–58.

Article Google Scholar

52. Kumar S, Jain A, Rani S, Alshazly H, Idris SA, Bourouis S. Deep neural network based vehicle detection and classification of aerial images. Intelligent Autom Soft Comput. 2022;34(1):119–31.

Article Google Scholar

53. Lakshmi Mallika I, Venkata Ratnam D, Raman S, Sivavaraprasad G. A new ionospheric model for single frequency GNSS user applications using Klobuchar model driven by auto regressive moving average (SAKARMA) method over Indian region. IEEE Access. 2020;8:54535–53.

Article Google Scholar

54. Thirugnanasambandam K, Rajeswari M, Bhattacharyya D, Kim J-Y. Directed artificial bee colony algorithm with revamped search strategy to solve global numerical optimization problems. Autom Softw Eng. 2022. https://doi.org/10.1007/s10515-021-00306-w.

Article Google Scholar

55. Sasank VVS, Venkateswarlu S. An automatic tumour growth prediction-based segmentation using full resolution convolutional network for brain tumour. Biomed Signal Process Control. 2022;71:103090.

Article Google Scholar

56. Budati AK, Katta RB. An automated brain tumor detection and classification from MRI images using machine learning techniques with IoT. Environ Dev Sustain. 2022;24(9):10570–84.

Article Google Scholar

57. Gopi Tilak V, Ghali VS, Vijaya Lakshmi A, Suresh B, Naik RB. Proximity based automatic defect detection in quadratic frequency modulated thermal wave imaging. Infrared Phys Technol. 2021;114:103674.

Article Google Scholar

58. Bhimanpallewar RN, Narasingarao MR. AgriRobot: Implementation and evaluation of an automatic robot for seeding and fertiliser microdosing in precision agriculture. Int J Agric Resour Gov Ecol. 2020;16(1):33–50.

Google Scholar

59. Thamizhazhagan P, et al. AI based traffic flow prediction model for connected and autonomous electric vehicles. Comput Mater Contin. 2022;70(2):3333–47.

Google Scholar

60. Vesala GT, Ghali VS, Lakshmi AV, Naik RB. Deep and handcrafted feature fusion for automatic defect detection in quadratic frequency modulated thermal wave imaging. Russ J Nondestr Test. 2021;57(6):476–85.

Article Google Scholar

61. Vijayalakshmi A, Ghali VS, Chandrasekhar Yadav GVP, Gopitilak V, Muzammil Parvez M. Machine learning based automatic defect detection in non-stationary thermal wave imaging. ARPN J Eng Appl Sci. 2020;15(2):172–8.

Google Scholar

Funding

Not applicable.

Author information

Authors and Affiliations

Department of Networks and Cybersecurity, Faculty of Information Technology, Al Ahliyya Amman University Country, Amman, Jordan
Ghayth Almahadin

School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, 600127, Tamil Nadu, India
Maheswari Subburaj

Department of Networks and Cybersecurity, Information Technology, Al Ahliyya Amman University, Amman, Jordan
Mohammad Hiari

Department Computing Technology, School of Computing, SRM Institute of Science and Technology, Kattankulathur Campus, Chennai, 603203, Tamil Nadu, India
Saranya Sathasivam Singaram

Department of Computer Science and Engineering, Koneru Lakshmaiah Education Foundation, Vaddeswaram, Guntur, 522302, India
Bhanu Prakash Kolla

Department of Computer Science and Engineering, Swami Keshvanand Institute of Technology, Management and Gramothan (SKIT), Jaipur, 302017, Rajasthan, India
Pankaj Dadheech

Symbiosis Institute of Computer Studies and Research (SICSR), Symbiosis International (Deemed University), Pune, 411016, MH, India
Amol D. Vibhute

Department of Computer Science and Engineering, PSN College of Engineering and Technology, Tirunelveli, 627152, Tamil Nadu, India
Sudhakar Sengan

Corresponding authors

Correspondence to Maheswari Subburaj or Sudhakar Sengan.

Ethics declarations

Conflict of interest

Not applicable.

Additional information

Publisher's Note

This article is part of the topical collection "Soft Computing in Engineering Applications" guest edited by Kanubhai K. Patel.

Rights and permissions

About this article

Cite this article

Almahadin, G., Subburaj, M., Hiari, M. *et al.* Enhancing Video Anomaly Detection Using Spatio-Temporal Autoencoders and Convolutional LSTM Networks. *SN COMPUT. SCI.* **5**, 190 (2024). https://doi.org/10.1007/s42979-023-02542-1

Received
10 September 2023

Accepted
11 November 2023

Published
11 January 2024

DOI
https://doi.org/10.1007/s42979-023-02542-1

Share this article

Anyone you share the following link with will be able to read this content:

Get shareable link

Provided by the Springer Nature SharedIt content-sharing initiative

Keywords

Machine learning | Anomaly detection | LSTM | Autoencoders | Spatio-temporal features | Suspicious activities