

Surveillance detection of anomalous activities with optimized deep learning technique in crowded scenes

Omobayo Ayokunle Esan, Dorcas Oladayo Esan, Munienge Mbodila, Femi Abiodun Elegbeleye, Kesewaa Koranteng

Department of Information Technology Systems, Faculty of Economics and Information Technology Systems, Walter Sisulu University, Queenstown, South Africa

Article Info

Article history:

Received Jul 26, 2022

Revised Sep 30, 2022

Accepted Nov 2, 2022

Keywords:

Anomalies

Convolutional neural network

Deep neural network

Long-short term memory

Optimization

ABSTRACT

The performance of conventional surveillance systems is challenged by high error detection rates in busy scenes, which has significantly affected the accurate detection of the current surveillance system. Feature representation and object pattern extraction from different scenes have made deep learning (DL) promising methods in surveillance systems, compared to the approaches where features are created manually. To improve the detection accuracy, this paper presents an intelligent DL technique that combines convolutional neural network (CNN) and long short-term memory (LSTM). CNN extracts and learns the object features from a set of raw images, while the LSTM is then used by gated mechanisms to store important information from the extracted features. The proposed method was validated using datasets from the University of California San Diego (UCSD). The result shows that the model achieves 95% accuracy, which is superior compared to other conventional detection models.

This is an open access article under the [CC BY-SA](#) license.



Corresponding Author:

Omobayo Ayokunle Esan

Department of Information Technology Systems

Faculty of Economics and Information Technology Systems, Walter Sisulu University

Queenstown, South Africa

Email: oesan@wsu.ac.za

1. INTRODUCTION

Video surveillance technologies are used in both private and public sectors for monitoring and security purposes [1], [2]. This technology works with full human involvement, which requires manual assessment to detect anomalous activities, which is time-consuming and costly [1]–[4]. Detecting anomalous behaviors in video surveillance remains a challenging task due to insufficient feature extraction methods in current surveillance systems. Although several feature extraction and behavior pattern detection methods have been reported in the literature [5], [6]. None of these methods have provided satisfactory performance that can be used for detecting anomalies in crowded scenes [7].

Events that are inconsistent with normal events are referred to as anomalies [5]. An event that is anomalous in one scene (e.g., a person running) may not be anomalous in a second scene because the normal events in the second scene may include people running, while this is not the case in the first scene [8]. Therefore, the anomalies are not sufficiently large and similar to be modeled effectively. There is an urgent need for systems that can intelligently identify anomalous events within short intervals. Traditional solutions in the literature use local patterns that extract low-level features, such as histogram of oriented gradients (HOG), histogram of oriented flows (HOF), and optical flow [9]–[11] which are based on the appearance and motion information of the objects [8]. In most cases, these techniques do not provide accurate results,

exposing the security system to a high false detection rate (false alarm). Remarkable success has been achieved with the use of deep learning (DL) techniques in computer vision for object recognition and classification [12], [13], which require training of labeled datasets (supervised) and the use of convolutional neural network (CNN) as an unsupervised technique to learn the extracted features [14]. The use of DL techniques compared to manual detection methods has shown the importance of extracting rich features for accurate detection of abnormal activities.

The combination of CNN and long short-term memory (LSTM) is presented in this paper to fill the gap in the accuracy of the current monitoring system. We used CNN input images to reconstruct and train important features from the image. The features are fed to the LSTM for storage and recognition. During recognition, the reconstructed images are compared with the incoming images to identify patterns that deviate from the norm and consider these patterns as anomalies [15]. The comparison of the proposed anomaly detection system with other basic detection methods was performed using University of California San Diego (UCSD) pedestrian datasets. The experimental results show that our method CNN-LSTM is superior to other selected methods used in the implementation. The paper contributes in following aspects: i) input image feature extraction and reconstruction using combining the CNN with LSTM method, ii) improving the anomalous detection systems' accuracy and reducing false positive error rates, and iii) detailed experimental evaluations of the new model and benchmarked with baseline detection models, using publicly available data to detect possible crime patterns in the future. The following sections of this paper are arranged in the following order: section 2 presents the proposed method, section 3 presents the method used, section 4 discusses the results, and section 5 concludes the paper.

2. RELATED WORKS

Much work has been done for anomalous activities from video streams [16] to address the problem of identifying anomalous events in a busy scene within short intervals using computer vision based on CNN. The proposed model was used for pre-training feature extraction from the anomalous frames. The results of the proposed model in the experiments with the Avenue and UCSD ped 2 datasets give an area under curve (AUC) of 71.97% and 89.52%, respectively. In comparison with other models, the authors found that the model performs better than other detection models.

