iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 12 **Issue:** IX **Month of publication:** September 2024

DOI: https://doi.org/10.22214/ijraset.2024.64181

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 12 Issue IX Sep 2024- Available at www.ijraset.com

A Comprehensive Comparative Analysis of Deep Learning Architectures for Suspicious Activity Detection in Video Surveillance

Kashish D.Thanki[1], Bhargav V. Patel[2], Shaifali Malukani[3]
Department of Computer Engineering, Dharmsinh Desai University, Nadiad, Gujarat, India - 387001

Abstract: ***In the landscape of modern security infrastructure, video surveillance has evolved into a ubiquitous and indispensable tool for ensuring public safety and safeguarding critical assets. This research paper delves into an extensive examination of various deep learning architectures for the purpose of detecting suspicious activities in video surveillance. The investigation encompasses Convolutional Long Short-Term Memory (ConvLSTM), Convolutional Neural Network (CNN) combined with Long Short-Term Memory (LSTM), ConvLSTM, Bidirectional Long Short-Term Memory (BiLSTM), and diverse combinations thereof. Each model is rigorously trained and tested on a carefully curated dataset designed to encapsulate a spectrum of normal and suspicious activities like shooting and fighting. The study aspires to identify the most efficacious model for enhancing the accuracy of suspicious activity detection in complex and dynamic environments. Metrics such as accuracy, precision, recall, and F1 score will be rigorously assessed to ascertain the model's comparative performances.***
Keywords: Video surveillance, Deep Learning, Suspicious activity, Shooting, Fighting, Anomaly detection.

I. INTRODUCTION

Video surveillance has become an integral component of modern security systems, providing a ubiquitous layer of vigilance across public spaces, critical infrastructure, and private establishments [14]. With the proliferation of surveillance cameras, the need for advanced techniques to detect and respond to suspicious activities within the vast volumes of video data has emerged as a critical research frontier [11]. Suspicious activity detection is a complex task, often requiring sophisticated algorithms capable of discerning abnormal behaviors amidst the myriad of routine actions.

Traditionally, surveillance systems relied on rule-based methods and handcrafted features for activity analysis. Work in [2] introduces a human activity recognition method that combines temple posture matching and fuzzy rule reasoning and outperforms HMM-based approaches in recognition accuracy. The paper [16] presents an intelligent vision-based analyzer for real-time consumer video surveillance, employing innovative background subtraction and object classification techniques to detect unattended objects and enhance public safety in large cities. OBSERVER [3], an intelligent video surveillance system uses the Dynamic Oriented Graph (DOG) method for real-time unsupervised learning to detect and predict abnormal behaviors, outperforming the prior N-ary Trees classifier in experimental evaluations with synthetic data.

However, the advent of deep learning has revolutionized this landscape, offering a paradigm shift towards data-driven, end-to-end models that can automatically learn hierarchical representations from raw input data. Among these, Convolutional Neural Networks (CNNs) have demonstrated remarkable success in image analysis tasks, capturing spatial dependencies and patterns effectively. Long Short-Term Memory (LSTM) networks, on the other hand, excel in modeling temporal dependencies, making them well-suited for time-series data such as video sequences. This research embarks on a comprehensive exploration of deep learning architectures for suspicious activity detection, including Convolutional Long Short-Term Memory (ConvLSTM) networks.

The significance of this investigation lies in the nuanced interplay between spatial and temporal features within video data, necessitating a careful consideration of model architectures that can capture both aspects seamlessly. A range of architectures is explored, including CNN + BILSTM, CNN + LSTM, CONVLSTM, CONVLSTM + LSTM, CONVLSTM + BILSTM, CONVLSTM + CNN + LSTM, CNN + CONVLSTM + BILSTM, CONVLSTM + CNN + BILSTM, CNN + CONVLSTM, and CNN + CONVLSTM + LSTM, with the aim of identifying the most effective model for this challenging task.

The motivation for this study arises from the need for robust, adaptable, and accurate surveillance systems capable of preemptively identifying and responding to anomalous activities. As public safety concerns continue to escalate, the deployment of sophisticated models becomes imperative to enhance the capabilities of surveillance infrastructure.

By systematically comparing various deep learning architectures, this research endeavors to contribute valuable insights that can inform the design and implementation of surveillance systems geared towards real-world scenarios. The subsequent sections of this paper detail the related work in the domain, articulate the proposed research methodology, outline the experimental setup, present and analyze results, and conclude with implications and avenues for future research.

II. RELATED WORK

In modern security infrastructure, video surveillance has become integral for ensuring public safety and protecting critical assets. Traditional methods are being replaced by advanced systems capable of real-time monitoring and analysis. Deep learning approaches have emerged as key players in this domain, leveraging their ability to extract meaningful patterns from extensive visual data.

The work in [1] proposes a system to detect suspicious behavior in academic environments using CCTV footage. It employs a two-part framework: feature computation from video frames and classification of activities as suspicious or normal. The system aims to prevent crimes by alerting authorities in real-time, and while initially tailored for academic settings, it can be adapted for broader use in public or private spaces. The article in [10] presents an artificial intelligence-driven approach for predicting and detecting Robbery Behavior Potential (RBP) in indoor camera surveillance. Utilizing three detection modules—head cover, crowd, and loitering—their system aims to prevent robberies by analyzing real-world video images. The approach retains the YOLOV5 model and leverages a fuzzy inference machine to predict the robbery behavior. The paper [8] presents a method for video anomaly detection by combining CNNs for appearance encoding and ConvLSTMs for motion capture. Integrating these components with an Auto-Encoder yields the ConvLSTM-AE framework to the regular patterns of appearance and motion in normal events. The paper [4] introduces an attention-based convolutional LSTM algorithm to enhance human action recognition in videos by effectively capturing spatial and temporal features. By combining GoogleNet feature extraction with spatial transformer network attention and convolutional LSTM modeling, it achieves competitive performance on UCF-11, HMDB-51, and UCF-101 datasets while reducing training time through temporal coherence analysis.

