IIETA International Information and Engineering Technology Association
Advancing the World of Information and Engineering

Ingénierie des Systèmes d'Information
Vol. 29, No. 5, October, 2024, pp. 2031-2038
Journal homepage: http://iieta.org/journals/isi

Deep Learning Based Multistage Approach for Anomaly Detection

Megha G. Pallewar[1*], Vijaya R. Pawar[2], Arun N. Gaikwad[1]

[1] E&TC Engineering, Zeal College of Engineering & Research, Savitribai Phule Pune University, Pune 411041, India
[2] E&TC Engineering, Bharati Vidyapeeth's College of Engineering for Women, Savitribai Phule Pune University, Pune 411043, India

Corresponding Author Email: msyannawar@gmail.com

https://doi.org/10.18280/isi.290534

Received: 14 December 2023
Revised: 30 August 2024
Accepted: 14 September 2024
Available online: 24 October 2024

Keywords:
anomaly detection, human activity detection, deep learning, convolutional neural networks (CNN), long short time memory (LSTM), accuracy

ABSTRACT

Anomaly detection is the process of identifying patterns or data points that substantially depart from the normal or expected behavior and nowadays it is applied in different fields. The ultimate aim is to detect unusual trends or outliers that could correspond to problems, the future issues, or interesting findings. Machine learning-based classification and recognition methods have emerged as the leading approach for detecting anomalous activities. This work specifically focuses on the detection of anomalous activities in videos, employing the UCF crime database as the primary dataset. The proposed method works with a multistage approach, integrating convolutional neural network (CNN) and long short-term memory (LSTM). The image frames from the videos are utilized as input to the CNN, which processes and extracts salient features. Through the cascading of CNN and LSTM, the system successfully identifies twelve distinct anomalous activities: Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fight, Road Accident, Robbery, Stealing, Shoplifting, and Vandalism. To train and evaluate the system, an 80-10-10 split of the dataset is employed for training, testing, and validation respectively, including cross-validation steps. The present system attains 99% accuracy showcasing its robustness in anomalous activity detection.

1. INTRODUCTION

In today's demanding world, automated video surveillance has become a crucial concern, playing a pivotal role in both individual and national security. Video from surveillance cameras can be analyzed manually, semi-automatically, or entirely automatically to find anomalies or unexpected events. Conventional surveillance systems are entirely manual and reliant on people, demanding the workforce's continuous observation and attention to analyze events or behaviors in order to determine whether the recorded activities are suspicious or unusual. Multiple video feeds must be observed by the human operators at once, which are laborious and extremely hard on their attention span, resulting in poor performance. Security officers typically use a display screen attached to a video camera to watch unusual events such as traffic accidents, fires, explosions, robberies, stampedes, etc. Human inspection of surveillance footage is not optimal, though, it requires prolonged periods of focus. As a result, an automated system is needed to identify and detect anomaly or anomalous activity.

The primary aim of automated video surveillance lies in distinguishing between normal and abnormal activities. Monitoring devices are deployed to continuously observe activities in the public spaces and identify any behavior that deviates from the normal behavior. The core objective is to promptly detect unusual human behavior, particularly in in circumstances where stopping crime and mitigating security issues require quick action. The UCF Crime dataset provides a comprehensive list of thirteen anomalous acts, including Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fight, Road Accident, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. Efficient tracking systems that can autonomously identifying suspicious conduct are crucial for maintaining safety. To achieve this objective, an in-depth study of human activity recognition is imperative. This involves gaining a thorough understanding of the unique characteristics of each action.

The rest of the paper is structured as follows to provide a comprehensive overview of the work. In section II related work portraits, the various methodologies used similar field. Section III describe the present methodology. Section IV elaborates the experimentation and its results. Section V discussion on performance parameters and its analysis. The last section VI is about conclusion and future scope.

2. RELATED WORK

The research published both regionally and worldwide has been thoroughly surveyed and a summary of the key finding and insight of the research are discussed below. In computer vision, anomaly detection is one of the most difficult and enduring problems [1-5]. Initially basic models were

employed to verify various performance parameters. Using the 3D CNN technique the accuracy of system is obtained from 81.8% to 94.6%.

Yuan et al. [6] found that established methods are restricted to detect and classify the already defined events and the methodologies. To address this issue, they constructed very unique modal surveillance video dataset by meticulously adding fine-grained event content and timing annotations to the real-world surveillance dataset UCF-Crime. Wang and Chen [7] developed a Dual-Stream Memory Network (DSM-Net) module to provide the anomaly detection network with more information. Writing and reading is done by the memory module using a queue of data features. A moving average encoder is used in the writing process to capture the historical data of video events, and optical flow is employed in the reading process to identify behavioural patterns in RGB images. Further Patwal et al. [8] developed system with a low-cost algorithm for detecting crowd irregularities. The model uses the convolutional neural network based on DenseNet121 as the feature extractor. The established framework has an AUC of 86.63% on the UCF-Crime dataset. The development mainly focused on detecting behavioural analysis from the gained data attributes. Kumar et al. [9] firstly employed an attention mechanism, a bidirectional long short-term memory (Bi-LSTM), and a convolutional neural network (CNN) to extract the distinct spatiotemporal properties of unprocessed video streams. This work uses a deep learning approach to identify unusual human activity. They used variety of datasets UCF11, UCF50, and UCF Crime to obtain the results. The pre-trained models [10] are also very commonly used to enhance the anomaly detection performance.