An effective system for detecting anomalies in animated scenes was presented in [17] using variational autoencoders. The authors extracted appearance and motion patterns of objects in different scenes as features then compared features with manual feature extraction, which is mostly used for anomaly detection. In addition, Gaussian models were applied to the UCSD pedestrian dataset to predict the anomaly scores of the corresponding receptive field, and the performance of the model was found to be competitive with the existing models.

According to Maqsood *et al.* [14] presents surveillance anomaly detection to solve the problem of sparse occurrence of an anomalous event in a busy scene by using deep 3-dimensional convolutional networks (3D convNets) to learn the spatio-temporal feature on the University of Central Florida (UCF) crime video dataset. The results of the proposed model show that multiclass learning can improve the generalized competencies of the 3D convNet by effectively learning frame-level information and spatial additions to a fine-tuned pre-trained model. The approach significantly outperforms other state-of-the-art approaches in anomalous activity detection accuracy with an AUC of 82%.

A DL model based on cascaded autoencoders as one-class learning for anomaly detection and localization in surveillance videos was presented by [13]. The convolutional autoencoder and a sequence-to-sequence LSTM autoencoder were respectively used for spatiotemporal learning of the video images. One-class classification was used for training the model on the normal data and tested on the anomalous test data. The results obtained showed that the model performed quite remarkably compared to the other models in terms of the same error rate and the time required for anomaly detection and localization. According to Mehmood [18] worked on efficient anomaly detection in crowd videos to obtain low-cost anomaly detection by using pre-trained 2D CNN for motion information and a lighter form of 2D CNN to achieve high detection accuracy. The experiment on the publicly available crime dataset shows that the proposed model outperforms the existing approach in terms of recognition accuracy and provides better performance in generating input images. Our approach is similar to [5], but we use a CNN with LSTM to improve the detection of anomalous activities in video surveillance while reducing the false alarm rate.

2.1. Theoretical background

We utilized DL technique that harmonized CNN and LSTM technique, which is discussed in sections 2.1-2.2. The CNN stages are divided into sections 2.1.1 -2.1.5, this is the stage where the features are extracted and learned. The LSTM is sub-divided into sections 2.2.1-2.2.3, respectively.

2.1.1. Proposed convolutional neural network

Convolutional neural network (CNN) is an artificial neural network approach which is used for anomaly detection in time-series data. CNN is one of the most common neural network techniques used in image recognition, classification, object detection, and face recognition [19]. The layers comprise the convolutional layer (Kernel), the pooling layer, and an activation layer with the rectified linear unit (ReLU) as shown in Figure 1.

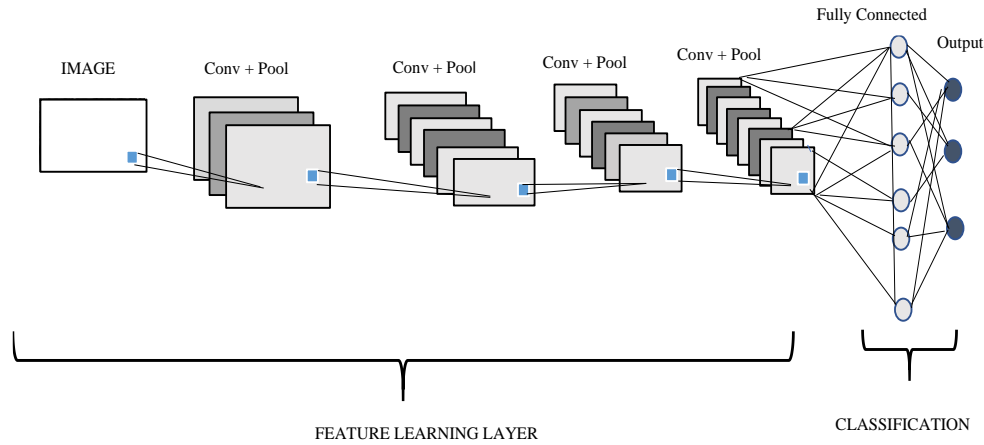


Figure 1. CNN network adopted from [20]

2.1.2. Convolutional layer

The first layer of the CNN that traverses the image to extract various features such as edges, lines, and corners is the convolutional layer [21], as shown in Figure 1. The feature maps in the image are determined using the Kernel function in the convolution layer, which determines the tensor of the feature maps. These Kernels use "stride(s)" to convolve the entire input image so that the output becomes integers (vectors). Immediately following the convolution layer is the dimensionality reduction of the image, which is used for the stride process as in (1):

$$F(i) = (I * K)(i, j) = \sum \sum ((i + m, j + n)K(m, n)) \quad (1)$$

where I is the input matrix, K refers to a 2D filter of size $m \times n$, and F is the output of the 2D feature map, the operation of the convolution layer is denoted by $I \times K$.