The work in [13] proposes an unsupervised anomaly detection framework called ACLAE-DT for multivariate time series data in smart manufacturing. It employs a Convolutional Long Short-Term Memory (ConvLSTM) Autoencoder with Attention Mechanism, utilizing Dynamic Thresholding. The framework preprocesses data, constructs feature images to capture system statuses, and feeds them into an attention-based ConvLSTM autoencoder to encode temporal behavior. Work in [6] proposes a hybrid deep learning technique, 1D CNN-BiLSTM, for detecting anomalies in univariate time series data. This method combines one-dimensional convolutional neural networks (1D CNN) with bidirectional long short-term memory (BiLSTM) networks. The STC-1D CBiAM-RF method proposed in the [15] combines 1D CNN, BiLSTM, and attention mechanism for feature extraction and anomaly detection in UAV flight data, addressing issues of parameter selection, feature extraction, and noise mitigation. The work in [12] introduces a novel deep-learning model aiming to optimize human activity recognition, proposing a combination of 3D Convolutional Neural Networks (3DCNN) and Convolutional Long Short-Term Memory (ConvLSTM) layers.

The SABiAE[7] method combines self-attention-based encoder and bi-directional LSTM to capture global appearance and temporal features for anomaly detection in videos. The method in [9] introduces an attention-based variational autoencoder (A-VAE) architecture for detecting traffic anomalies in videos, leveraging 2D CNN and BILSTM layers with an attention mechanism.

Authors in [5] present a lightweight computational model for improved classification of violent and non-violent activities, crucial for addressing rising public violence. Leveraging deep learning, specifically a Convolutional Neural Network-based Bidirectional LSTM, the model achieves high accuracy.

Considering the popularity of the CNN, LSTM and CONVLSTM networks for suspicious activity detection, this paper seeks to conduct an extensive examination of various combinations of these deep learning architectures for detecting suspicious activities in video surveillance. The investigation encompasses a range of architectures, including CNN + BILSTM, CNN + LSTM, CONVLSTM, CONVLSTM + LSTM, CONVLSTM + BILSTM, CONVLSTM + CNN + LSTM, CNN + CONVLSTM + BILSTM, CONVLSTM + CNN + BILSTM, CNN + CONVLSTM and CNN + CONVLSTM + LSTM. Through rigorous training and testing on carefully curated datasets, the aim is to identify the most efficacious model for enhancing the accuracy of suspicious activity detection in complex and dynamic environments.

III. RESEARCH FLOW

This section describes the workflow of the conducted research. In the first step, a dataset for video classification was created using videos of required categories available from the Kaggle dataset.

The dataset comprises three classes, namely Fight, Normal activity, and Shooting. Additionally, the dataset was divided into two folders: 75% of the videos were allocated to the train folder, while the remaining 25% were placed in the test folder. Frames and features were then extracted from the video frames using the CNN model, as illustrated in Figure 1. Following this, various deep learning models were employed to classify the data, with training conducted on 75% of the training set. Subsequently, the trained models were utilized to predict outcomes on the unseen test set. This process is outlined in Figure 2.

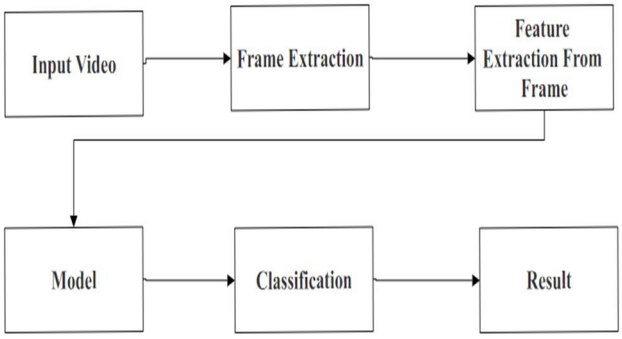


Fig. 1. Proposed methodology for suspicious activity detection

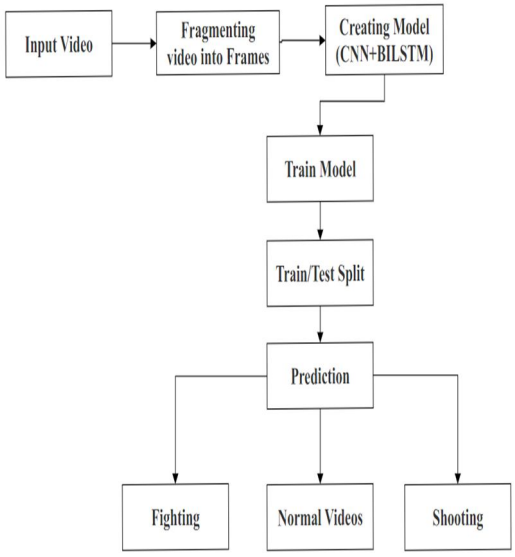


Fig. 2. Process of activity detection

1) *Dataset Description* : The dataset contains a total of 251 videos. There are 67 videos for fighting, 100 videos for normal activity and 84 videos for shooting. Every video has a different duration.
2) *Data Annotation* :Before initiating the labeling process, frames are extracted from the video and stored in one list. Python code reads the name of the folder for which it is going to perform labeling. Folder name is assigned as a label.
3) *Dataset Preparation :*Videos were collected from kaggle[1] and subsequently trimmed to include only relevant segments for analysis, ensuring consistency and relevance across the dataset. Two empty lists are initialized: 1) Frame List to store frames 2) Label List to store labels for frames of that video. Python OpenCV library is used to extract frames for each video. The interval *n*, after which the frame is stored is calculated using the formula:

$$n = \frac{Frame\ Count}{Sequence\ Length}$$

These frames are then stored in the Frame List. Name of the folder to which the video belongs to is assigned as a label for frames of that video using the Label list. Labels corresponding to particular videos are also stored in the list.

