Enhancing Security and Efficiency with Smart Surveillance Using Machine Learning

Deepjyoti Purkayastha Saswasto Basak Rejwan Azam Mondal Saikat Gharami Pushpita Roy Prianka Dey Tanmoy Ghosh
Narula Institute of Technology, Agarpara, India
{Kayastha@deepjyoti7147.in}
{saswasto16, rejwan.azam.mondal321, Saikatgharami2, roypushpi, Priankadey2011, tanmoy.g.331}@gmail.com

***Abstract*—The proposed system offers an innovative solution to address crowd control, security, and worker tracking challenges using Artificial Intelligence and Machine Learning. Specifically designed to support the Indian Railways and relevant authorities, the system aims to enhance operational efficiency and security. Its core functionality focuses on advanced crowd management through ML algorithms, instantly alerting authorities when predefined thresholds are exceeded. The system also utilises AI to detect potentially violent actions and suspicious activities, providing real-time insights through a user-friendly dashboard for 24/7 monitoring. With versatility in resource allocation and scalability in densely populated areas, the system's cloud-based infrastructure ensures ease of integration, cost efficiency, and adaptability. However, success depends on high-quality video feeds, stable internet connectivity, user-friendliness, and seamless integration with existing CCTV systems. In conclusion, this AI and ML-based system offers a comprehensive, real-time solution for security enhancement and efficiency across public and private sectors.**

***Index Terms*—Real Time Crowd Analysis, Cloud server infrastructure, Densely Populated Areas, Public Safety.**

I. Introduction

Improving both security and operational efficiency is now a central objective across diverse industries, particularly in settings such as transportation hubs where maintaining safety amid significant foot traffic is critical. Smart surveillance systems [1], driven by machine learning algorithms, have emerged as a transformative force in modernizing conventional security protocols. This integration provides a proactive method for surveillance, allowing for the real-time detection of potential threats while streamlining resource distribution and operational processes. For instance, in railway stations, using machine learning in security cameras brings big changes. These new systems can learn how people normally behave, telling the difference between regular actions and things that seem strange or out of place. With tools like predictive analytics and recognizing patterns, these smart cameras can find potential threats better. They send alerts to security staff quickly, so they can deal with any security problems fast.

Nowadays, in railway station security, the need to keep passengers safe and maintain smooth operations faces challenges in quickly spotting potential threats or unusual behavior in large and bustling environments. Regular surveillance systems can struggle to give instant feedback [1], causing delays in detecting threats and managing resources. While past research [2] has looked into different deep learning algorithms like SSD and Faster R-CNN for detecting weapons, managing crime centrally, and capturing incidents in real-time, there's still a need for a unified approach that uses advanced machine learning techniques to overcome the shortcomings and gaps seen in current methods.

The primary challenges include the tradeoff between speed and accuracy in weapon detection, the lack of explicit algorithms mentioned in centralised crime management systems, and the need for more precise human action detection in crowded environments. Furthermore, existing methodologies may not fully leverage posture tracking for comprehensive human activity analysis. There is also a notable gap in addressing the limitations of current research, highlighting the need for an improved, adaptive, and efficient surveillance system in railway stations.

Furthermore, surveillance using machine learning keeps getting better over time. These systems learn from past information, making their methods smarter to fit the changing atmosphere of railway stations. As an example, the HOG descriptor [3], [4] looks at the shape of objects in the surveillance system. This method outperforms other methods because it takes into account both the magnitude and direction of changes to identify crucial details. Our module will detect and confirm anomalous activities [5] in real-time, using Machine Learning and an event-driven approach to send anomalous data to protective services and police units enabling immediate action while conserving resources by uploading the data of the cctv footage for a certain period of time.

In essence, the convergence of machine learning and smart surveillance [6] technologies in railway station security marks a significant leap forward. This synergy not only fortifies security protocols but also streamlines operations, fostering a safer and more efficient environment for commuters and staff alike. For instance, in a busy railway station, smart surveillance using machine learning changes security. Cameras equipped with AI algorithms swiftly detect anomalies, differentiating between routine activity and potential threats like unattended luggage. These systems continuously learn from station patterns, reducing false alarms and enhancing threat recognition.

This proactive approach ensures a safer environment while optimising operations, allowing security personnel to focus on genuine risks. Ultimately, passengers benefit from heightened security measures and smoother travel experiences.