2.1.3. Rectified linear unit activation

ReLU is an example of a non-saturated activation function. An increasing number of nonlinear properties and system network connections make use of the ReLU layer. Negative values are removed from an activation map on the ReLU layer by setting all negative values as in (2) and (3):

$$ReLU(x) = \max(0, x) \quad (2)$$

$$\frac{d}{dx} ReLU(x) = \begin{cases} 1 & \text{if } x = 0 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

2.1.4. Pooling layer

In the pooling layer, the dimension of the input image is reduced, and assumptions are made about the features of the input image. It also prevents overfitting and makes the features more resistant to noise and distortion by achieving a deeper representation of successive layers [21], [22]. This layer helps in cross-channel subsampling of features as in (4):

$$X_j^i = X_i^{l-1 \times p_i^l}, i \in [1] \quad (4)$$

where X_j^i is the input image, j^{th} channel of a layer l ; P_i^l the i^{th} channel of the pooling operator in layer l ; and I is the channel amount of both layers $l - 1$ and l .

2.1.5. Fully connected layer

At the fully connected layer (FCL) stage of CNN are used as the final layers of a CNN where the input from other layers are converted into the vector [20]. Neurons in a FCL have connections to all activations in the previous layer (these layers mathematically sum a weighting of the previous layer of features to determine a specific target output result). It transforms the output into any desired number of classes into the network.

2.2. Long short-term memory

A LSTM network extends the ability of RNNs to remember things even further than what a typical RNN can do [19]. In a LSTM, there are four gates that determine whether the new input should be allowed, deleted, or whether it should affect the output [19]. As shown in Figure 2 and discussed in more detail in the following sections.

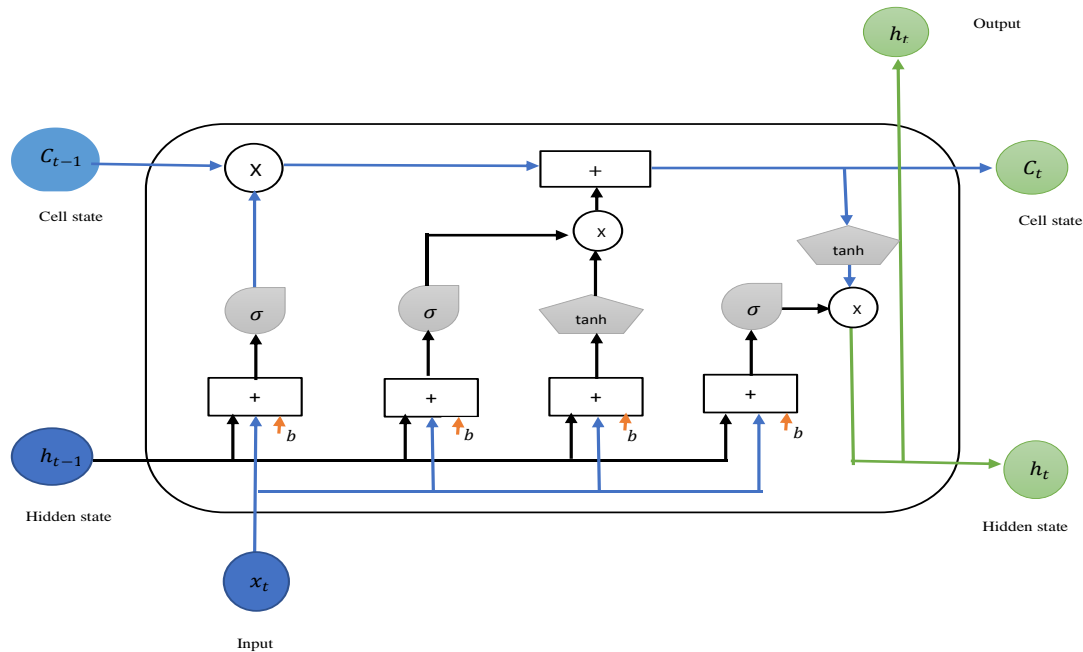


Figure 2. LSTM adopted from [20]

2.2.1. Forget gate

A forget gate identifies information that is not needed and therefore not included in the cell state by using a sigmoid function to identify and remove data that comes from both the output of the last LSTM unit (h_{t-1}) and the current input at a time (t). In addition, the sigmoid function determines which parts of the old output should be eliminated. f_t represents in this case the number in each cell (C_{t-1}) with vector values between 0 and 1, as in (5):

$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f) \quad (5)$$

where sigmoid function is represented as σ , and weight and bias are W_f and b_f , respectively, x_t is the image input sequence at a time step (t) and (h_{t-1}) is the cell output activation function at a time step (t-1).

2.2.2. Input gate

At this gate, the information about the new input x_t is updated and stored. The steps that are involved in the input gates include the sigmoid layer which is the layer that determines whether the new information in the cell state should be eliminated or updated, and the tanh layer is the layer that provides the weight to the values of the new information. The new cell state is updated by multiplying sigmoid values and tanh as (6) and (7) respectively.