4) *Feature Extraction :*Multiple Conv2D layers are employed to capture spatial hierarchies and local patterns within each frame of the vIdeo sequence. Followed by each Conv2D layer, MaxPooling is applied to downsample the spatial dimensions.

IV. DATA MODELING

This section outlines the methods utilized for suspicious activity detection in this work. The mathematical foundations of the basic models are explored, followed by a detailed description of the combination of these models employed in the study.

A. *Mathematical foundations*

1) *CNN Model* : CNN architecture consists of several convolutional layers, each followed by either RELU or sigmoid or softmax activation functions to introduce non-linearity. The convolutional layers are responsible for extracting features from input images through the application of filters, which capture local patterns and structures. Max-pooling layers are incorporated after certain convolutional layers to reduce spatial dimensions. The output of the convolutional and pooling layers is flattened and fed into fully connected layers for classification.

The output of a convolutional layer can be represented as follows:

$$Z = f(W * X + b) \quad (1)$$

where X is an Input image, W is a Filter, b is Bias term, $*$ represents Convolution operation and f is an Activation function (such as ReLU).

2) *CONVLSTM Model*: ConvLSTM architecture extends the traditional LSTM cells by incorporating convolutional operations within the gates and cell states. This modification allows the network to learn spatial patterns and temporal dependencies simultaneously. Each ConvLSTM cell accepts input tensors representing both spatial and temporal information, which are convolved with learnable filters to extract features. The forget, input, and output gates of the ConvLSTM cell are updated using convolutional operations, enabling the model to selectively retain or discard information over time.

The operations for one time step in a ConvLSTM cell are given by:

$$f_t = \sigma(W_f * X_t + U_f * H_{t-1} + b_f) \quad (2)$$
$$i_t = \sigma(W_i * X_t + U_i * H_{t-1} + b_i) \quad (3)$$
$$CCS = tanh(W_c * X_t + U_c * H_{t-1} + b_c) \quad (4)$$
$$C_t = f_t \circledast C_{t-1} + i_t \odot CCS \quad (5)$$
$$o_t = \sigma(W_o * X_t + U_o * H_{t-1} + b_o) \quad (6)$$
$$H_t = o_t \odot tanh(C_t) \quad (7)$$

where X_t is Input at timestep t. W_f, W_i, W_c, and W_o Weight matrices for the forget gate, input gate, cell state, and output gate, respectively. U_f, U_i, U_c and U_o are recurrent weight matrices for the forget gate, input gate, cell state, and output gate, respectively. b_f, b_i, b_c, and b_o are the bias vectors for the forget gate, input gate, cell state, and output gate, respectively. f_t is the forget gate activation vector at time step t. It determines how much of the previous cell state C_{t-1} should be forgotten. It is the input gate activation vector at time step t. It determines which values from the input X_t should be updated into the cell state. CCS is the candidate cell state vector at time step t. It represents the new candidate values that could be added to the cell state C_t. H_{t-1} is the output of the previous time step t - 1. o_t is the output gate activation vector at time step t. It determines which values from the cell state C_t should be used to generate the output H_t. H_t is the output vector at time step t. It represents the information that will be passed to the next time step. σ is the Sigmoid activation function. $\odot$ is the Element-wise multiplication operation. $*$ is the Convolution operation. $\circledast$ is the Convolution operation followed by an element-wise multiplication. tanh is a hyperbolic tangent activation function.

3) *LSTM Model:* Long Short-Term Memory (LSTM) networks are proficient in capturing temporal dependencies in sequential data. By combining these two architectures, hierarchical representations learned by CNNs with the ability of LSTMs to model long-range dependencies, enabling more effective analysis of sequential data such as time series or video frames.

$$f_t = \sigma(W_f * X_t + U_f * H_{t-1} + b_f) \quad (8)$$
$$i_t = \sigma(W_i * X_t + U_i * H_{t-1} + b_i) \quad (9)$$
$$CCS = tanh(W_c * X_t + U_c * H_{t-1} + b_c) \quad (10)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot CCS \quad (11)$$
$$o_t = \sigma(W_o * X_t + U_o * H_{t-1} + b_o) \quad (12)$$

$$H_t = o_t \odot tanh(C_t) \quad (13)$$

4) *BILSTM Model :* Bidirectional Long Short-Term Memory (BiLSTM) networks have emerged as a powerful variant of LSTM networks, particularly in tasks involving sequential data analysis. By processing input sequences in both forward and backward directions, BiLSTMs enable the model to capture contextual information from past and future observations simultaneously.

Forward LSTM **:**

$$f_t = \sigma(W_f * X_t + U_f * H_{t-1} + b_f) \quad (14)$$
$$i_t = \sigma(W_i * X_t + U_i * H_{t-1} + b_i) \quad (15)$$
$$CCS = tanh(W_c * X_t + U_c * H_{t-1} + b_c) \quad (16)$$
$$C_t = f_t \odot C_{t-1} + i_t \odot CCS \quad (17)$$
$$o_t = \sigma(W_o * X_t + U_o * H_{t-1} + b_o) \quad (18)$$
$$H_t = o_t \odot tanh(C_t) \quad (19)$$

Backward LSTM :

$$f_t^b = \sigma(W_f^b * X_t + U_f^b * H_{t+1}^b + b_f^b) \quad (20)$$
$$i_t^b = \sigma(W_i^b * X_t + U_i^b * H_{t+1}^b + b_i^b) \quad (21)$$
$$CCS_b = tanh(W_c^b * X_t + U_c^b * H_{t+1}^b + b_c^b) \quad (22)$$
$$C_t^b = f_t^b \odot C_{t+1}^b + i_t^b \odot CCS_b \quad (23)$$
$$o_t^b = \sigma(W_o^b * X_t + U_o^b * H_{t+1}^b + b_o^b) \quad (24)$$
$$H_t^b = o_t^b \odot tanh(C_t^b) \quad (25)$$