The present approach is a multistage approach that addresses the limitations of current automated video surveillance systems, which often depend on predefined and restricted parameters. The system introduces a more adaptable framework, allowing the basic attributes to be modified according to the complexity of the targeted activity. This adaptable, unconstrained parameter strategy is implemented using classifiers based on machine learning. The proposed principal model is designed to cover a wide range of anomalous activities from the UCF crime dataset, achieving improved performance in detecting these events.

3. THE PRESENT METHODOLOGY

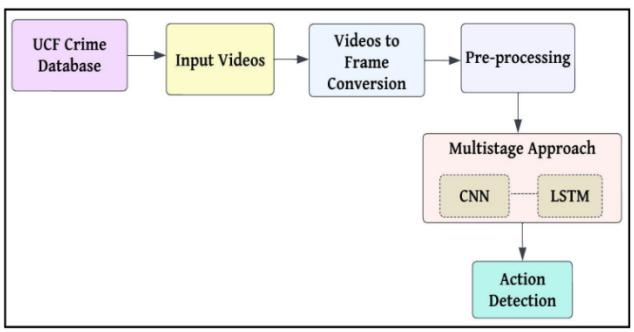

Figure 1. The block diagram of CNN and LSTM combination model for anomalous activity detection

The process for video-based recognition of anomalies using LSTM and convolutional neural networks is shown in Figure 1. Acquisition of datasets, extracting the videos and frames from datasets are its components. Various pre-processing techniques are like Histogram of Gradients (HoG) is applied. Filtered images are obtained and applied as an input to multistage approach. The step-wise flow of research work is shown in Figure 2.

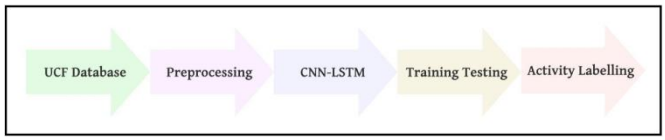

Figure 2. The stepwise flow of work

3.1 UCF crime database

The previous datasets used for anomaly detection were quite small in size, Sultani et al. [11] created the UCF-Crime Dataset. It contains unprocessed surveillance videos capturing 13 common real-world crime scenarios which include theft, shoplifting, vandalism, fighting, robbery, explosion, and accidents. The activities included are namely Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fight, Road Accident, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. The UCF crime dataset consist of 128 hours of footage comprising 1900 videos, each with 7247 average frames. These videos are evenly divided in two categories: 950 are ordinary recordings and the other 950 are uncut real-world surveillance recordings. With 70 percentages of the videos typically running for about four minutes, this dataset is particularly valuable for addressing the challenge of video categorization.

To achieve the primary objective of the system, it is essential to have wide variety anomalous activities. This requirement is effectively met by the UCF Crime dataset which includes 13 distinct and relevant actions, providing a comprehensive foundation for anomaly detection.

3.2 Multistage approach with layered CNN-LSTM

CNN is a very efficient algorithm which gives good results and it verified by different models developed [12-15]. When CNN and LSTM are combined into one model, it becomes possible to extract features from spatial data (which CNN handles well) and capture temporal dependencies in sequences (which LSTM handles well). The layered architecture has two distinct layers.

CNN Layers: CNN layers are usually the first layers that are used to extract spatial information from the input data. Pooling and convolutional procedures are possible components of these layers. Figure 3 shows the working architecture of CNN model.

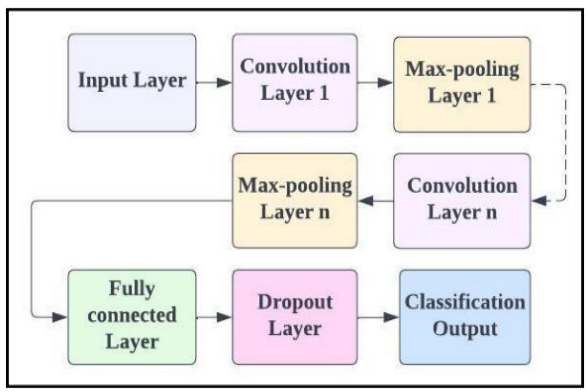

Figure 3. CNN model architecture

The input image maintains a consistent size of 224×224×3 throughout the process. It is then processed through a series of convolutional layers stacked together, with ReLU (Rectified Linear Unit) serving as the activation function for each layer, involving 4,096 nodes. Through the application of max-pooling techniques, the output feature vectors' dimensions are reduced in the CNN, improving the representation's computational efficiency and manageability. In the dense layer, the Softmax activation function is employed, which is crucial for making the final classification decisions. This layer contains approximately 1,500 nodes. For a detailed overview of the implemented CNN model, refer to Table 1, which provides a summary of each layer and the associated parameter count. The convolutional layer extracts a collection of spatial features which are then pass on to the long short term memory layers for further processing. A collection of spatial features is what these layers produce, and these are subsequently sent to the LSTM layers.