Our proposed method uses computer vision, CNN model and posture tracking for identifying anomalies in railway station. Our method uses Convolution neural Network (CNN) which revolves around analysing visual data captured by surveillance cameras. CNNs can detect unusual activities within the monitored area, signalling security teams to potential security risks. CNNs [2] process visual data in real-time, enabling immediate analysis and response to security concerns. Computer vision, as discussed in [7], enhances surveillance by employing machine learning to improve efficiency. Posture tracking involves a deep learning technique in which the algorithm follows the motion of an object.

The challenge is to improve railway station security through an integrated surveillance system that utilizes advanced machine learning algorithms for posture tracking, human action detection, and activity captioning. This system aims to surpass current methods by offering a more precise and effective way of identifying potential threats, differentiating between normal and unusual activities, and providing real-time alerts for immediate security action. The goal is to develop a comprehensive solution that maximizes resource utilization, reduces false alarms, and promotes a safer and more efficient travel experience for all rail station commuters.

The paper is organised as follows. Section II discusses about methodologies applied in the earlier literature for providing surveillance at railway stations. Section III addresses the problem statement, whereas the Section IV describes the overall solution methodology of our problem. Section V shows the implementation of our proposed methodology to the UCF50 dataset to provide the result. Section VI concludes our work.

II. Literature Survey

Multiple research works are studied and their findings are summarised in this section. The authors of [2] propose a method to find weapons in images. They use deep learning tools like Single Shot Detector (SSD) and FASTER R-CNN for this task. The image goes through different steps, including passing through a Convolutional Neural Network (CNN) backbone, which is often a pre-trained network like VGG16 in the FASTER R-CNN. This network has two parts: a RPN (Region Proposal Network) and Fast-RCNN. The RPN makes suggestions about regions using the feature maps, which are boxes that might contain objects. The RPN works by going through the feature map with a small window (usually 3×3), deciding if there's an object in each spot, and then changing the box's position if needed.

The Faster R-CNN, by using the RPN to suggest areas of interest, makes the process of detecting objects much faster compared to previous methods like Fast R-CNN. Previously, techniques such as Fast R-CNN relied on external methods, such as selective search, to identify areas of interest. In contrast, SSD or Single Shot Detector is designed to rapidly detect objects in a single pass through the data, unlike two-step methods like Faster R-CNN.

In [6], the authors describe the process of extracting information from CCTV footage and organizing it within a centralized system for crime management. According to the study, each video feed must be divided and analyzed separately.

The paper suggests employing an encoder-decoder architecture, a type of neural network architecture commonly used for sequence-to-sequence tasks like machine translation and text summarization. It consists of two main parts: an encoder and a decoder. The encoder processes an input sequence to generate a vector representation that captures the sequence's meaning and context. It can be implemented using various neural networks like recurrent neural networks (RNNs) or transformers. The decoder uses the vector representation from the encoder to produce an output sequence, generating it one element at a time with the help of the previous elements of the output sequence. The decoder is also implemented using different neural networks. Ensuring the safety and protection of passengers is paramount.

The study aims to predict crime patterns and identify hotspots in cities or states. The author employed various algorithms, such as Random Forest and Naive Bayes, and conducted a comparative analysis of their results. This involved applying these algorithms to a dataset or problem and evaluating their performance metrics to determine which one is more effective in achieving the desired outcome. Surveillance systems often struggle to quickly identify potential threats or unusual activities in large and busy areas. However, integrating machine learning into surveillance cameras revolutionizes the monitoring process. The shift from storing video to storing incident captions for real-time crime detection [8] demonstrates practical approaches to overcoming challenges related to time and space. Using machine learning algorithms to predict crime patterns and identify hotspots in a city or state [9] reflects a proactive approach to crime prevention.

Railway stations can substantially improve security by installing smart surveillance systems driven by machine learning algorithms. These cameras can independently identify suspicious behaviors, like unattended luggage or people loitering in restricted zones. For example, a machine learning-based camera system can swiftly distinguish between normal commuter activity and erratic behavior, sending real-time alerts for immediate security action. Additionally, these systems continually learn and adjust to the station's patterns and dynamics, improving their accuracy in detecting threats over time.