Output :

$$H = [H_t; H_t^b] \quad (26)$$

H is the Final Output. It represents concatenation of output of forward and backward LSTM. ; Indicates concatenation. f_t^b is the forget gate activation vector at time step t in the backward direction. It determines how much of the next time step's cell state C_{t+1} should be forgotten. I_t^b is the input gate activation vector at time step t in the backward direction. It determines which values from the input X_t and the next time step's hidden state H_{t+1} should be updated into the cell state. CCS_b is the candidate cell state vector at time step t in the backward direction. It represents the new candidate values that could be added to the cell state C_t^b. C_t^b is the cell state vector at time step t in the backward direction. It represents the memory of the LSTM cell at time t in the backward direction. o_t^b is the output gate activation vector at time step t in the backward direction. It determines which values from the cell state C_t^b should be used to generate the output H_t^b. H_t^b is the output vector at time step t in the backward direction. It represents the information that will be passed to the previous time step.

B. Model Combinations

Following are the model combinations explored in this work.

1) *CNN + BILSTM (M1):* This combination involves using a CNN for feature extraction from input data followed by a bidirectional LSTM layer to capture temporal dependencies in both forward and backward direction.
2) *CNN + LSTM (M2):* This combination involves using a CNN for feature extraction from input data followed by an LSTM layer to capture temporal dependencies in the extracted features.
3) *CONVLSTM(M3):* This model allows the network to learn spatial and temporal dependencies simultaneously.
4) *ConvLSTM + LSTM (M4):* ConvLSTM layers will capture spatial-temporal patterns directly from the input data, while the LSTM layers further refine these representations by modeling longer-term dependencies in the sequence.
5) *ConvLSTM + BiLSTM(M5)*: This combination involves using ConvLSTM to capture spatial-temporal patterns and BiLSTM to model long-range dependencies in both forward and backward directions simultaneously.
6) *ConvLSTM + CNN + LSTM(M6)*: ConvLSTM is used first to capture spatial-temporal patterns, followed by a CNN layer for additional feature extraction and an LSTM layer for sequence modeling.
7) *CNN + ConvLSTM + BILSTM(M7):* CNN is used for spatial feature extraction, followed by ConvLSTM to capture spatial-temporal patterns, and BiLSTM further models long-range dependencies in both forward and backward directions simultaneously.

8) *ConvLSTM + CNN + BILSTM(M8) :*This combination starts with ConvLSTM for spatial-temporal pattern learning, followed by CNN for additional feature extraction, and BiLSTM for capturing long-range dependencies.
9) *CNN + CONV LSTM(M9):* This combination involves using a CNN for feature extraction from input data followed by CONVLSTM to capture spatial and temporal dependencies simultaneously.
10) *CNN + ConvLSTM + LSTM(M10) :*This architecture combines the feature extraction capabilities of CNNs with ConvLSTM, which can learn spatial-temporal patterns directly from input sequences, followed by an LSTM layer for further sequence modeling.

C. Evaluation Metrics

In the proposed study, the model's effectiveness is assessed utilizing a range of evaluation criteria encompassing Precision , F1-Score , Accuracy and the Recall.

Precision evaluates the ratio of correctly identified positive predictions among all positive predictions generated by the model.

$$Precision = \frac{T\square}{T\square + F\square} \quad (27)$$

F1-Score: It is the harmonic mean of precision and recall, providing a balance between these two metrics.

$$F1 - Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (28)$$

Accuracy is calculated as the ratio of correct predictions to the total number of predictions made by the model.

$$Accuracy = \frac{No.\ of\ Correct\ Predictions}{Total\ Predictions} \quad (29)$$

Recall assesses the percentage of correctly identified positive predictions out of all actual positive instances present in the dataset.. formula is used to calculate the Recall.

$$Recall = \frac{T\square}{T\square + F\square} \quad (30)$$

Where T□ represents True Positive
F□ represents False Positive
F□ represents False Negative

V. RESULT ANALYSIS

After preparing the dataset, it was divided into training and testing sets using a train-test split ratio of 75:25, respectively. This split ensured that 75% of the data was allocated for training the machine learning models, while the remaining 25% was reserved for evaluating their performance. Subsequently, experiments were conducted using various machine learning models to analyze the effectiveness of the approach. Each model was trained on the prepared dataset for 70 epochs. Additionally, to prevent overfitting during model training, the early stopping technique was employed.

This technique monitors the accuracy metric on the validation set during training and stops the training process if there is no improvement in accuracy for a specified number of epochs. It's noteworthy that each model converged at different rates, resulting in variations in the number of epochs required for training.

The performance metrics obtained from the evaluation of different models are presented in *Table 1*. Each model's precision, recall, F1-Score, and accuracy are listed, providing insights into their effectiveness in classifying human activities from video**s.**

First column of Figure 3 and Figure 4 displays the training and validation accuracy curves over epochs for models M1 to M10 in that order.

It illustrates the model's learning progress, with the training accuracy gradually increasing and the validation accuracy indicating how well the model generalizes to unseen data. Second column of Figure 3 and Figure 4 presents the training and validation loss curves for models M1 to M10 in that order. . It depicts the model's convergence during training, with the training loss decreasing over epochs. The validation loss curve helps identify if the model is overfitting, as increasing validation loss suggests poor generalization.