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \quad (6)$$

$$\tilde{C}_t = \tanh (W_c \cdot [h_{t-1}, x_t] + b_c) \quad (7)$$

The updated new cell state is added to the old memory (C_{t-1}) to obtain the output (C_t) as (8):

$$C_t = f_t \times C_{t-1} + i_t \times C_t \quad (8)$$

where C_{t-1} and C_t are the cell states at the time $(t - 1)$ and (t) while W and b are the weight matrices and bias, respectively, of the cell state.

2.2.3. Output gate

In the final step, the output value (h_t) is based on the output cell state (o_t). First, a sigmoid layer decides which parts of the cell state make it to the output. Next, the output of the sigmoid gate is multiplied by the new values created by the tanh layer from the cell state (C_t), with a value ranging between -1 and 1 , as in (9) and (10) respectively.

$$o_t = \sigma(W_o[h_{t-1}, x_t]) + b_o \quad (9)$$

$$h_t = o_t \times \tanh (C_t) \quad (10)$$

where W_o and b_o are the weight matrices and bias, respectively, of the output gate.

3. METHOD

Here, the development of the proposed anomaly detection model is explained in detail for the better understanding of the reader. The method used to develop our CNN-LSTM model is divided into several components. This matter has been explained in detail in the following sections.

3.1. Image acquisition component

For this study, we used a dataset from the UCSD [23], which is publicly available. The dataset contains videos from various static cameras located at different points overlooking a busy pedestrian walkway. The anomalous activities consist of someone driving bicycle at pedestrian walkway as well as someone on wheelchair moving in pedestrian walkway. For the normal activities, the images consist of only normal pedestrian people walking on pedestrian walkway.

3.2. Feature engineering component

The data are passed to feature engineering, where features are extracted from the image to allow accurate detection of anomalies. The feature engineering used in this study is divided into two stages: image discretization and parameter selection. These are explained in more detail in the following sections.

3.2.1. Image discretization

Here the output of the normalized image frames is passed into the discretization stage. This involves dividing the range of normalized image values into k equal-sized bins by finding the image pixel maximum value denoted as ($I_{maximum}$) and the image pixel minimum value denoted as ($I_{minimum}$). The equal-width interval of the image pixel is computed by dividing the range of the observed value as in (11):

$$I_{equal_width} = \frac{(I_{maximum} - I_{minimum})}{k} \quad (11)$$

where k is a parameter supplied during the experiment.

3.2.2. Parameter turning for CNN-LSTM

Since increasing in numbers of parameter iterations increase the model performance. Therefore, tuned parameters such as epoch, batch size, Kernel, and learning rate at optimal. This is to obtain optimal performance which is one of the objectives of this research.

3.3. Detection component

Here, the image is passed to the CNN model and the features are obtained from the frames as explained in the previous sections. The feature is extracted from the image by the convolution block, which is coupled with a 2D CNN and a pooling layer; the ReLU serves as the activation function. As part of the convolution operation, the convolution Kernel multiplies the overlay matrix to extract the convolution feature. After two-dimensional convolution, the feature maps are extracted using the maximum pooling filter. The extracted feature maps are

forwarded to the LSTM layer to store relevant features for recall. The SoftMax feature is used in the fully linked layer for final recognition. The model architecture for the implementation is shown in Table 1.

Table 1. Summary of the CNN-LSTM network architecture

Layers	Ty	K_S	St	Ker	I_S	Layers	Ty	K_S	St	Ker	I_S
1	Convolutional	3×3	1	64	224×224×3	11	Convolutional	3×3	1	512	28×28×512
2	Convolutional	3×3	1	64	224×224×64	12	Convolutional	3×3	1	512	28×28×512
3	Pool	2×2	2	-	224×224×64	13	Pool	2×2	2	-	28×28×512
4	Convolutional	3×3	1	128	112×112×64	14	Convolutional	3×3	1	512	14×14×512
5	Convolutional	3×3	1	128	112×112×128	15	Convolutional	3×3	1	512	14×14×512
6	Pool	2×2	2	-	112×112×128	16	Convolutional	3×3	1	512	14×14×512
7	Convolutional	3×3	1	256	56×56×128	17	Pool	2×2	2	-	14×14×512
8	Convolutional	3×3	1	256	56×56×256	18	LSTM	-	-	-	49×512
9	Pool	2×2	2	-	56×56×256	19	FC	-	-	64	25,088
10	Convolutional	3×3	1	512	28×28×256	20	Output	-	-	3	64

where Ty stands for type, K_S for kernel size, St for stride, Ker for Kernel, and I_S for input size. The architecture shown in Table 1 consists of layers 1-17 of the network, which are the convolutional layers and the pooling layers, layer 18 represents the LSTM layer, and layer 19 is the fully connected output prediction layer. In the last layer of pooling is the output form of CNN (7, 7, 512). The output is fed into the LSTM layer with an input size of (49, 512). Layer 20 is the output of the LSTM. In the architecture, the input images are processed by a fully linked layer for detection (normal or anomalous).