More accurate threat identification leads to the optimization of the operational efficiency and reduces the generation of false alarms. By reducing the number of false alarms and focusing human attention where it is most necessary, these systems allow security personnel to allocate their resources more effectively, ultimately leading to a safer travel environment for all commuters.

III. Problem Statement

Our problem statement is as follows.

Input: The input to our problem is the real time videos captured by CCTVs incorporated in the railway stations.

Output: The output of our problem is the detection of the criminal activities from the real time videos captured by CCTVs and sends the alert to the nearby security personnel.

Objective: The objective of our problem is to develop intelligent surveillance system that can analyze and interpret video data in real-time for enabling the tasks to identify the criminal activities using object recognition, anomaly detection, and behavioral analysis of the crowd. It also sends the alert to the nearby security personnel if any suspicious activity is detected.

IV. Detailed Methodology

Our methodology for enhancing surveillance system in railway stations revolves around the integration of advanced machine learning algorithms, particularly focusing on posture tracking for human action detection and activity captioning. The aim is to address existing limitations and improve the precision and efficiency of surveillance systems in identifying potential threats and unusual activities. The overview of our methodology is explained in the flowchart below.

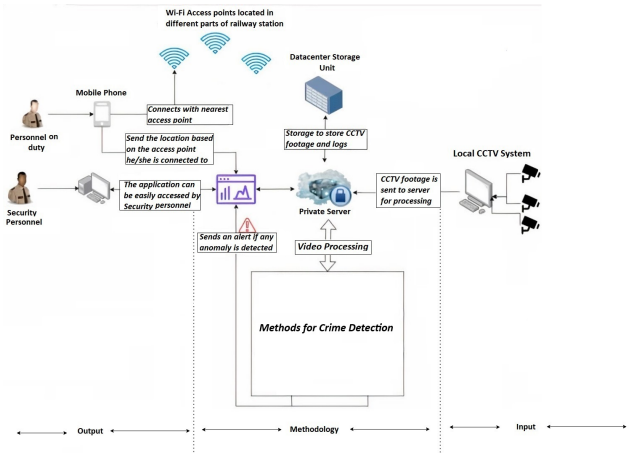


Fig. 1: Applied Methodology

Figure 1 shows the underneath steps of our solution methodology. The input to our surveillance system is captured through local CCTV systems which is integrated with PC or laptop. Then the footage from the CCTV system is uploaded to the allocated private server and also stored in the DSU (Datacenter Storage Unit) for one week for the further use. The storage of this huge amount of data for long duration of time is costly enough. Hence, the DSU will be reset after each week to make it cost effective. The input images / videos are processed further for performing crime detection by detecting any anomalous activity in railway station. AlphaPose and YOLOv3 (You Only Look Once version 3) [10] can be used for detecting crime in the crowd.

Algorithm 1: identifyPotentialThreats($comInfo$)

```
for each frame in comInfo do
    threatScore = 0;
    /* extract relevant information from the frame */
    objects ← frame.objects /* output from YOLOv3 */
    humanposes ← frame.humanposes /* output from alphapose */
    /* Analyse object detections */
    for each object in objects do
        if object.type == "weapon" then
            threatScore += WEIGHT_WEAPON_DETECTION;
        end
    end
    /* Analyse human poses */
    for each pose in humanPoses do
        if pose.action == "aggresive" then
            threatScore+ = WEIGHT_AGGRESIVE_SCORE;
        end
    end
    /* consider contextual information (e.g location) */
    location ← frame.location
    if location == restricted_area then
        threatScore+ = WEIGHT_RESTRICTED_AREA;
    end
    /* evaluate the overall threat score for  */
    if threatScore ≥ THREAT_THRESHOLD then
        generatealert(frame, threshold)
    end
end
```

AlphaPose is an advanced deep-learning algorithm used for estimating human poses from images or videos. It accurately identifies key points [11] in the human body and is capable of detecting multiple individuals [12] in a scene simultaneously. YOLOv3 (You Only Look Once version 3) [10], is a state-of-the-art object detection algorithm. Unlike traditional object detection methods, YOLOv3 [13] considers the entire image at once, enabling fast and accurate detection of multiple objects in a single pass. Given in figure 1 the methods of crime detection uses these algorithms to determine if any anomalous activity has taken place. In this paper, we propose a method to detect the anomalous activity in the crowd. We also propose another method that sends alerts to the officials after detecting any potential threat in the crowd in railway stations. Our proposed method not only tracks the posture of the person in the crowd, it also detects the weapons carried or used in the crowd that may cause the potential threats.