Table 1
Model's precision, recall, F1-Score and Accuracy

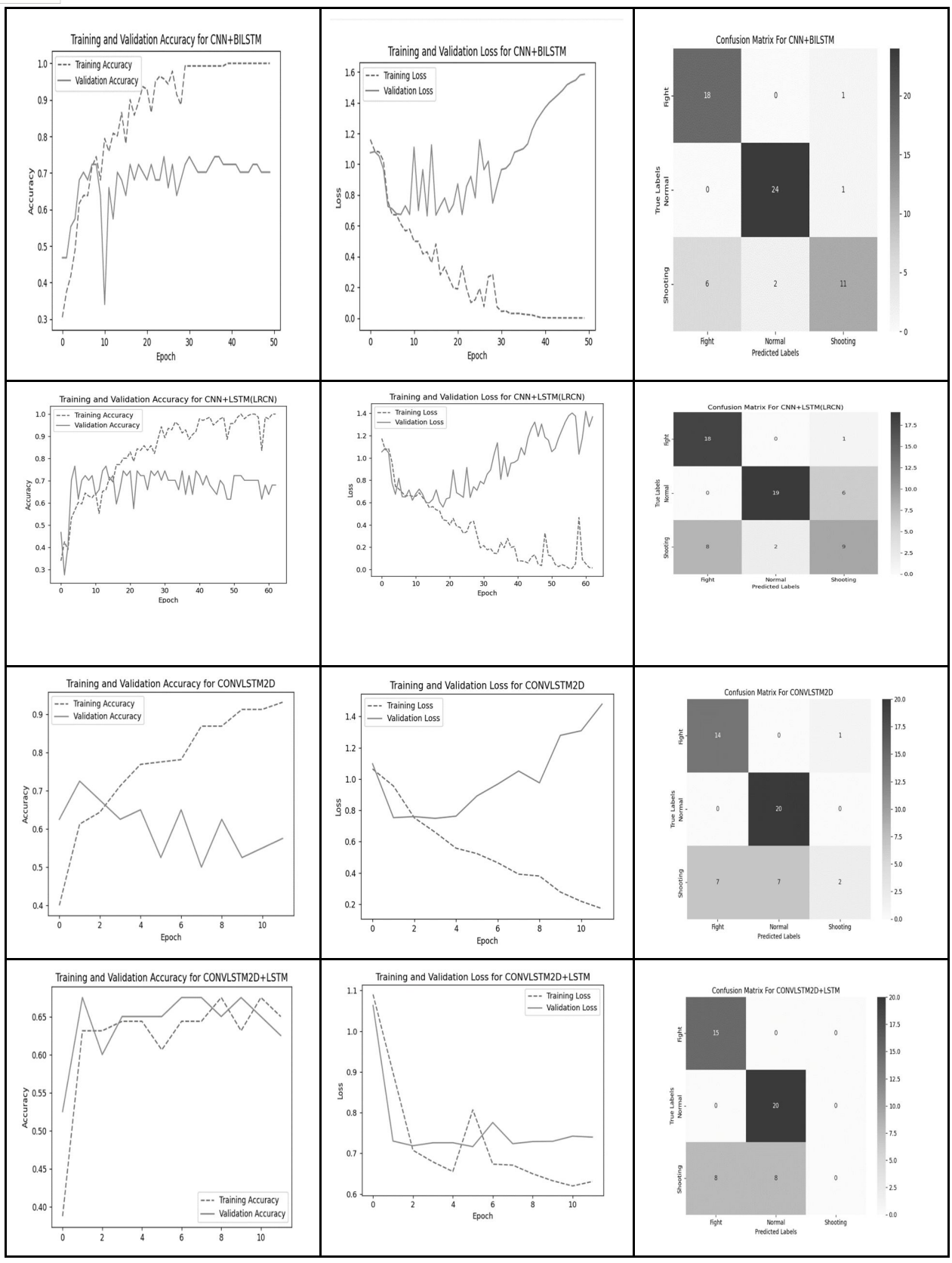
Model	Precision	Recall	F1-Score	Accuracy
M1	84.77%	84.13%	83.33%	84.13%
M2	73.75%	73.02%	72.41%	73.02%
M3	69.57%	70.59**%**	62.86%	70.59%
M4	47.19%	68.63%	55.90%	68.63%
M5	62.06%	66.67%	61.77%	66.67%
M6	61.63%	65.08%	63.30%	65.08%
M7	68.50%	60.32%	61.59%	60.32%
M8	56.42%	57.14%	56.77%	57.14%
M9	54.17%	55.56%	54.47%	55.56%
M10	17.28%	26.98%	19.07%	26.98%

Third column of Figure 3 and Figure 4 visualizes the confusion matrix for models M1 to M10 in that order. Offering insights into the model's classification performance across different activity classes. The diagonal elements represent correctly classified instances, while off-diagonal elements indicate misclassifications.

From the results, it is evident that Model M1, utilizing a combination of Convolutional Neural Network (CNN) and Bidirectional Long Short-Term Memory (BILSTM), achieved the highest precision, recall, F1-Score, and accuracy among all models. This indicates that the incorporation of both CNN and BILSTM contributes significantly to the model's performance in accurately classifying human activities from video data.

Models M2 to M10 exhibit varying levels of performance, with some achieving moderate precision, recall, and accuracy scores, while others demonstrate poorer performance. Models M3 to M5, which utilize ConvLSTM or combinations thereof, generally exhibit moderate performance. Convolutional architectures inherently excel in spatial feature extraction, yet their performance in capturing temporal dynamics might be limited compared to models incorporating recurrent layers. Models with multiple architectural components (M6 to M10) do not consistently outperform simpler architectures. This suggests that while incorporating additional layers and structures may enhance the model's capacity to learn complex patterns, it also introduces additional complexity and potential performance degradation.