3.4. Evaluation criteria

This section discusses the evaluation metrics for the proposed model, our CNN-LSTM model is evaluated using the following performance measures in [3]: here we used (a) TP, which stands for positive instances detected, (b) TN, which stands for negative instances detected by the model as negative, (c) FN, which stands for positive instances detected by the model as negative, and (d) FP, which stands for negative instances detected by the model as negative. Accuracy: this can be expressed as the proportion of behavioral instances divided by model correctly detected instances model as (12):

$$Accuracy = \frac{a + b}{a + b + c + d} \quad (12)$$

Precision: this is the portion of instances that are correct from all the detection instances that are positive as shown in (13):

$$Precision = \frac{a}{a + d} \quad (13)$$

Recall: measures the portion of the behavioral instances that is positively detected to the total positive instances as (14):

$$Recall = \frac{a}{a + c} \quad (14)$$

Receiver operating characteristic (ROC) curve: this is a plot of the positive instances correctly detected on the y-axis versus the negative instances detected as negative by the model on the x-axis. AUC: it is the portion of space under the ROC curve that is used to determine the accuracy of model. This implies that if the area is closer the value to 1, the detection method is good; and if the area is closer to 0.5, the detection model is not that good. This can be expressed as (15):

$$AUC = \sum_{i \in \text{positive class}} \frac{\text{rank}_i - \frac{X(1+X)}{2}}{X \times Y} \quad (15)$$

where the number of positive instances is represented as X and the number of negative instances is Y . The performance of the model is further validated using the technique of cross-validation, which requires splitting the training, validation, and testing data into 70%, 15%, and 15%, respectively [24], [25].

4. RESULTS AND DISCUSSION

In this study, the implementation of the systems was performed on the Python 3.6.9 platform (Anaconda 3) along with the associated Python libraries and the Spyder 3 platform IDE for the execution of the Python programs. The following libraries were used to implement the new technique: Keras, OpenCV

library, Tensor flow, Scikit-image, NumPy, SciPy, Pillow/Pill, and Matplotlib. In addition, the DL technique requires a high GPU configuration to train the networks to their maximum size.

4.1. Datasets

To evaluate our CNN-LSTM model, UCSD ped 1 and ped 2 [1], [23] were used for training (contains normal events) and testing (both normal and abnormal events). The UCSD Ped1 data set has 34 training video clips and 36 testing video clips; the UCSD Ped2 data set has 16 training video clips and 12 test video clips. We use Ped1 and Ped2 to indicate UCSD Ped 1 and UCSD Ped 2. Experimental results of the proposed implementation are presented in the next sections.

4.2. Result

Here, we carried out extensive experiments to verify the performance of our CNN-LSTM model on UCSD pedestrian dataset. We also performed experiments using LSTM, Conv-AE, convolution, variational autoencoder, and CNN-LSTM. A different variant of the CNN was tried in our experiment analysis before concluding the final choice of the proposed CNN-LSTM model. The proposed CNN-LSTM model achieved quite promising results as compared with the baseline techniques, which are presented in Table 2. Some of the visual results of the proposed model for anomalous activities in crowded scenes are shown in Figure 3.

Table 2. Comparison of AUC of our CNN-LSTM with other selected models

Model	UCSD Ped 1	UCSD Ped 2
	AUC	AUC
LSTM	0.9183	0.8915
CNN	0.9086	0.9003
Conv-AE	0.9296	0.8847
VAE	0.9348	0.9497
Our CNN-LSTM	0.9547	0.9522

Figures 3(a) and (d) show the original images with anomalous behavior patterns, Figures 3(b) and (e) show the reconstructed images, Figures 3(c) and (f) show the areas where anomalous activities were detected, such as bicyclists and cars moving on pedestrian paths shown in red. In addition, other performance indicators are also used to verify the performance of the proposed model. These are presented in the following sections.