Table 1. Layer-wise summary of CNN model

Layer (Type)	Parameters
Rescaling 1	0
Conv2D	448
Maxpooling2D	0
Conv2D 1	4640
Maxpooling2D 1	0
Conv2D 2	18496
Maxpooling2D 2	0
Flatten	0
Dense	3965056
Dense 1	1548
Total Parameters	3990188

Layers of LSTM: A typical LSTM model architecture is shown in Figure 4. Temporal dependencies are captured by the LSTM layers through their sequential processing of the spatial information. Understanding long-range dependencies in the data is made possible by the internal memory state that the LSTM layers maintain. Regression, classification, and other tasks can be performed with the final output of the LSTM layers [16-18].

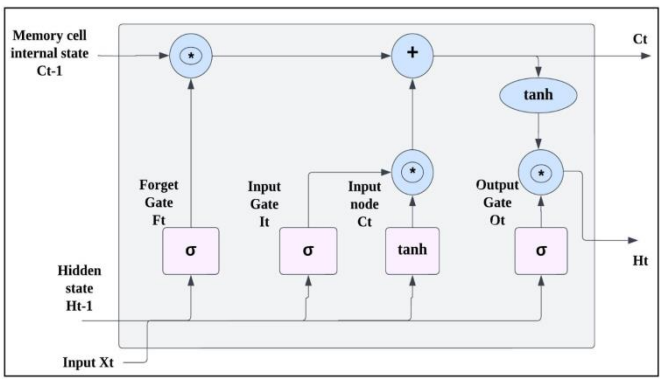

Figure 4. The LSTM model architecture

LSTM architectures typically incorporate three fundamental types of logical gates that serve as the building blocks of the system. The essential elements within an LSTM are h_t and c_t, which play a crucial role in its functioning. The following Figure 5 elaborates all equations which are used to evaluate and determine the two units, h_t and c_t.

The σ is the sigmoidal function, φ is the hyperbolic tangent, ⊗ represents the product with the value of the gate and the weights of the matrix denoted by W_{ij}.

The LSTM model summary is provided in Table 2.

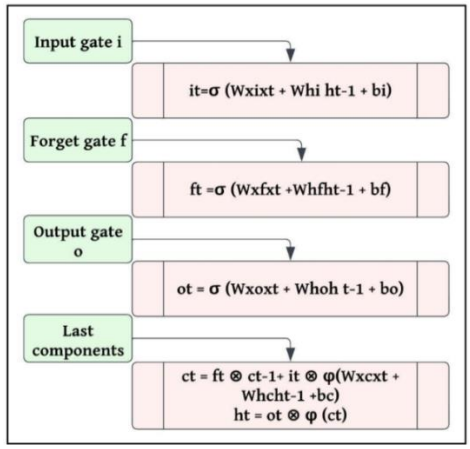

Figure 5. The LSTM model with elements and equations

Table 2. Layer-wise summary of LSTM model

Layer (Type)	Number of Parameters
Input Layer	0
LSTM	1,28,7168
LSTM	197,120
LSTM	49,408
Flatten	0
Dense	491,776
Dropout	0
Dense	32,896
Dropout	0
Dense	258

CNN-LSTM multistage architecture is explained in Figure 6. Initially, the process involves extracting frames from videos, setting a threshold of 40 frames. This guarantees that the input of the video is fully standardized into frames, and the later processing is done by selecting these 40 frames. Using eighty percent of the videos in the database, the CNN model is trained. Then, the features and components of the video are extracted using the competent typical CNN architecture. A total of 25088 features are garnered from each frame, resulting in 40 sets of 25088 features. These are subsequently fed into the Long Short-Term Memory (LSTM), where each vector is processed step-by-step. The output was finally recovered from the model's dense layer.

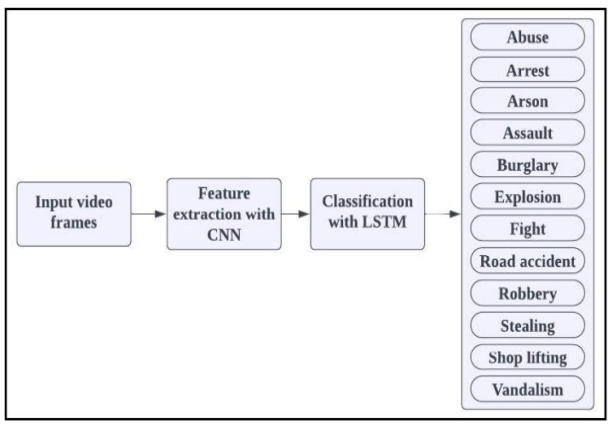

Figure 6. System architecture

4. EXPERIMENTATION AND RESULTS

An extensive experimentation is done by using the environment where present system performs with Python 3.7.0 version and is implemented on 2.7 GHz Intel Core i7-4600U CPU 8GB RAM 64 bit Operating system.

The experiment involved testing with different frame counts, specifically 20, 40, and 60 frames. Prior to testing, training was conducted for every single group of activity using 40 frames. The analysis process commenced with 10, 20, and 30 epochs, while also varying batch sizes, filter counts, and dense layer counts. In total, 30 epochs were employed to obtain the results. The initial work started with classification of only two activities then it further extended with six activity classifications and labeling. As the model is performing well for both rules, now the focus extended to classification of 12 activities. The 12-activity class includes Abuse, Arrest, Arson, Assault, Burglary, Explosion, Fight, Road Accident, Robbery, Stealing, Shoplifting, and Vandalism. The model worked perfectly for six classes [19].