In the context of suspicious activity detection, the trade-off between precision and recall holds significant implications. High precision ensures that when the model raises an alert, it is trustworthy and accurate, crucial for minimizing false alarms and maintaining operational efficiency. Conversely, high recall guarantees that the model does not miss any genuine suspicious activities, thus reducing the risk of false negatives and enhancing overall security.In the investigation, a notable trade-off between precision and recall was observed across the evaluated deep learning models. Models such as M4 (CONVLSTM + LSTM) and M7 (CNN + CONVLSTM + BILSTM) demonstrated high precision, suggesting accurate identification of suspicious activities. However, these models may exhibit lower recall, indicating a potential risk of missing some true positive instances. Conversely, models with higher recall, such as M5 (ConvLSTM + BiLSTM) or M6 (ConvLSTM + CNN + LSTM), tended to prioritize sensitivity over precision, thus increasing the likelihood of capturing actual suspicious activities but potentially introducing more false positives.However, achieving high precision often necessitates setting stringent thresholds or criteria, which may inadvertently lead to the model missing some true positive instances, consequently lowering


Training and Validation Accuracy for CNN+BILSTM
Training Accuracy
Validation Accuracy
Accuracy
Epoch
Training and Validation Loss for CNN+BILSTM
Training Loss
Validation Loss
Loss
Epoch
Confusion Matrix For CNN+BILSTM
Fight
Normal
Shooting
True Labels
Predicted Labels
18 0 1
0 24 1
6 2 11
Training and Validation Accuracy for CNN+LSTM(LRCN)
Training and Validation Loss for CNN+LSTM(LRCN)
Confusion Matrix For CNN+LSTM(LRCN)
18 0 1
0 19 6
8 2 9
Training and Validation Accuracy for CONVLSTM2D
Training and Validation Loss for CONVLSTM2D
Confusion Matrix For CONVLSTM2D
14 0 1
0 20 0
7 7 2
Training and Validation Accuracy for CONVLSTM2D+LSTM
Training and Validation Loss for CONVLSTM2D+LSTM
Confusion Matrix For CONVLSTM2D+LSTM
15 0 0
0 20 0
8 8 0

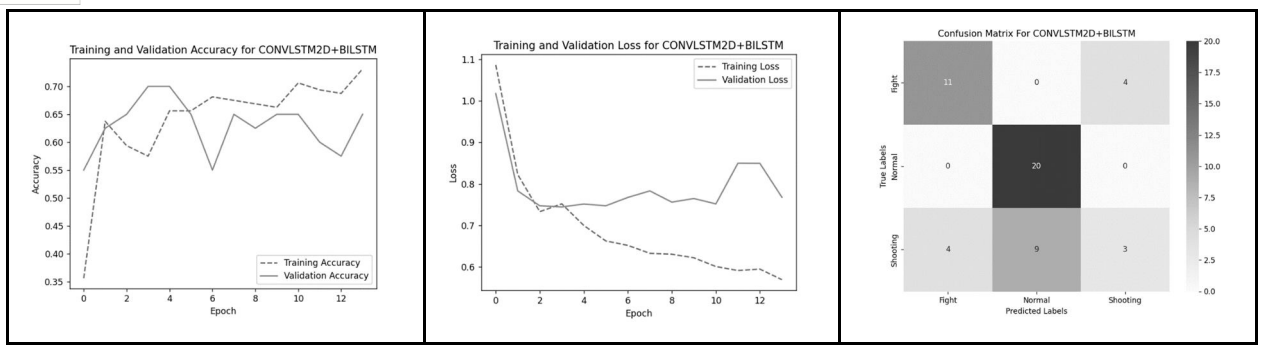
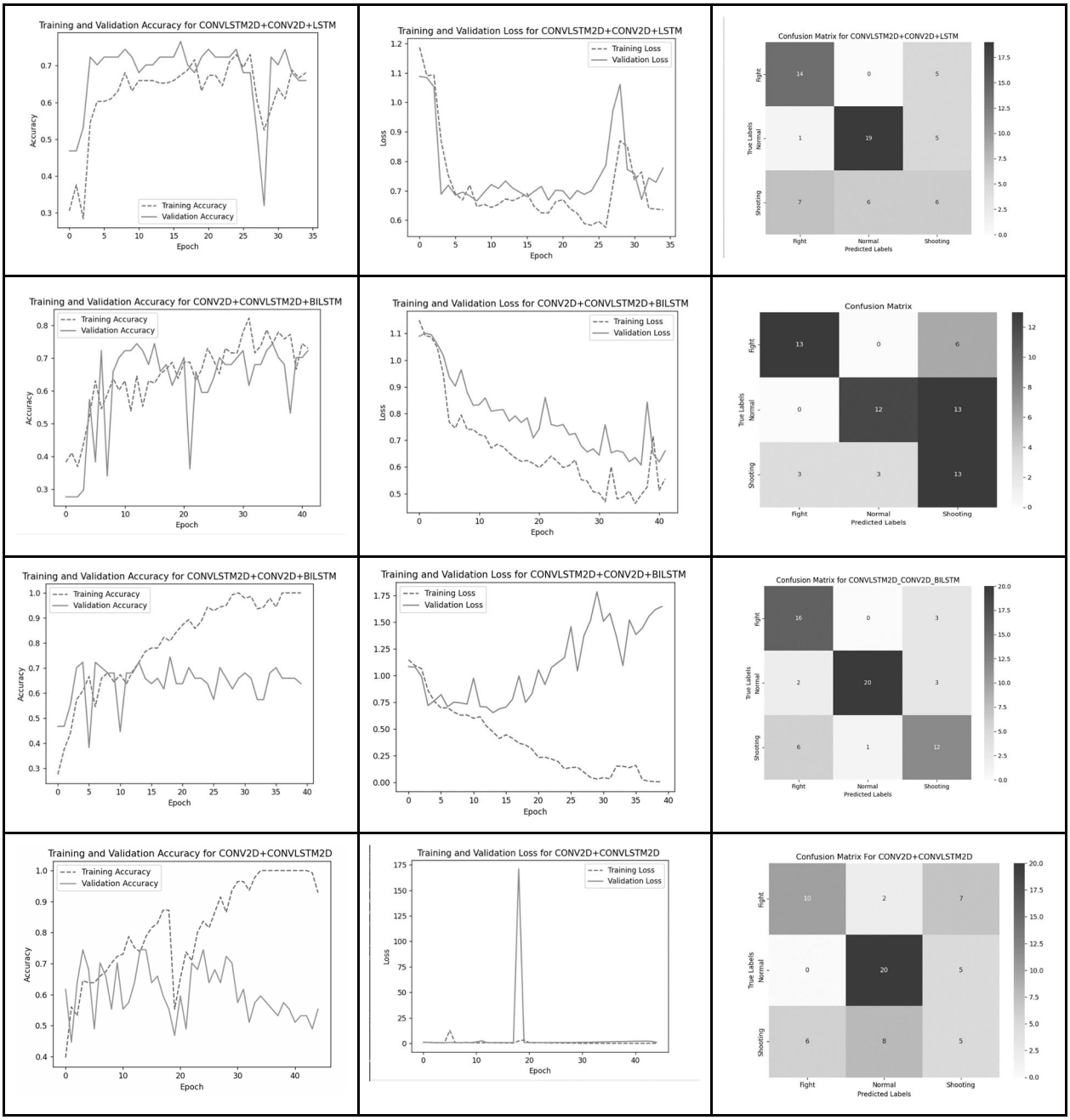