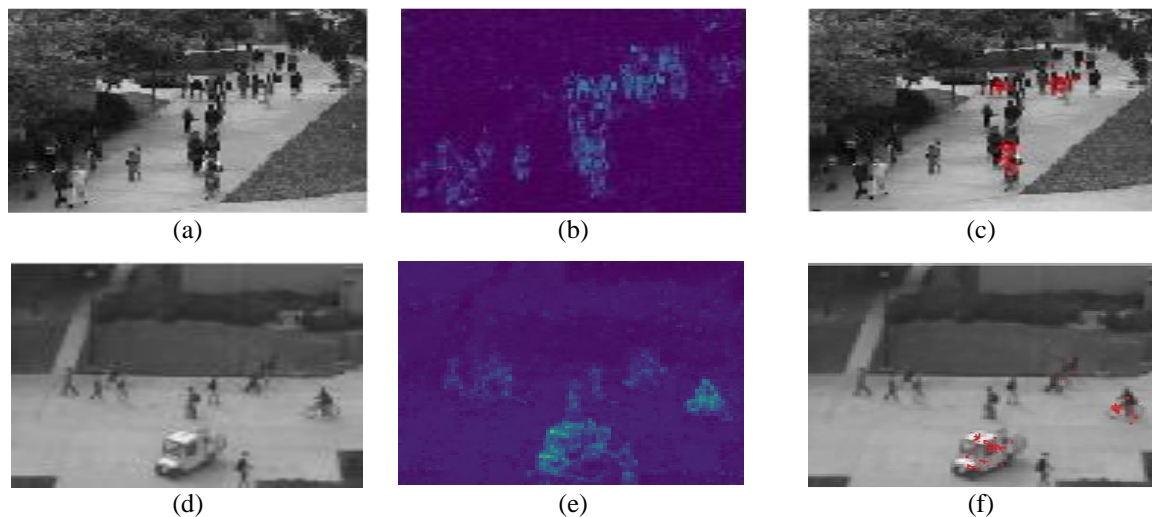


Figure 3. Quantitative observation of anomalous behavioral pattern (a), (d) original image at a time (t), (b), (e) reconstructed image, and (c), (f) detected anomalous using the proposed model

4.3. ROC evaluation

Figure 4 shows the ROC curve used to compare our CNN-LSTM with other baseline detection models for the UCSD ped1 and ped2 datasets. From Figure 4(a), one can observe that the proposed model achieved AUC of 0.9547 which is higher compared to other models. Furthermore, in Figure 4(b), it can be seen that our CNN-LSTM has a superior AUC of 0.9522 when compared with the selected models on ped 2

dataset. This performance is because of the relevant features extracted by CNN and the gate mechanism of LSTM which helps to remember these features during the detection stage.

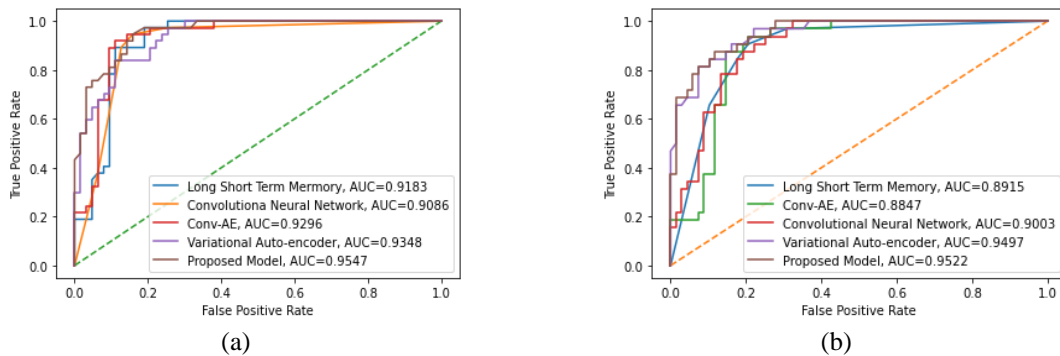


Figure 4. ROC curve (a) ROC curve for ped1 dataset and (b) ROC curve for ped2 dataset

4.4. AUC and accuracy evaluation

Table 2 shows the AUC comparison of our CNN-LSTM model with other detection models. In addition, Table 3 shows the result we obtained from the comparative analysis of our CNN-LSTM model with other detection models in terms of precision (pre), recall (rec), and accuracy (acc) with two different datasets. From Table 3, one can observe that our CNN-LSTM model has higher values in terms of precision, recall, and accuracy on UCSD ped 1 dataset compared to other models. Also, on UCSD ped 2, the proposed model has superior performance over all other selected models.

Table 3. Comparison of accuracy of our CNN-LSTM with the other selected models

	UCSD Ped 1			UCSD Ped 2		
Model	Pre	Rec	Acc	Pre	Rec	Acc
LSTM	0.903	0.923	91.83	0.892	0.815	89.15
CNN	0.898	0.906	90.86	0.889	0.922	90.14
Conv-AE	0.928	0.912	92.96	0.867	0.875	88.47
VAE	0.932	0.902	93.28	0.937	0.929	94.97
Our CNN-LSTM	0.947	0.943	95.47	0.952	0.944	95.22

4.5. Evaluation time

In addition, the average evaluation time of our model is measured and compared with other detection models used in this research with the same image resolution on UCSD datasets ped1 and ped2. Also, with the same GPU and hardware. The results are shown in Table 4. From this, our model has a better average runtime for the two datasets compared to the other detection models used in the experiments.