Figure 7 shows the class wise result and 12 class activity labelling respectively. For forty sample frames, all actions are identified. For every class, the model underwent comparable testing for 60, 80, and 100 frames.

Figure 7. 12 class activity labelling

Table 3 shows the video classification result for various methodologies used. The AUC and Accuracy is compared with other methodologies. As the utmost accuracy for anomaly detection, the present system achieves 98.6% accuracy.

The tabular comparative analysis provides an overview of some commonly used methodologies in the field of anomaly detection. This analysis allows us to evaluate and compare different approaches based on key factors such as the dataset used, accuracy and Area Under the Curve (AUC) values. Here the primary focus is on discussing the most effective machine learning algorithms and how they perform relative to these benchmarks. The multistage CNN-LSTM approach, in particular, stands out for its ability to handle multidimensional dataset while still achieving superior accuracy and AUC values. This method effectively combines the strengths of CNNs for spatial feature extractions for LSTMs for temporal sequence modelling leading to improved performance in detecting anomalies.

Table 3. Result of video classification

Model Used	Year	Dataset	Accuracy	AUC
CNN-Bi LSTM [9]	2023	UCF11 UCF50 UCF Crime	98.9 96.04 61.04	--
CubicSVM KNN [15]	2023	CIFAR100	99.24	--
VAE [20]	2021	UCSDped1 Avenue	--	92.3 82.1
DenseNet121 [21]	2023	UCF Crime	--	86.63
ST-GCN [22]	2023	UCSD Ped1 UCSD Ped2 ShanghaiTech	89.1 - -	85.5 97.9 83.8
CNN-LSTM (Proposed model)	2022	UCF Crime	98.6	92.8

5. PERFORMANCE ANALYSIS

Performance parameter like accuracy, specificity, recall, precision etc. are considered for the performance analysis of the present work.

The detail analysis is discussed here.

5.1 Performance metrics

The evaluation metrics to analyse the effectiveness of the implemented model is discussed below.

Accuracy: To determine the percentage of values that were successfully classified, accuracy has been used. It indicates the number of times the used classifier is accurate. It is calculated by dividing the total values by the sum of all true values.

$$\text{Accuracy} = (\text{TP} + \text{TN})/(\text{TP} + \text{TN} + \text{FP} + \text{FN}) \quad (1)$$

Precision: The ability of the model to accurately classify positive values is computed using precision. It is calculated by dividing the total number of projected positive values by the true positives.

$$\text{Precision} = \text{TP}/(\text{TP} + \text{FP}) \quad (2)$$

Recall: The model's predictive capacity for positive values is computed using it. "What is the frequency of the model's correct positive value predictions?" It is the true positives divided by the total number of actual positive values.

$$\text{Recall} = \text{TP}/(\text{TP} + \text{FN}) \quad (3)$$

F1-Score: It is the precision and recall harmonic mean. When precision and recall must be considered simultaneously, it can be helpful.

$$\text{F1score} = 2 * (\text{precision} * \text{recall})/(\text{precision} + \text{recall}) \quad (4)$$

The ratio of accurate negative predictions to the total number of negatives is used to calculate specificity. Another name for it is true negative rate (TNR). Specificity ranges from 0 to 1, with 1 being the finest.

$$\text{TNR / Specificity} = \frac{\text{TN}}{\text{TN+ FP}} \qquad (5)$$
$$\text{FPR} = \text{FP} / (\text{TN} + \text{FP})$$

The performance parameters precision, recall, F1 score and accuracy can be better understood by the graphical representation. Also, the class wise comparison for the single parameter can be easily done by Figure 8 and Figure 9.

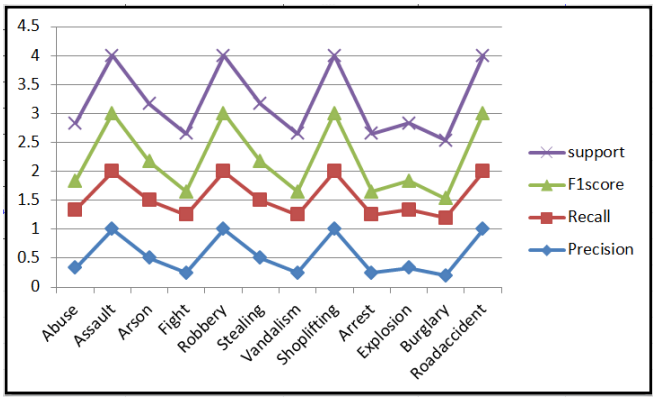

Figure 8. Class wise performance parameters

5.2 Training and validation accuracy and loss curves

The training and validation loss values are crucial for assessing model performance as they provide insight into how the model's learning evolves over epochs. These metrics help diagnose issues related to underfitting or overfitting and guide the selection of optimal model weights for inference.

Training and Validation Loss are vital parameters for defining the model performance, as discussed below.

Training Loss: Represents how well the model fits the training data. A decrease in training loss over epochs indicates that the model is learning and improving its performance on the training set.

Validation Loss: Reflects how well the model generalizes to unseen data. A decreasing validation loss suggests that the model is also performing well on the validation set, indicating good generalization.

The loss values are typically plotted against the number of epochs to visualize learning performance. The Figure 10 depicts the training and validation accuracy curves for 2 and 10 epochs. It also shows that accuracy is increased as the number of epoch increases. It indicates that the model's performance improves as it continues to train. The Figure 11 shows the training and validation loss curves for 2 and 10 epochs. It reveals that the loss function decreases as the number of epochs increases, indicating that the model becomes better at minimizing errors over time.

By analyzing these curves, it can be determined:

(1) Learning Trends: A decreasing loss and increasing accuracy with more epochs generally indicate effective learning.

(2) Optimal Epochs: The epoch where the validation loss stabilizes or starts to increase can be chosen to avoid overfitting. This epoch represents the point where the model weights should be used for inference.

Monitoring these metrics helps in selecting the best model configuration and preventing potential issues such as overfitting or underfitting.

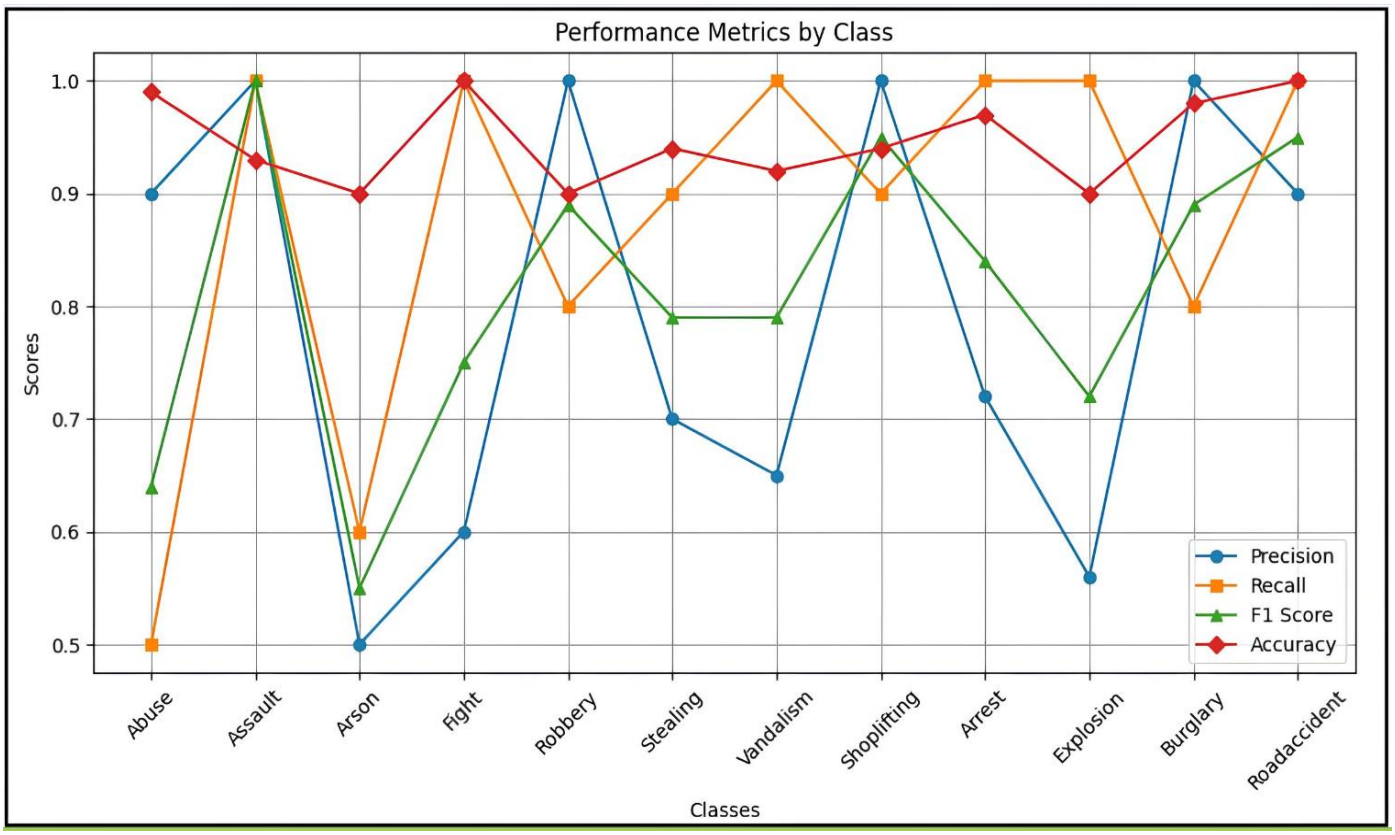

Figure 9. Class wise line graph for precision, recall, accuracy and F1 score

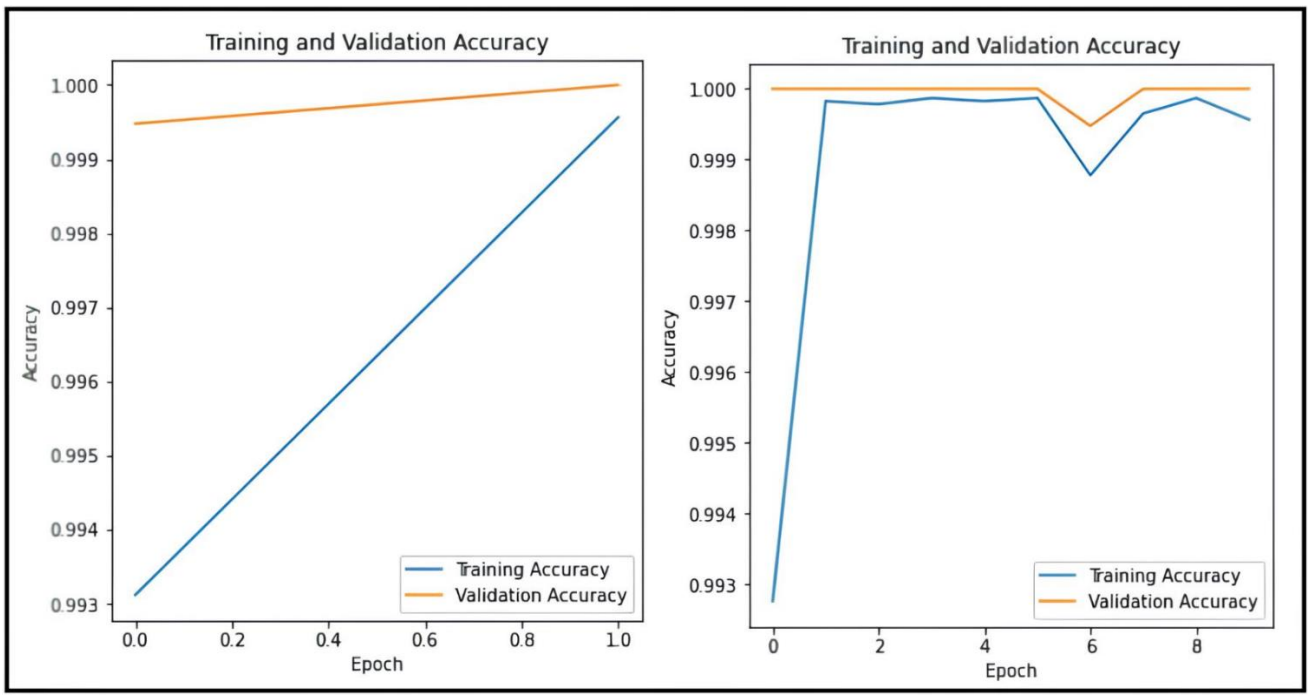

Figure 10. Training and validation accuracy for 2 and 10 epoch

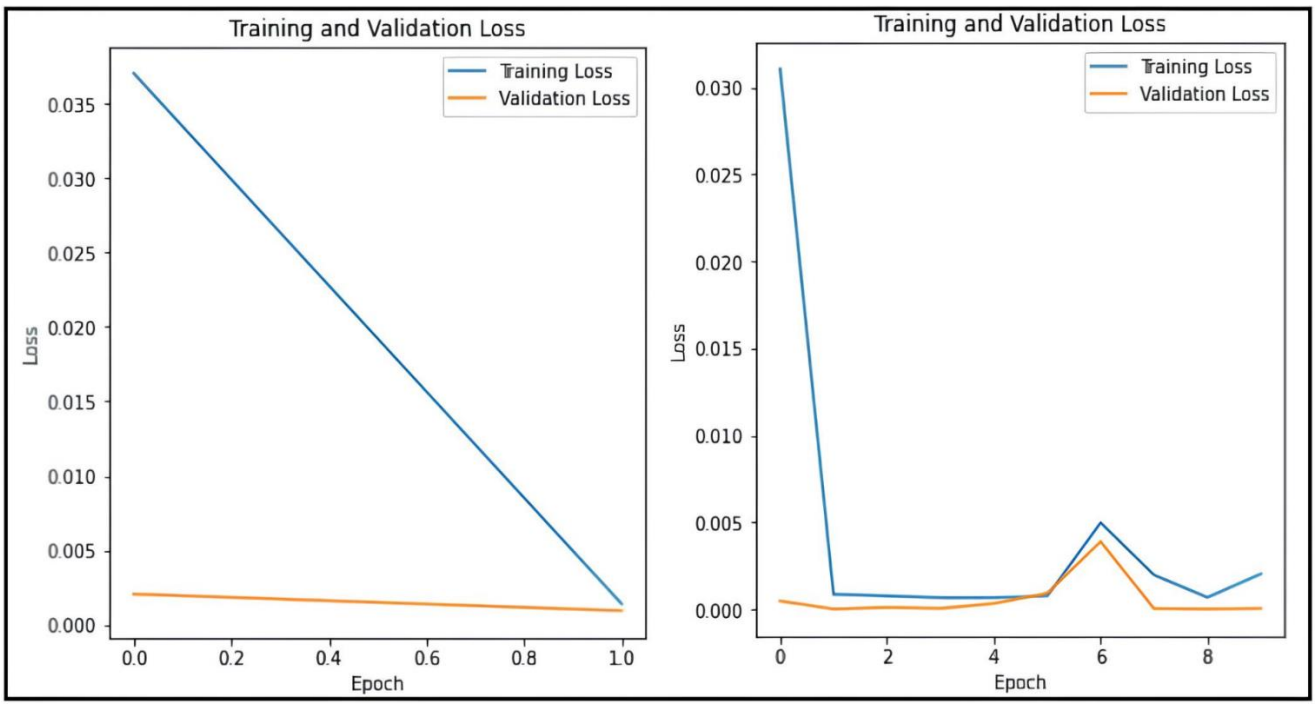

Figure 11. Training and validation loss for 2 and 10 epoch

5.3 Confusion matrix

A confusion matrix provides a comprehensive overview of the performance of a classification model by displaying a table layout of the prediction verses actual outcome. The confusion matrix shows individual class-wise accuracy, through which one can identify the percentage accuracy of all 12 classes which is shown in Figure 12.

The experimental results of the CNN-LSTM model are derived on a most challenging dataset that is UCF Crime dataset. The other publicly available databases are having single viewed, similar background, typical angle videos but in UCF crime dataset there is a great variety of actions and videos for single action. Other datasets commonly used basic actions whereas UCF dataset comprises crime activities happened at real time grounds and captured by CCTVs.

The developed system faced main challenge during experimentation is the processing time required for model performance. This is very crucial thing which effects on the overall performance on the video surveillance system. The response time required foe the system must be as small as possible so that the corrective actions can be taken in consideration with the anomaly detected through proposed system.

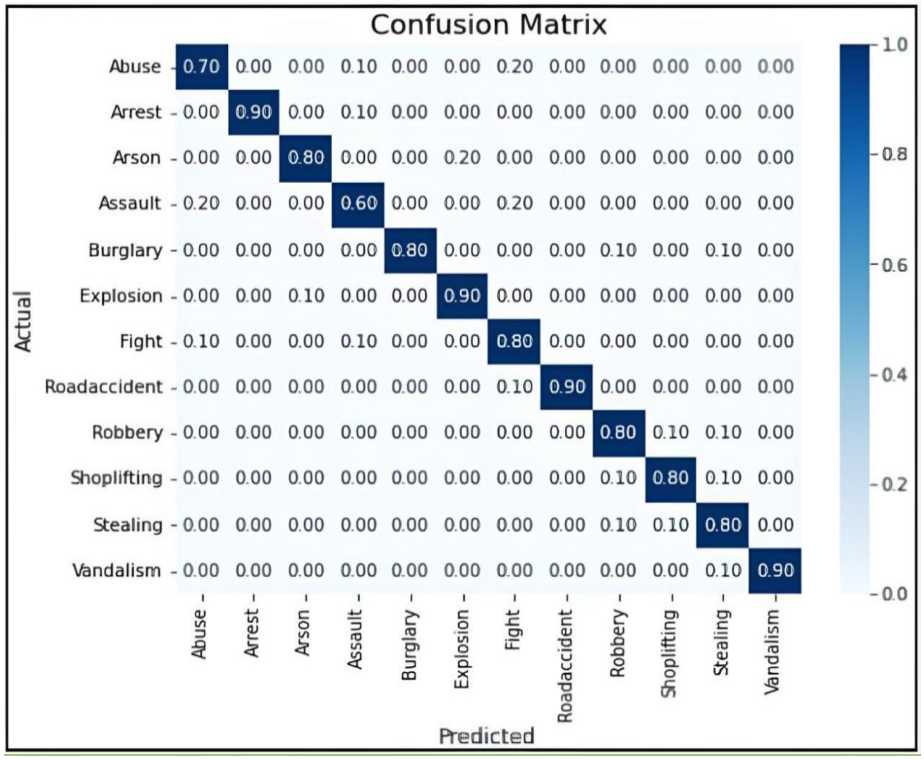

Figure 12. Confusion matrix for 12 activity recognition

6. CONCLUSION AND FUTURE SCOPE

The experimental findings show that on the UCF Crime dataset, the current method performs remarkably well. A variety of anomalous behaviors that are most likely to take place in public spaces are successfully identified by the CNN-LSTM combination model. This proposed system can be utilized to enhance human safety by feeding the detected output to an indicator device, system, or authorized organization that can respond to such abnormal behavior in communal spaces. In future work, work can be extended to handle all 13 classes of the UCF Crime dataset. This can be anticipated that similar successful results can be achieved in this broader context. However, one drawback of present model is the lengthy training time. By concentrating on areas of the scene where action is observed and disregarding unimportant portions of the frame, this problem can be lessened. This approach would reduce the system's response time, thereby improving overall performance metrics. The more accurate real time video surveillance system can be developed by implementing self-supervised learning approaches like MoCo (Momentum Contrast) and SimCLR (Simple Framework for Contrastive Learning of Visual Representations) or by using multimodal models like MAE (Masked Autoencoders).

ACKNOWLEDGEMENT

The authors of this work sincerely thank Mr. Sham Yannawar and Mr. Ganesh Pallewar for their unwavering support and essential advice and assistance during this project. I appreciate your support and presence throughout this entire process, which has inspired me. We would like to extend our gratitude to Dr. A.M. Kate and Dr. M.G. Unde for providing us with the required facilities and their kind support.

REFERENCES

[1] Dubey, S., Boragule, A., Jeon, M. (2019). 3D ResNet with ranking loss function for abnormal activity detection in videos. In 2019 international conference on control, automation and information sciences (ICCAIS), Chengdu, China, pp. 1-6. https://doi.org/10.1109/ICCAIS46528.2019.9074586

[2] Singh, V., Singh, S., Gupta, P. (2020). Real-time anomaly recognition through CCTV using neural networks. Procedia Computer Science, 173: 254-263. https://doi.org/10.1016/j.procs.2020.06.030

[3] Huang, W., Liu, Y., Zhu, S., Wang, S., Zhang, Y. (2020). TSCNN: A 3D convolutional activity recognition network based on rfid rssi. In 2020 International Joint Conference on Neural Networks (IJCNN), Glasgow, UK, pp. 1-8. https://doi.org/10.1109/IJCNN48605.2020.9207590

[4] Alaghbari, K.A., Saad, M.H.M., Hussain, A., Alam, M.R. (2022). Activities recognition, anomaly detection and next activity prediction based on neural networks in smart homes. IEEE Access, 10: 28219-28232. https://doi.org/10.1109/ACCESS.2022.3157726

[5] Vrskova, R., Hudec, R., Kamencay, P., Sykora, P. (2022). A new approach for abnormal human activities recognition based on ConvLSTM architecture. Sensors, 22(8): 2946. https://doi.org/10.3390/s22082946

[6] Yuan, T., Zhang, X., Liu, K., Liu, B., Jin, J., Jiao, Z. (2023). UCF-Crime Annotation: A Benchmark for Surveillance Video-and-Language Understanding. https://doi.org/10.48550/arXiv.2309.13925

[7] Wang, Z., Chen, Y. (2023). Anomaly detection with dual-stream memory network. Journal of Visual Communication and Image Representation, 90: 103739. https://doi.org/10.1016/j.jvcir.2022.103739

[8] Patwal, A., Diwakar, M., Tripathi, V., Singh, P. (2023).

An investigation of videos for abnormal behavior detection. Procedia Computer Science, 218: 2264-2272. https://doi.org/10.1016/j.procs.2023.01.202

[9] Kumar, M., Patel, A.K., Biswas, M., Shitharth, S. (2023). Attention-based bidirectional-long short-term memory for abnormal human activity detection. Scientific Reports, 13(1): 14442. https://doi.org/10.1038/s41598-023-41231-0

[10] Duong, H.T., Le, V.T., Hoang, V.T. (2023). Deep learning-based anomaly detection in video surveillance: A survey. Sensors, 23(11): 5024. https://doi.org/10.3390/s23115024

[11] Sultani, W., Arshad, Q.A., Chen, C. (2020). Action recognition in real world videos. Computer Vision, 1-11. https://doi.org/10.1007/978-3-030-03243-2_846-1

[12] Tang, Y., Teng, Q., Zhang, L., Min, F., He, J. (2020). Layer-wise training convolutional neural networks with smaller filters for human activity recognition using wearable sensors. IEEE Sensors Journal, 21(1): 581-592. https://doi.org/10.1109/JSEN.2020.3015521

[13] Wang, S., Mesaros, A., Heittola, T., Virtanen, T. (2021). A curated dataset of urban scenes for audio-visual scene analysis. In ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Toronto, ON, Canada, pp. 626-630. https://doi.org/10.1109/ICASSP39728.2021.9415085

[14] Kondratyuk, D., Yuan, L., Li, Y., Zhang, L., Tan, M., Brown, M., Gong, B. (2021). Movinets: Mobile video networks for efficient video recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 16020-16030. https://doi.org/10.1109/CVPR46437.2021.01576

[15] Saba, T., Rehman, A., Latif, R., Fati, S.M., Raza, M., Sharif, M. (2021). Suspicious activity recognition using proposed deep L4-branched-ActionNet with entropy coded ant colony system optimization. IEEE Access, 9: 89181-89197. https://doi.org/10.1109/ACCESS.2021.3091081

[16] Ullah, I., Mahmoud, Q.H. (2022). Design and development of RNN anomaly detection model for IoT networks. IEEE Access, 10: 62722-62750. https://doi.org/10.1109/ACCESS.2022.3176317

[17] Rehman, A., Saba, T., Khan, M.Z., Damaševičius, R., Bahaj, S.A. (2022). Internet of things based suspicious activity recognition using multimodalities of computer vision for smart city security. Security and communication Networks, 2022(1): 8383461. https://doi.org/10.1155/2022/8383461

[18] Sherif, E., Mohamed, H., Agwad, E., Mostafa, M. (2020). Deep learning based crowd scene analysis Survey. Journal of Imaging, 11.

[19] Pallewar, M.G., Pawar, V.R., Gaikwad, A.N. (2024). Human Anomalous Activity detection with CNN-LSTM approach. Journal of Integrated Science and Technology, 12(1): 704. https://pubs.thesciencein.org/journal/index.php/jist/article/view/a704.

[20] Gangloff, H., Pham, M.T., Courtrai, L., Lefèvre, S. (2022). Leveraging vector-quantized variational autoencoder inner metrics for anomaly detection. In 2022 26th International Conference on Pattern Recognition (ICPR), Montreal, QC, Canada, pp. 435-441. https://doi.org/10.1109/ICPR56361.2022.9956102

[21] Solanki, S., Shah, Y., Rohit, D., Ramoliya, D. (2023). Unveiling anomalies in surveillance videos through various transfer learning models. In 2023 International Conference on Self Sustainable Artificial Intelligence Systems (ICSSAS), Erode, India, pp. 300-306. https://doi.org/10.1109/ICSSAS57918.2023.10331722

[22] Yang, Y., Fu, Z., Naqvi, S.M. (2023). Abnormal event detection for video surveillance using an enhanced two-stream fusion method. Neurocomputing, 553: 126561. https://doi.org/10.1016/j.neucom.2023.126561