Fig:3 Training and Validation Accuracy, Validation Loss and Confusion Matrix for M1 to M5



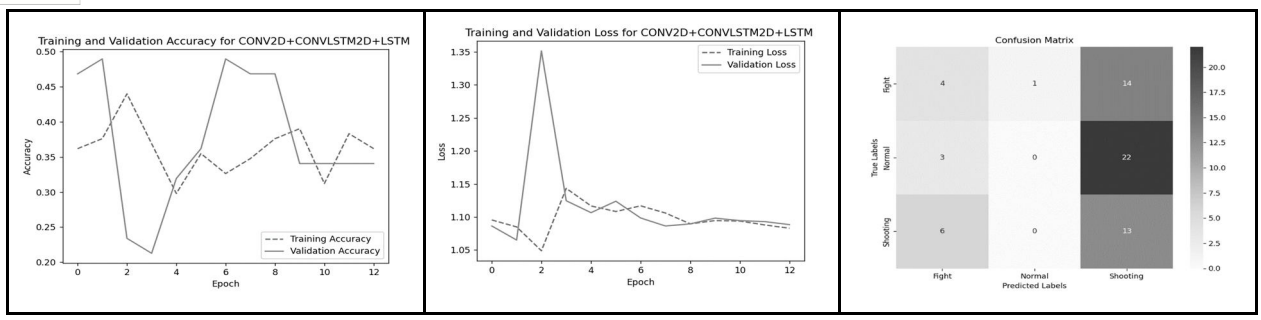


Fig.4: Training and Validation Accuracy, Validation Loss and Confusion Matrix for M6 to M10

recall. Conversely, prioritizing high recall may result in more false positives, diminishing precision and potentially inundating security personnel with spurious alerts. Therefore, striking a balance between precision and recall is paramount in the development of effective surveillance systems.

VI. CONCLUSION

This study has provided valuable insights into the effectiveness and trade-offs associated with various deep learning architectures for suspicious activity detection in video surveillance. Models combining CNN with recurrent layers, such as LSTM or BiLSTM, demonstrated superior performance, achieving a balanced approach between precision and recall. Additionally, our research emphasized the balance between model complexity and effectiveness, with simpler architectures showing competitive performance. These findings underscore the importance of architectural design and empirical evaluation in achieving optimal neural network performance.

Moving forward, future research should focus on refining model architectures and optimizing parameters. Additionally, exploring the application of these models on real-time videos will be crucial for validating their effectiveness in dynamic surveillance environments.

VII. CONFLICT OF INTEREST

The authors declare that they have no competing interests.

REFERENCES

[1] C. V. Amrutha, C. Jyotsna and J. Amudha, "Deep Learning Approach for Suspicious Activity Detection from Surveillance Video," 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA), Bangalore, India, 2020, pp. 335-339,doi:10.1109/ICIMIA48430.2020.9074920.

[2] Chang, Jyh-Yeong, Jia-Jye Shyu, and Chien-Wen Cho. "Fuzzy rule inference based human activity recognition." In 2009 IEEE Control Applications,(CCA) & Intelligent Control,(ISIC), pp. 211-215. IEEE, 2009.

[3] Duque, Duarte, Henrique Santos, and Paulo Cortez. "Prediction of abnormal behaviors for intelligent video surveillance systems." In 2007 IEEE Symposium on Computational Intelligence and Data Mining, pp. 362-367. IEEE, 2007.

[4] Ge, Hongwei, Zehang Yan, Wenhao Yu, and Liang Sun. "An attention mechanism based convolutional LSTM network for video action recognition." Multimedia Tools and Applications 78 (2019): 20533-20556.

[5] Halder, Rohit, and Rajdeep Chatterjee. "CNN-BiLSTM model for violence detection in smart surveillance." SN Computer science 1, no. 4 (2020): 201.

[6] Ibrahim, Moamen, Khaled M. Badran, and Ahmed Esmat Hussien. "Artificial intelligence-based approach for Univariate time-series Anomaly detection using Hybrid CNN-BiLSTM Model." In 2022 13th International Conference on Electrical Engineering (ICEENG), pp. 129-133. IEEE, 2022.

[7] J. Zhang, X. Qi and G. Ji, "Self Attention based Bi-directional Long Short-Term Memory Auto Encoder for Video Anomaly Detection," 2021 Ninth International Conference on Advanced Cloud and Big Data (CBD), Xi'an, China, 2022, pp. 107-112, doi: 10.1109/CBD54617.2021.00027.

[8] Luo, Weixin, Wen Liu, and Shenghua Gao. "Remembering history with convolutional lstm for anomaly detection." In 2017 IEEE International conference on multimedia and expo (ICME), pp. 439-444. IEEE, 2017.