Table 4. Comparison of evaluation time of our CNN-LSTM with other selected detection methods

Model	UCSD Ped 1 Running time (s)	UCSD Ped 2 Running time (s)
LSTM	0.02283	0.03776
CNN	0.01011	0.01828
Conv-AE	0.02036	0.01519
VAE	0.00848	0.00886
Our CNN-LSTM	0.00403	0.00452

5. CONCLUSION

This paper presents CNN-LSTM architectures as a DL technique for intelligent anomaly detection in surveillance systems. This research focuses on the detection of behavioral patterns that are different from normal patterns in busy scenes. For the experiments conducted, the data from the UCSD pedestrian datasets (ped1 and ped2) were used. The CNN is used for image feature extraction and reconstruction, while the LSTM keeps the extracted features for recall, which helps to drastically improve the time complexity of anomaly detection compared to other benchmark approaches. From the result, our CNN-LSTM performs better compared to the other models.

The superior performances were obtained by optimally training our model with different parameters of Kernel, Kernel size, learning rates, and epoch values until the best recognition rate of 95% was achieved. However, we were careful not to overtrain the model during training to avoid overfitting the data. Many research approaches on DL techniques for anomaly detection in surveillance systems rely on unsupervised learning techniques to improve the performance of the detection model. This method is not yet mature for real-world use because it results in high false alarms and low detection rate. In the future, further research can reduce the false alarms by applying different learning functions to the proposed models. We also plan to implement the proposed model in a real system and test its performance in terms of accuracy.

ACKNOWLEDGEMENTS





Author thanks the Department of Information Technology Systems, Walter Sisulu University for the resources and financial support made available.

REFERENCES





- [1] C. C. Aggarwal, "An Introduction To Outlier Analysis In Outlier Analysis," In: *Outlier Analysis*. Springer, Cham, pp. 1-40, 2016, doi: h10.1007/978-3-319-47578-3_1.
- [2] D. Esan, P. A. Owolawi, and C. Tu, "Anomalous Detection in Noisy Image Frames using Cooperative Median Filtering and KNN," *IAENG International Journal of Computer Science*, vol. 49, no. 1, 2022.
- [3] V. A. Kotkar and V. Sucharita, "A Comparative Analysis Of Machine Learning-Based Anomaly Detection Techniques In Video Surveillance," *Journal of Engineering and Applied Sciences*, no. 12, pp. 9376-9381, 2017, doi: 10.36478/jeasci.2017.9376.9381.
- [4] S. V. Rajenderan and T. Ka Fei, "Real-Time Detection of Suspicious Human Movement," *Proceedings of the International Conference on Electrical, Electronics, Computer Engineering and their Applications*, pp. 56-69, 2016.
- [5] K. C. Baumgartner, S. Ferrari, and C. G. Salfati, "Bayesian Network Modelling of Offender Behaviour for Criminal," Master of Science, Department of Mechanical Engineering and Material science, Duke University, 2016.
- [6] E. L. Piza, J. Caplan, and L. W. Kennedy, "CCTV as a tool for early police intervention: Preliminary lessons from nine case studies," *Security Journal* vol. 30, no. 1, doi: 10.1057/sj.2014.17.
- [7] T. Zhang, Y. Y. Tang, Z. Shang, and X. Liu, "Face Recognition Under Varying Illumination Using Gradientfaces," *IEEE Transactions*, vol. 18, pp. 2599 – 2606, 2017, doi: 10.1109/TIP.2009.2028255.
- [8] NUMBEO. Crime Index [Online] Available: <https://www.numbeo.com/common/>
- [9] S. A. P. S. (SAPS), "Annual Crime Statistics In Republic Of South Africa ", 2019.
- [10] R. Grober, "Violence In South Africa School Is Worse Than You Think, And Spanking Is Part Of The Problem," *Journal of Social Sciences And Humanities* vol. 16, no. 1, pp. 1-12, 2019.
- [11] T. Mhlongo, "The Perceptions And Experience of Students Regarding Weapons In Schools In Umgungundlovu District, KwaZulu-Natal," *Durban University of Technology*, 2017, 2017.
- [12] C. Direkogul, M. Sah, and N. E. O'connor, "Abnormal Crowd Behaviour Detection Using Novel Optical-Flow Based Feature " in: *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*, 2017, doi: 10.1109/AVSS.2017.8078503.
- [13] G. A. A. A. Samra, M. Y. Dahab, and A. A. I. A. Ibrahim, "Real-Time Anomalous Behavior Detection of Students in Examination Rooms Using Neural Networks and Gaussian Distribution," *International Journal of Scientific and Engineering Research (IJSC)*, pp. 1-10, 2018.
- [14] T. Wang and H. Snoussi, "Detection of Abnormal Events via Optical Flow Feature Analysis," *Sensors* vol. 15, no. 4, pp. 7156-7171, 2015, doi: 10.3390/s150407156.
- [15] M. Giuhoski, N. Marcelo, R. Aquino, M. Ribeiro, A. E. Lazzaretti, and H. S. Lopes, "Detection of Video Anomalies Using Convolutional Autoencoder and One-Class Support Vector Machine," *presented at the Brasillero Of Intelligence Computational 2017*, 2017, doi: 10.21528/CBIC2017-49.
- [16] P. Afsar, P. Cortez, and H. Santos, "Automatic Human Action Recognition from Video Using Hidden Markov Model," *IEEE 18th International Conference on Computational Science and Engineering*, 2016, doi: 10.1109/CSE.2015.41.
- [17] A. M. Kamoona, A. K. Gostary, A. Bab-Hadiasharz, and R. Hoseinneshad, "Sparsity-Based Naive Bayes Approach for Anomaly Detection in Real Surveillance Videos," in *2019 International Conference on Control, Automation and Information Sciences (ICCAIS)*, 2019.
- [18] X. Gu, L. Akoglu, and A. Rinaldo, "A Statistical Analysis of Nearest Neighbor Methods For Anomaly Detection," *33rd Conference on Neural Information Processing Systems (NeurIPS 2019)*, 2019.
- [19] B. Yang, X. Fu, N. D. Sidiropoulos, and M. Hong, "Towards K-Means-Friendly Spaces: Simultaneous Deep Learning And Clustering.," *ICLM*, 2017.
- [20] D. O. Esan, P. A. Owolawi, and C. Tu, "Detection of Anomalous Behavioural Patterns In University Environment Using CNN-LSTM," *IEEE 23rd International Conference on Information Fusion (FUSION)*, pp. 1-8, 2020, doi: 10.1109/CSCI51800.2020.00012.
- [21] S. Albelwi and A. Mahmood, "A Framework for Designing the Architectures of Deep Convolutional Neural Networks," *Entropy*, vol. 19, no. 6, 2017, doi: 10.3390/e19060242.
- [22] O. A. Esan and I. O. Osunmakinde, "Towards Intelligent Vision Surveillance for Police Information Systems," In: *Silhavy, R. (eds) Cybernetics Perspectives in Systems. CSOC 2022. Lecture Notes in Networks and Systems*, vol. 503, 2022, doi: 10.1007/978-3-031-09073-8_13.
- [23] X. Hu, S. Hu, X. Zhang, H. Zhang, and L. Luo, "Anomaly detection based on local nearest neighbour distance descriptor in crowded scenes," *The Scientific World Journal* vol. 2014, no. 6, 2014, doi: 10.1155/2014/632575.
- [24] K. Deshpande, N. S. Pun, S. K. Sonbhadra, and S. Agarwal, "Anomaly detection in surveillance videos using transformer based attention model," *In International Workshop on Urban Computing*, pp. 1-6, 2022.
- [25] S. W. Khan et al., "Anomaly Detection in Traffic Surveillance Videos Using Deep Learning," *Sensors*, vol. 22, no. 17, pp. 1-28, 2022, doi: 10.3390/s22176563.

BIOGRAPHIES OF AUTHORS







Omobayo Ayokunle Esan     he is completing his Ph.D in Computer Science from the University of South Africa (UNISA). He is a lecturer in the Department of Information Technology Systems at Walter Sisulu University, South Africa. His research interests include image processing, machine learning, computer vision, cybersecurity, and the internet of things (IoT). He can be contacted at email: oesan@wsu.ac.za.







Dorcas Oladayo Esan     Currently, she is starting her Ph.D in Computer Systems Engineering at Technology Tshwane University of Technology (TUT), South Africa. Her research interests include machine learning, computer vision, and data mining. She can be contacted at email: alakedo@tut.ac.za.







Munienge Mbodila     he is completing his Ph.D in Computer Science from Northwest University South Africa. He is the head of the department in the department of Information Technology Systems at Walter Sisulu University, South Africa. His research interests are wireless networks, computer networks, ICT in education and the use of ICT in teaching student engagement. He can be contacted at email: mmbodila@wsu.ac.za.



Femi Abiodun Elegbeleye     he is a Ph.D student in Computer Science from Northwest University, South Africa in the Department of Computer Science. He is a lecturer in the department of Information Technology Systems at Walter Sisulu University, South Africa. His research interests include wireless networks and computer networks. He can be contacted at email: felegbeleye@wsu.ac.za.



Kesewaa Koranteng     she holds a Ph.D degree in Information Technology from the University of Cape Town, South Africa in the Department of Information Technology Systems. She is a lecturer in the department of Information Technology Systems at Walter Sisulu University, South Africa. Her research interests include e-learning and ICT technology in the classroom. She can be contacted at email: kkoranteng@wsu.ac.za.