Algorithm 2: generateAlert($frame, threatScore$)

```
/* Implement Alert generation logic here */
/* Thia could involve sending notifications,
   activating alarms, or other actions */
print("Potential threat detected in frame:", frame.frameNumber, "with threat score:", threatScore)

/* example weights for different factirs
   affecting threat score */
WEIGHT_WEAPON_DETECTION = 0.7
WEIGHT_AGGRESIVE_POSE = 0.5
WEIGHT_RESTRICTED_AREA = 0.8

/* Threshold to trigger an alert */
THREAT_THRESHOLD = 1.5

/* Example data structure for combined
   information */
comInfo = [
  {frame : 1, objects : [...], humanPoses : [...], location : "public_area"},
  {frame : 2, objects : [...], humanPoses : [...], location : "restricted_area"},
  /* more frames */
]

/* Call the function to identify the potential
   threats */
identifyPotentialThreats(comInfo)
```

Algorithm 1 shows the steps to detect the posture movements of the person in the crowd for identifying as the unusual; movements in the crowd. Algorithm 2 generates alerts to the officials after detecting any potential threat. After detecting any unusual movements in the crowd by Algorithm 1, Algorithm 2 gets triggered and generates the alerts. Our proposed methodology not only generates the alert, but it also detects the weapon being used or carried in the crowd or not.

In Algorithm 1, we identify potential threat, it iterates through each frame in $comInfo$, which presumably contains combined information about objects, human poses, and location. For each frame, it initializes $threatScore$ to 0. It then analyzes object detections and human poses in the frame, incrementing $threatScore$ based on predefined weights for detecting weapons and aggressive human poses. It also considers contextual information such as location, incrementing $threatScore$ if the location is identified as a $restrictedarea$. After calculating the $threatScore$, if it exceeds or equals the $THREAT\_THRESHOLD$, it generates an alert using the $generateAlert()$ function, passing the frame and the calculated $threatScore$. We need to ensure that the constants like $WEIGHT\_WEAPON\_DETECTION$, $WEIGHT\_AGGRESSIVE\_SCORE$, $WEIGHT\_RESTRICTED\_AREA$, and $THREAT\_THRESHOLD$ are defined with appropriate values. This algorithm generates a basic threat detection system that analyzes frames of information, such as objects, human poses, and location, to identify potential threats.

Algorithm 2 is responsible for generating alerts when a potential threat is detected in a frame. It prints out a message indicating the potential threat detected along with the frame number and the calculated threat score. Instead, it's left to be implemented externally based on the requirements of the system. The example weights and threshold provided are used by the threat detection logic in Algorithm 1 to determine whether a potential threat is significant enough to trigger an alert.

(a) Taichi

(b) Horse Riding

Fig. 2: Sample dataset for object detection in YOLOv3

In this study, a comprehensive model employing machine learning algorithms was developed to bolster security measures in the Indian Railways. A subset comprising 30 percentage of images from a diverse range of classes in the surveillance dataset is meticulously selected to evaluate the model's accuracy.

A. Model Structure

We use ConvLSTM as the model structure in our work.The ConvLSTM model structure integrates convolutional and LSTM layers to process sequential data with spatial and temporal dependencies. It combines convolutional operations with LSTM units, enabling the model to capture both spatial and temporal patterns simultaneously. Input data undergoes feature extraction through convolutional layers, while LSTM cells handle memory and state management to capture long-term dependencies. Here is the steps of input and output and the modules used in the feature extraction of the images given in the above figure 3.

V. Results

This study gives an appraisal of recent object detection and anomalous movement detection methodologies, development of smart surveillance of CCTV in crowded areas like railway stations and the challenges faced during the creation

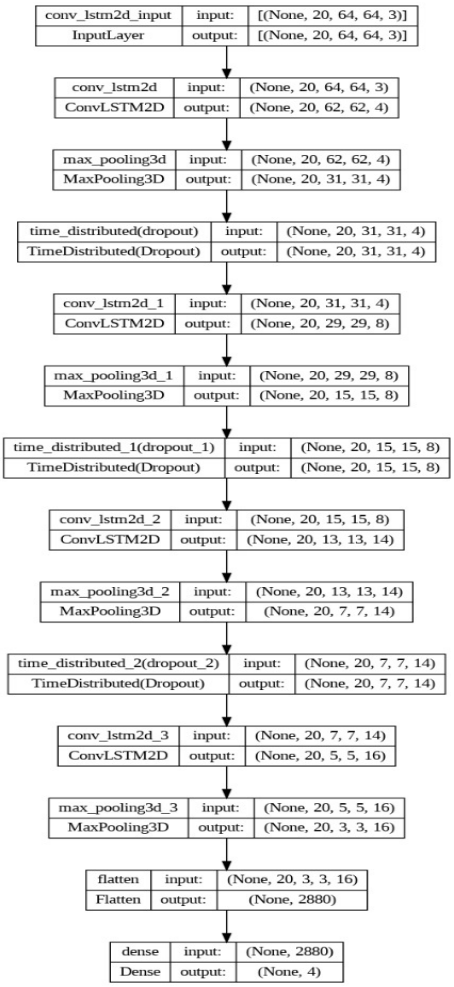


Fig. 3: ConvLSTM model structure

of the project. A comparative analysis [14] is provided to both obtainable databases and connected benchmarks. We have highlighted the shortcomings of the older projects, and evaluated a way to deal with the limitations.

A. Database

Here, we use UCF YouTube Action (UCF50) dataset for demonstrating our proposed methodology. The UCF YouTube Action (UCF50) dataset consists of unconstrained videos taken from the web and is a very challenging dataset, but it has only 11 action categories, all of which are very distinct actions. The UCF50 dataset, also contains videos downloaded from YouTube and has 50 action categories. The recently released HMDB51 dataset has 51 action categories, but after excluding facial actions like smile, laugh, chew, and talk, which are not articulated actions, it has 47 categories compared to 50 categories in UCF50. Most of the current methods would fail to detect an action/activity in datasets like UCF50 and HMDB51 where the videos are taken from web as given in figure 4.

These videos contain random camera motion, poor lighting conditions, clutter, as well as changes in scale, appearance, and view points, and occasionally no focus on the action of

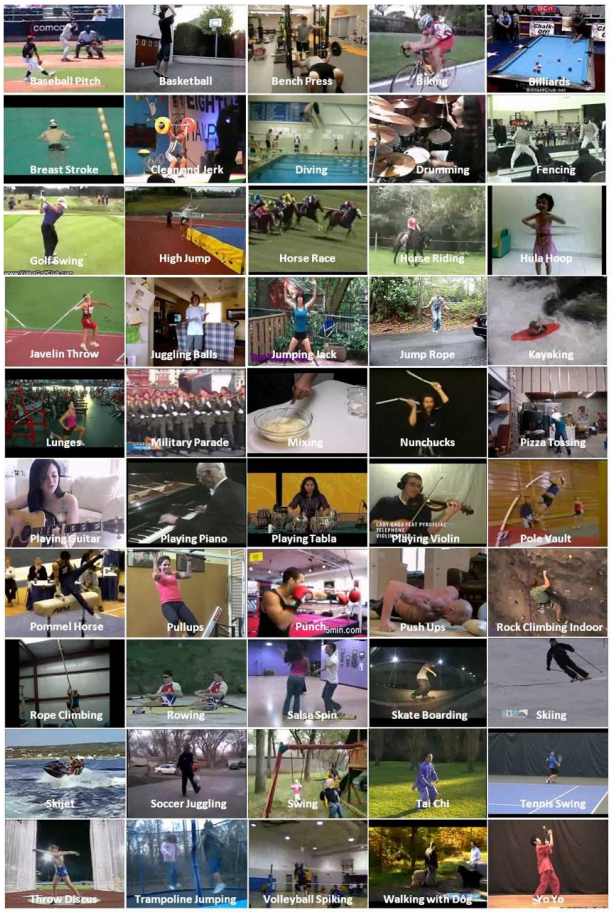


Fig. 4: UCF50 dataset

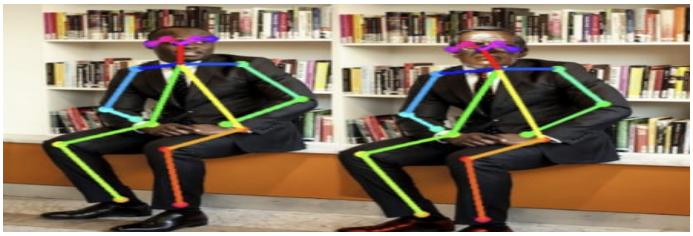
interest. Table I shows the list of action datasets. In this paper we study the effect of large datasets on performance, and propose a framework that can address issues with real life action recognition datasets(UCF50).

Dataset	Number of Actions	Camera Motion	Background
KTH	6	slight motion	static
Weizmann	10	not present	static
IXMAS	14	not present	static
UCF Sports	9	present	dynamic
HOHA	12	present	dynamic
UCF11	11	present	dynamic
UCF50	50	present	dynamic
HMDB51	51(47)	present	dynamic

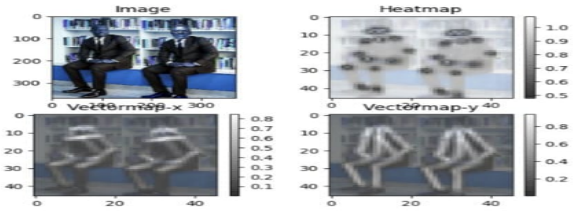
TABLE I: Action Datasets

The module is trained using the 80% to detect the movement and actions done by a person. Figure 5b shows the example of the output after the process of feature extraction. An example of output of our proposed methodology given in the figure 5a. Similarly, the sample dataset and its corresponding output is shown in figure 6.

We have trained our model using 80% of the images available in the input dataset and then we test the model using 20%



(a) Joint Tracking



(b) Feature Extraction

Fig. 5: Results for detection of movement in Alphapose

(a) Taichi

(b) Horse Riding

Fig. 6: Detected results in YOLOv3

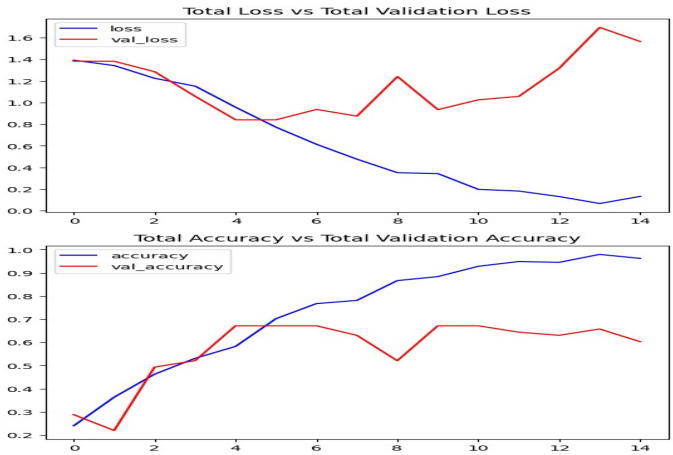


Fig. 7: Performance graph of ConvLSTM Approach

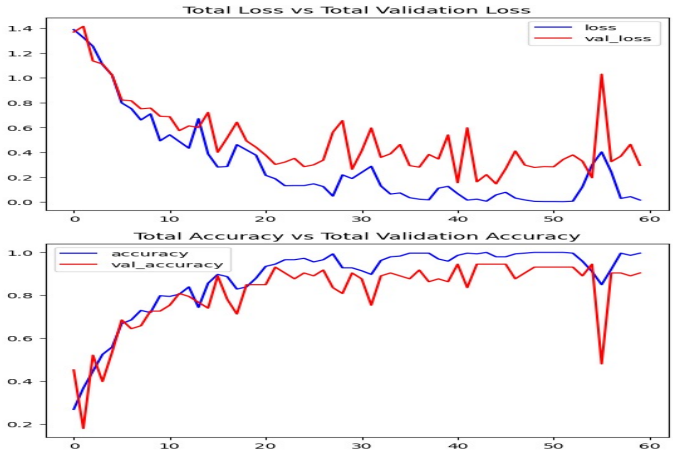


Fig. 8: Performance graph of LRCN Approach

of the data. The performance of our methodology is shown in Figure 7. Figure 7 shows the total accuracy and the total loss that we obtained after the demonstration of our method to the input dataset. Here, the accuracy is computed by the total number of correct prediction divided by the total number of predictions. Similarly, the loss can be computed by the total number of mis-predictions divided by the total number of attempts. Figure 8 shows the performance of our method using LRCN model.

In Figure 7, it's evident that the initial graph demonstrates a decrease in validation loss followed by an increase after a specific number of epochs, while the total loss decreases with an increase in epochs. The second graph illustrates a rise in total accuracy and validation accuracy for a certain number of epochs before stabilizing as the epochs increase. In Figure 8, the initial graph depicts both total loss and total validation loss, which exhibit a slight instability before decreasing with an increase in epochs. The second graph demonstrates a similar pattern with both total accuracy and total validation accuracy showing slight instability before increasing with additional epochs. It's noteworthy that an epoch represents a full cycle of training a neural network with all the available training data. It's also observed that the loss graph exhibits an inverse relationship with the accuracy graph and vice versa in both figures.

VI. Conclusion

The overarching objective of this project is to design, implement, and deploy an Artificial Intelligence (AI) and Machine Learning (ML) based smart surveillance system with a focus on enhancing crowd management, security measures, and resource allocation. Our model approximately counts the number of people in public spaces and trigger notifications to authorities when predefined thresholds are surpassed. The aim is to facilitate timely responses for effective crowd management, particularly in crowded areas such as railway stations. It also identifies the potentially violent human actions, detect suspicious activities, and monitor sudden movements. It also enhances the security by generating real-time alerts to the security personnel. We demonstrate our proposed methodology on UCF50 dataset to show the performance of our methodology.

REFERENCES

[1] K. Kardas and N. K. Cicekli, "Svas: surveillance video analysis system," *Expert Systems with Applications*, vol. 89, pp. 343–361, 2017.

[2] H. Jain, A. Vikram, A. Kashyap, A. Jain *et al.*, "Weapon detection using artificial intelligence and deep learning for security applications," in *2020 International Conference on Electronics and Sustainable Communication Systems (ICESC)*. IEEE, 2020, pp. 193–198.

[3] K. S. do Prado, "Face recognition: Understanding lbph algorithm," *Towards Data Science*, 2017.

[4] M. Tyagi, "Hog (histogram of oriented gradients): An overview," *Towards Data Science*, vol. 4, 2021.

[5] T. Sultana and K. A. Wahid, "Iot-guard: Event-driven fog-based video surveillance system for real-time security management," *IEEE Access*, vol. 7, pp. 134 881–134 894, 2019.

[6] S. T. Naurin, A. Saha, K. Akter, and S. Ahmed, "A proposed architecture to suspect and trace criminal activity using surveillance cameras," in *2020 IEEE Region 10 Symposium (TENSYMP)*. IEEE, 2020, pp. 431–435.

[7] R. Dayana, M. Suganya, P. Balaji, A. M. Thahir, and P. Arunkumar, "Smart surveillance system using deep learning," *International Journal of Computer Science and Mobile Computing*, vol. 8, no. 4, pp. 27–35, 2019.

[8] S. O. Sapna Patwa, Nagesh Nayak and S. Roychowdhury, "Real time crime detection by captioning video surveillance using deep learning," in *International Journal of Recent Technology and Engineering*, vol. 10, 2022, pp. d367–d376.

[9] Y.-L. Lin, T.-Y. Chen, and L.-C. Yu, "Using machine learning to assist crime prevention," in *2017 6th IIAI international congress on advanced applied informatics (IIAI-AAI)*. IEEE, 2017, pp. 1029–1030.

[10] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[11] I. Ozhiganov, "Convolution neural network vs cascade classifiers for object detection," in *DZone*, May 2017.

[12] P. Viola and M. Jones, "Computer vision and pattern recognition, 2001," in *IEEE Computer Society Conference*, vol. 1, 2001.

[13] A. Hampapur, L. Brown, J. Connell, N. Haas, M. Lu, H. Merkl, S. Pankanti, A. Senior, C.-F. Shu, and Y. Tian, "S3-r1: The ibm smart surveillance system-release 1," in *Proceedings of the 2004 ACM SIGMM workshop on Effective telepresence*, 2004, pp. 59–62.

[14] I. Taufik, M. Musthopa, A. R. Atmadja, M. A. Ramdhani, Y. A. Gerhana, and N. Ismail, "Comparison of principal component analysis algorithm and local binary pattern for feature extraction on face recognition system," in *MATEC Web of Conferences*, vol. 197. EDP Sciences, 2018, p. 03001.