[9] N. Aslam and M. H. Kolekar, "A-VAE: Attention based Variational Autoencoder for Traffic Video Anomaly Detection," 2023 IEEE 8th International Conference for Convergence in Technology (I2CT), Lonavla, India, 2023, pp. 1-7, doi: 10.1109/I2CT57861.2023.10126296.

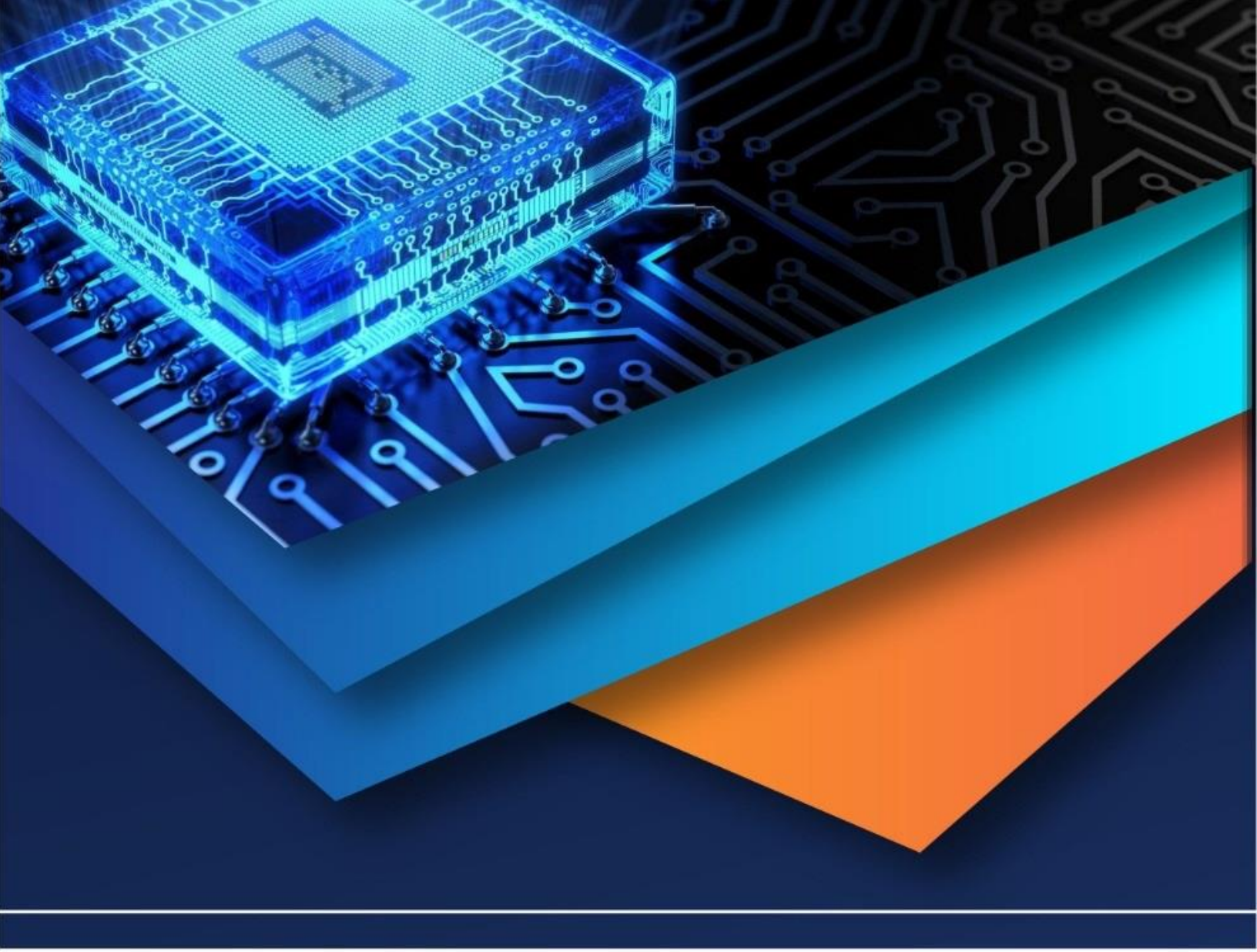
[10] Pouyan, Shima, Mostafa Charmi, Ali Azarpeyvand, and Hossein Hassanpoor. "Propounding first artificial intelligence approach for predicting robbery behavior potential in an indoor security camera." IEEE Access (2023).

[11] Sreenu, G. S. D. M. A., and Saleem Durai. "Intelligent video surveillance: a review through deep learning techniques for crowd analysis." Journal of Big Data 6, no. 1 (2019): 1-27.

[12] Stepanyan, Ivan V., and Safa A. Hameed. "An improved neurogenetic model for recognition of 3D kinetic data of humans extracted from the Vicon Robot system." Baghdad Science Journal 20, no. 6 (Suppl.) (2023): 2608-2608.

[13] Tayeh, Tareq, Sulaiman Aburakhia, Ryan Myers, and Abdallah Shami. "An attention-based ConvLSTM autoencoder with dynamic thresholding for unsupervised anomaly detection in multivariate time series." Machine Learning and Knowledge Extraction 4, no. 2 (2022): 350-370.

[14] Tripathi, Rajesh Kumar, Anand Singh Jalal, and Subhash Chand Agrawal. "Suspicious human activity recognition: a review." Artificial Intelligence Review 50 (2018): 283-339.
[15] Yang, Lei, Shaobo Li, Caichao Zhu, Ansi Zhang, and Zihao Liao. "Spatio-temporal correlation-based multiple regression for anomaly detection and recovery of unmanned aerial vehicle flight data." Advanced Engineering Informatics 60 (2024): 102440.
[16] Zin, Thi Thi, Pyke Tin, Hiromitsu Hama, and Takashi Toriu. "Unattended object intelligent analyzer for consumer video surveillance." IEEE Transactions on Consumer Electronics 57, no. 2 (2011): 549-557.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY