Smart surveillance using deep learning

Amsaveni Avinashiappan, Harshavarthan Thiagarajan, Harshwarth Coimbatore Mahesh, Rohith Suresh
Department of Electronics and Communication Engineering, Kumaraguru College of Technology, Coimbatore, India

Article Info

Article history:

Received Sep 10, 2022
Revised Jan 21, 2023
Accepted Feb 18, 2023

Keywords:

Convolutional neural network
Keras
Surveillance
Telegram API
Tensorflow

ABSTRACT

Smart surveillance systems play an important role in security today. The goal of security systems is to protect users against fires, car accidents, and other forms of violence. The primary function of these systems is to offer security in residential areas. In today's culture, protecting our homes is critical. Surveillance, which ranges from private houses to large corporations, is critical in making us feel safe. There are numerous machine learning algorithms for home security systems; however, the deep learning convolutional neural network (CNN) technique outperforms the others. The Keras, Tensorflow, Cv2, Glob, Imutils, and PIL libraries are used to train and assess the detection method. A web application is used to provide a user-friendly environment. The flask web framework is used to construct it. The flash-mail, requests, and telegram application programming interface (API) apps are used in the alerting approach. The surveillance system tracks abnormal activities and uses machine learning to determine if the scenario is normal or not based on the acquired image. After capturing the image, it is compared with the existing dataset, and the model is trained using normal events. When there is an anomalous event, the model produces an output from which the mean distance for each frame is calculated.

Corresponding Author:

Amsaveni Avinashiappan
Department of Electronics and Communication Engineering, Kumaraguru College of Technology
Chinnavedampatti, Coimbatore, Tamil Nadu, India
Email: amsaveni.a.ece@kct.ac.in

1. INTRODUCTION

Today, safety and security are growing in popularity because of their numerous benefits, and despite the developments, the safety of one's residence should not be disregarded. As a result, many changes are being made in the field of safety systems to ensure that proper safety is given to the user's property and users. Only if the system provides security and monitoring that monitors the number of alerts as well as dwelling protection against things like fire, road accidents, and attack, it will be considered ideal. When a client is not present at home for whatever reason, it might be likely that visitors to their site, or a web application that the user may access, will leave them offline, and users in the telegram app will receive an automatic message [1]–[5]. The major purpose of smart surveillance is to deliver security to users' homes or companies. If there is any automatic alerting, the camera is installed at the entry of the door to watch any scenario of a fire accident, road accident, or other violations taking place close to the house or organization [6]–[8]. The main motive of our research work is to build a system that monitors movement and responds quickly to abnormal events by showing an error that can be transmitted through telegram. The required systems are a camera, raspberry pi, and a good internet connection. This idea of surveillance could change a big way in rural areas and this surveillance could be monitored from any part of the world [9]–[12].

Several researchers have attempted to construct effective smart surveillance systems that can distinguish any human action using various methodologies; much work has already been done on smart surveillance systems. A tiny PIR sensor and a very less power alerting system monitor the real-time video using an embedded chip and programming methods. The Raspberry Pi serves as the main processing unit for the system, and it can be used for real-time video transmission in various applications [13]–[17]. The fundamental purpose of the article is to make a network that would allow photographs to be delivered and received from a base station to camera nodes. The purpose is to create a wireless security system. When a person enters a room, you'll get this simple alarm sensor. When a criminal is caught, the picture is matched to the image specified on the site, and the alert is triggered. Human bodies make heat through infrared, which is not perceived by the eye. However, an e-sensor can detect it [18]–[21].

Approximately 1.35 million individuals face the difficulty of being delayed due to traffic every year. It affects between 20 and 50 million individuals, according to data. People pay the price for their lives because of such road accidents. Such situations have arisen because of a lack of cooperation among the parties concerned. Furthermore, if the needed concepts and methods are not adequately practiced, the graph will rise. Excessive speed, inebriated driving, inattentive driving, poor infrastructure, improper cars, and exceeding limits are only a few of the hazards [22]–[25].

2. METHOD

Deep learning algorithms can be used to analyze and interpret visual data from surveillance cameras, allowing for real-time detection and identification of objects, people, and events. One example of a smart surveillance system using deep learning is object detection in video footage. A deep learning model can be trained on a dataset of labeled images to identify and classify objects in a video stream. The model can then be used in real-time to detect and track objects of interest, such as people or vehicles, and trigger alerts or actions based on specific events or behaviors. The proposed framework consists of three important stages, preprocessing, training and testing, and web development as displayed in Figure 1.

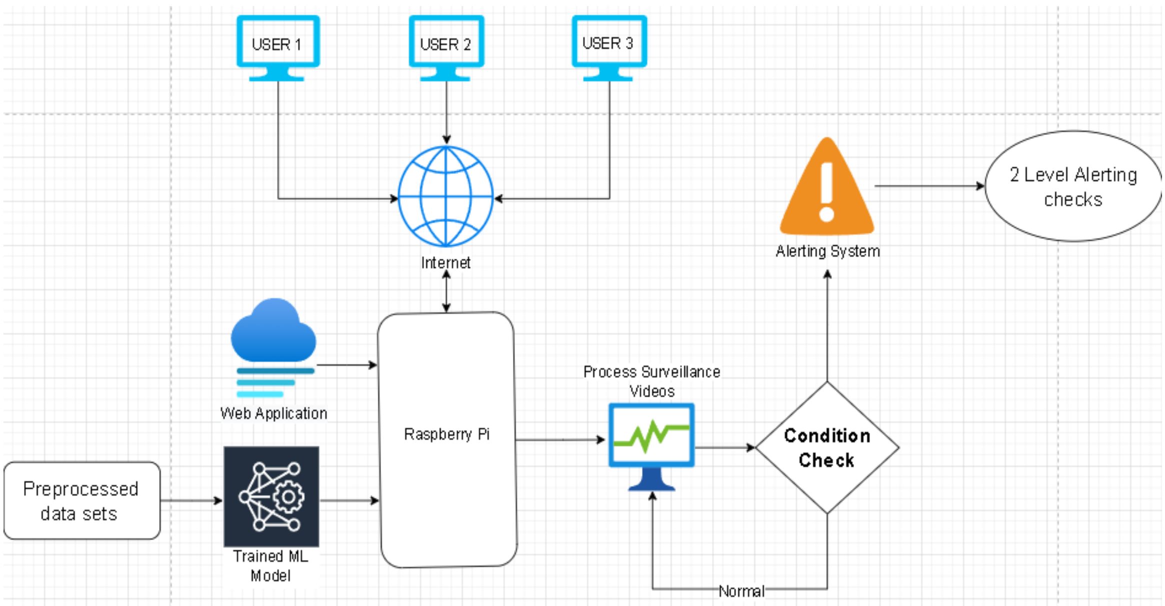


Figure 1. Block schematic of the proposed work

2.1. Preprocessing

The camera' video footage or the dataset footage obtained is used to train the model. Pre-processing denotes the separation of video content into frames, each of which is of a different size and has a unique grey scale. This technique handles and keeps these changes consistent, making training, and evaluating the model easier.

2.2. CNN and RNN

A convolutional neural network (CNN) is a type of artificial neural network that is often utilized in deep learning to examine visual images. CNNs are established using a shared-weight architecture design that is implemented using convolution kernels or filters that are a way path along its input features to yield translation-equivariant outputs known as feature maps. Because of the contradictory strategy of down sampling, they applied to the input, prior CNN are not unchanging. The photos and videos are sensed and then processed. Image classification, image subdivision, medical image observation, natural language processing, the brain-computer interfaces, and financial time series are some of the applications. RNNs are a certain type of neural network that remembers everything over time. Because of its capacity to remember past inputs, it is only useful for time series prediction. This is referred to as long short name memory (LSTM). To increase the effective pixel neighborhood, recurrent neural networks (RNN) are combined with convolutional layers.

2.3. Training and testing

The Keras deep learning library aids in the fast and easy development of neural network models. Keras models may be created in two ways: sequentially or functionally. The sequential application programming interface (API) builds the model layer by layer, like a linear stack of layers. Building a network appears to be a simple task. However, the sequential API has a few constraints that prevent us from creating models that share layers or have numerous inputs or outputs. The functional API is an alternate method for creating a neural network. It gives more flexibility to create a sophisticated network with various inputs or outputs and a model that can share layers.

These pre-processed frames are now used to train the neural network model. When an anomalous event is provided to the model, it produces an output from which the mean distance for each frame is determined. When the mean distance between the current frame and the last frame exceeds the threshold value, an abnormal event takes place.

2.4. Web development

Web development can enable the integration of smart surveillance systems with other technologies and platforms, such as mobile applications and cloud-based services. This can help to improve the scalability and flexibility of the system and allow for more efficient and effective monitoring and analysis of surveillance data. The website is built using Python and cascading style sheets (CSS). Users are given credentials (username and password), and they may use them to access the web page. There are two options for monitoring: providing the IP address of the camera-integrated phone or uploading the video clip that may be viewed.

3. RESULTS AND DISCUSSION

3.1. Functional model results

The functional model is based on RNN that is used for multiple inputs and multiple outputs for the trained model. The experimental results of the functional model are tabulated in Table 1. As shown in Figure 2, both epoch vs accuracy and epoch vs loss are important metrics to monitor the functional model of a neural network. The epoch vs accuracy measures how well the model can predict the correct output for a given input and is typically visualized using a learning curve. During the training process, the accuracy of the model typically improves with each epoch, as the network adjusts its weights and biases to better fit the training data as depicted in Figure 2(a). Epoch vs loss, on the other hand, measures how well the model can minimize its training loss function, which is a measure of how different the predicted output is from the true output. The loss value is decreased with epoch to an extent and remains constant as illustrated in Figure 2(b).

Table 1. Functional model data

Epoch	Accuracy in (%)	Loss
1	49.60	0.1647
2	50.40	0.1361
3	52.50	0.1261
4	53.50	0.1209
5	54.10	0.1176
6	54.60	0.1150
7	54.78	0.1140
8	54.95	0.1130
9	55.00	0.1130
10	55.34	0.1110

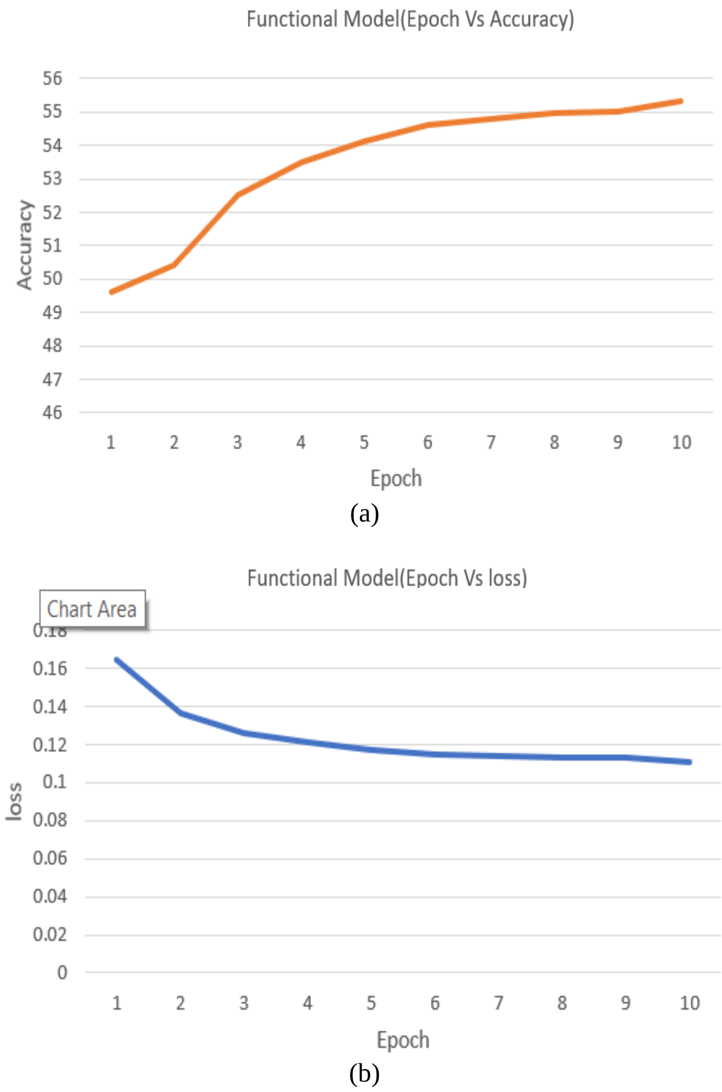


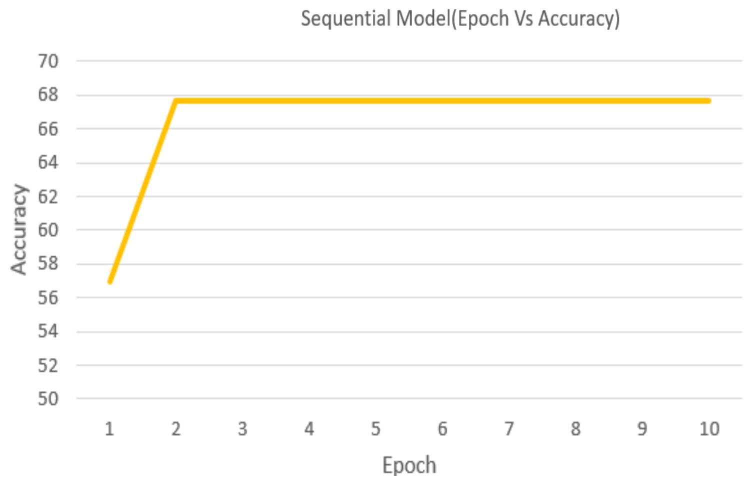
Figure 2. Functional model (a) accuracy vs. epoch and (b) loss vs. epoch

3.2. Sequential model results

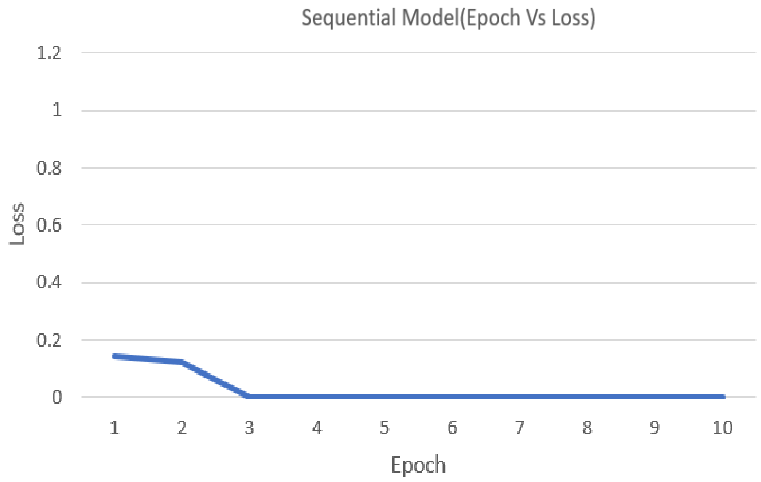
In the sequential model, layers are added one by one, in a linear manner, to form the neural network. Table 2 shows the experimental results of the sequential model. As shown in Figure 3, the performance of the model is evaluated using metrics such as accuracy or loss on a validation set. From Figure 3(a), it is inferred that if the epoch is increased, then the loss is decreased to an extent and remains constant, and the accuracy value is increased as shown in Figure 3(b) and remains constant after a certain iteration.

Table 2. Sequential model data

Epoch	Accuracy in (%)	Loss
1	57	0.1446
2	67.7	0.125
3	67.7	7.7506e-04
4	67.7	7.7506e-04
5	67.7	7.7506e-04
6	67.7	3.1514e-04
7	67.7	3.1514e-04
8	67.7	2.7058e-04
9	67.7	2.4251e-04
10	67.7	2.0977e-04



(a)



(b)

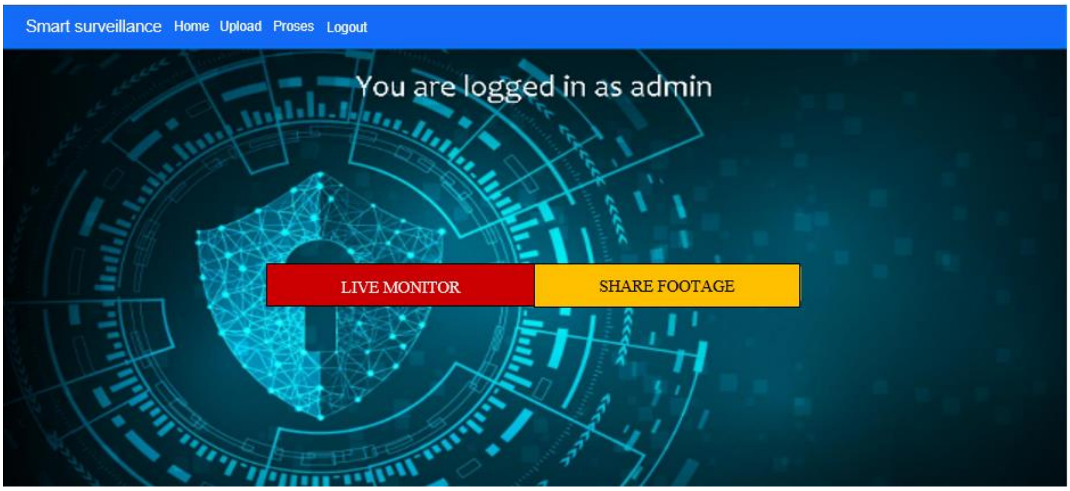
Figure 3. Sequential model (a) accuracy vs. epoch and (b) loss vs. epoch

3.3. Web application results

Figure 4 shows the web application for surveillance activity which could provide valuable insights and real-time updates on the status of surveillance cameras. Figure 4(a) demonstrates how the trained machine learning (ML) model asks us to enter the IP address for the login page of the security camera. The IP address of the phone with the camera or the footage video can be uploaded to this page to do surveillance. The IP address of the camera-integrated phone is entered in Figure 4(b).

Figure 5 shows the video footage entering page which is an important one for surveillance analysis, as it allows users to upload video footage quickly and easily for analysis and monitoring. Once the video footage is received, pre-processing steps are performed to ensure consistency and quality. This may involve tasks such as resizing, format conversion, frame extraction, denoising, and stabilization to enhance the overall quality of the footage. By using deep learning algorithms to analyze the video, the system can detect abnormal activity and alert security personnel in real time, helping to prevent potential threats or security breaches. The video footage entering page can be integrated with alerting systems to generate real-time notifications or alarms based on specific events or triggers identified within the video footage.

Abnormal activities like fire accidents, road accidents, and violations have been detected and the user has been alerted through the telegram app as shown in Figure 6. In the case of fire accidents, deep learning models can be trained to recognize specific patterns of heat and smoke that are associated with fires as depicted in Figure 6(a). Similarly, for road accidents, models can be trained to recognize damaged vehicles and injured people as shown in Figure 6(b). For violations, models can be trained to recognize patterns of behavior that are associated with illegal or unsafe activities as in Figure 6(c).



(a)

(b)

Figure 4. Web application (a) login page and (b) surveillance activity

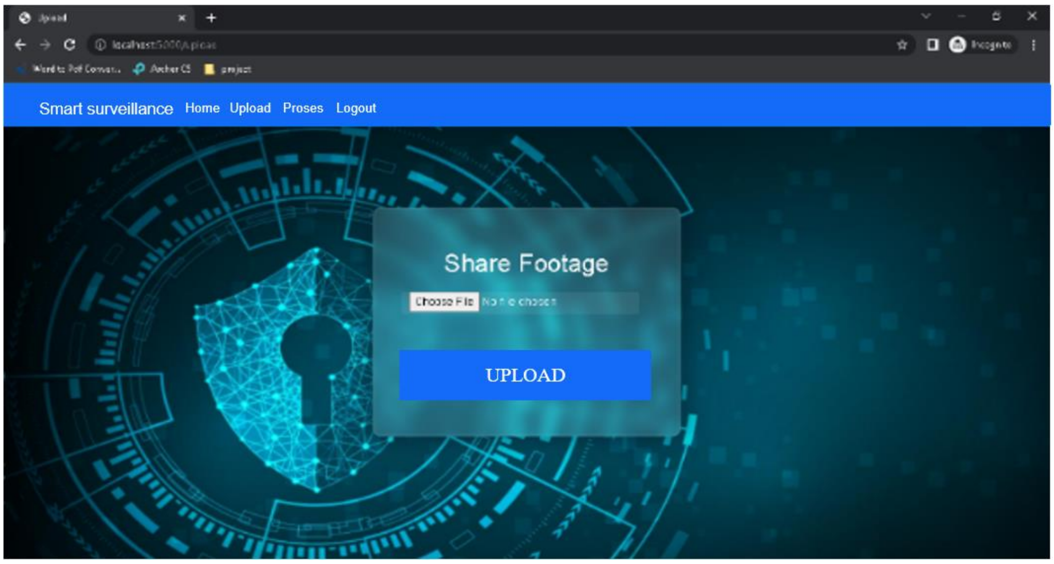
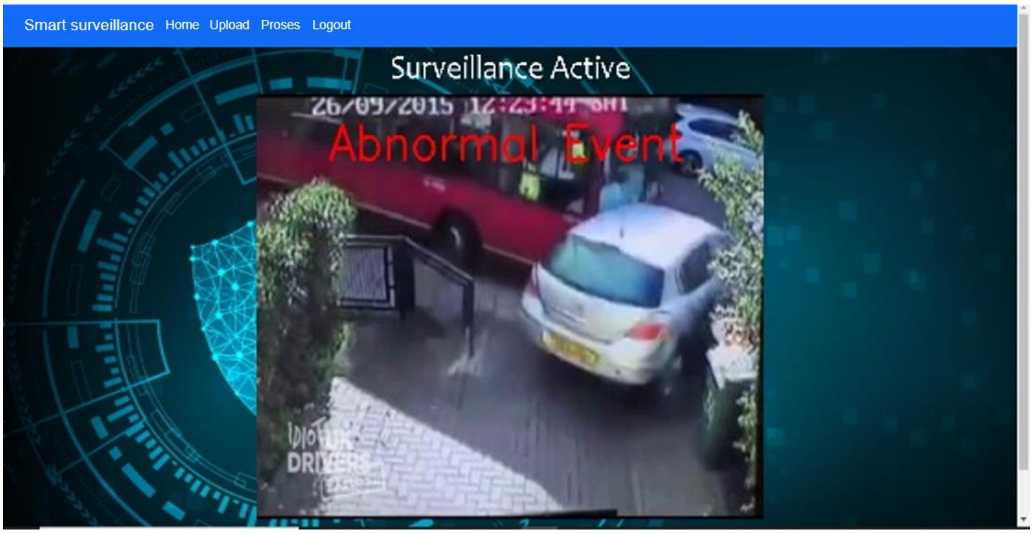
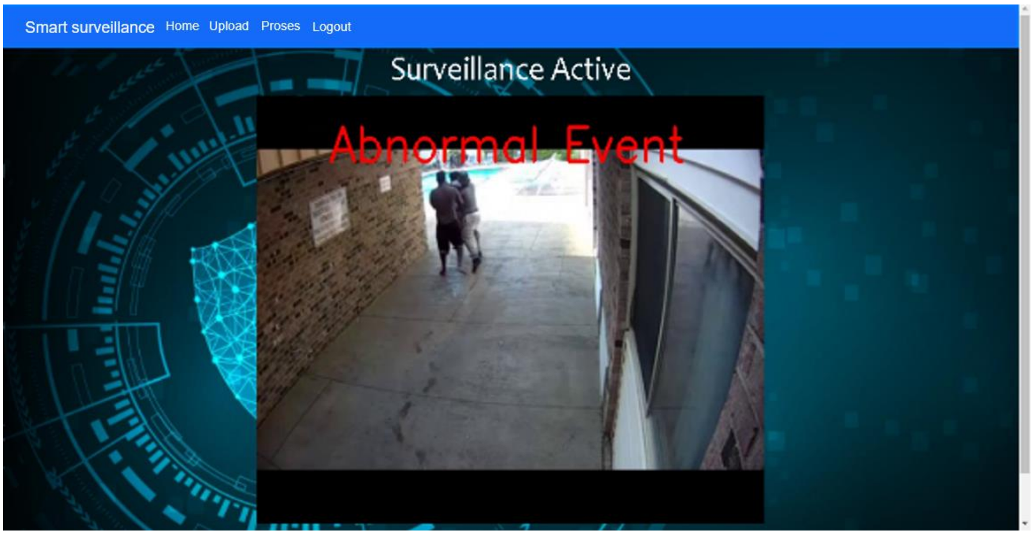


Figure 5. Video footage entering page

(a)



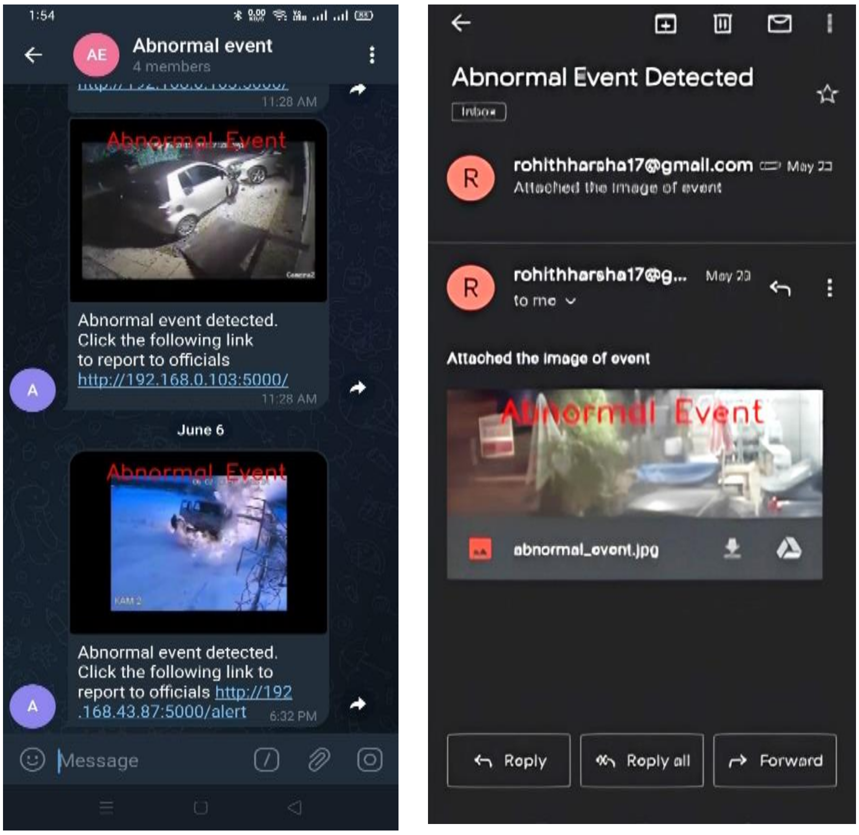
(b)



(c)

Figure 6. Abnormal activities (a) fire accident, (b) road accident, and (c) violation

The telegram app is a particularly useful tool for smart surveillance because it allows users to receive alerts in real-time, as shown in Figure 7, regardless of their location. This means that users can respond quickly and effectively to any abnormal activities that are detected, which can help to prevent or minimize damage and improve safety and security in the environment. As depicted in Figure 7(a), the telegram message is sent to the user when there is any abnormal activity displayed. When the system detects any abnormal activity, a link message is sent to the telegram app. A notification message is delivered to the user's registered email address in the web application when the link message is clicked, as illustrated in Figure 7(b).



(a) (b)

Figure 7. Notification in telegram app (a) message and (b) alert message to mail

4. CONCLUSION

A smart system for surveillance recording and capturing videos and images is designed and the images are also sent to smartphones when any abnormal event takes place. The application of this surveillance becomes beneficial to preserve both honesty and confidentiality. It is authenticated and encrypted on the receiving end, which allows only the intended recipient to read the information. This surveillance application becomes more useful during emergencies, such as a sick old person, military installations, smart homes, offices, factories, and diamond stores. Through this system, abnormal events like fire explosions, road accidents, and violence will be automatically detected by the web application and alert the user at the right time. This helps us to get immediate help from the officials thereby reducing the loss of life and properties. This system can be installed in places like industries, highway roads, and places of public gathering. The future work will be to count the population nearby and determine its worldwide location.

REFERENCES

[1] H. D. Park, O. G. Min, and Y. J. Lee, "Scalable architecture for an automated surveillance system using edge computing," *Journal of Supercomputing*, vol. 73, no. 3, pp. 926–939, 2017, doi: 10.1007/s11227-016-1750-7.

[2] G. Sreenu and M. A. S. Durai, "Intelligent video surveillance: a review through deep learning techniques for crowd analysis," *Journal of Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0212-5.

[3] W. Rahmaniar and A. Hernawan, "Real-time human detection using deep learning on embedded platforms: a review," *Journal of Robotics and Control (JRC)*, vol. 2, no. 6, 2021, doi: 10.18196/26123.

[4] L. Rajendran and R. S. Shankaran, "Bigdata enabled realtime crowd surveillance using artificial intelligence and deep learning," *Proceedings - 2021 IEEE International Conference on Big Data and Smart Computing, BigComp 2021*, pp. 129–132, 2021, doi: 10.1109/BigComp51126.2021.00032.

[5] S. M. Marvasti-Zadeh, L. Cheng, H. Ghanei-Yakhdan, and S. Kasaei, "Deep learning for visual tracking: a comprehensive survey," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 5, pp. 3943–3968, 2022, doi: 10.1109/TITS.2020.3046478.

[6] G. Ding, W. Chen, S. Zhao, J. Han, and Q. Liu, "Real-time scalable visual tracking via quadrangle kernelized correlation filters," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 1, pp. 140–150, 2018, doi: 10.1109/TITS.2017.2774778.

[7] A. S. Jadhav and D. Sudarshan, "Real time embedded video streaming using Raspberry Pi," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 5, no. 11, pp. 19315–19320, 2016, doi: 10.15680/IJIRSET.2016.0511109.

[8] A. S. Lande and B. P. Kulkarni, "Wireless security camera system," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 10, pp. 2751–2754, 2019, doi: 10.35940/ijitee.J9568.0881019.

[9] K. E. Ko and K. B. Sim, "Deep convolutional framework for abnormal behavior detection in a smart surveillance system," *Engineering Applications of Artificial Intelligence*, vol.67, pp. 226-234, 2018, doi:10.1016/j.engappai.2017.10.001.

[10] K. B. Lee and H. S. Shin, "An application of a deep learning algorithm for automatic detection of unexpected accidents under bad CCTV monitoring conditions in tunnels," *Proceedings - 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications, Deep-ML 2019*, pp. 7–11, 2019, doi: 10.1109/Deep-ML.2019.00010.

[11] D. Tian, C. Zhang, X. Duan, and X. Wang, "An automatic car accident detection method based on cooperative vehicle infrastructure systems," *IEEE Access*, vol. 7, pp. 127453–127463, 2019, doi: 10.1109/ACCESS.2019.2939532.

[12] T. Kalyani, S. Monika, B. Naresh, and M. Vucha, "Accident detection and alert system," *International Journal of Innovative Technology and Exploring Engineering*, vol. 8, no. 4S2, pp. 227–229, 2019, doi: 10.56726/irjmets33353.

[13] G. Wu, W. Lu, G. Gao, C. Zhao, and J. Liu, "Regional deep learning model for visual tracking," *Neurocomputing*, vol. 175, no. PartA, pp. 310–323, 2015, doi: 10.1016/j.neucom.2015.10.064.

[14] D. Singh and C. K. Mohan, "Deep spatio-temporal representation for detection of road accidents using stacked autoencoder," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 3, pp. 879–887, 2019, doi: 10.1109/TITS.2018.2835308.

[15] W. Liu, W. Luo, D. Lian, and S. Gao, "Future frame prediction for anomaly detection-a new baseline," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 6536–6545, 2018, doi: 10.1109/CVPR.2018.00684.

[16] A. Yasamorn, A. Wongcharoen, and C. Joochim, "Object detection of pedestrian crossing accident using deep convolutional neural networks," *Proceedings - 2022 Research, Invention, and Innovation Congress: Innovative Electricals and Electronics, RI2C 2022*. IEEE, pp. 297–303, 2022, doi: 10.1109/RI2C56397.2022.9910331.

[17] S. Supriya, S. P. Shankar, J. H. Jain, L. L. Narayana, and N. Gumalla, "Car crash detection system using machine learning and deep learning algorithm," *IEEE International Conference on Data Science and Information System, ICDSIS 2022*, 2022, doi: 10.1109/ICDSIS55133.2022.9915889.

[18] S. Ghosh, S. J. Sunny, and R. Roney, "Accident detection using convolutional neural networks," *2019 International Conference on Data Science and Communication, IconDSC 2019*. IEEE, 2019, doi: 10.1109/IconDSC.2019.8816881.

[19] A. P. Shah, J. B. Lamare, T. Nguyen-Anh, and A. Hauptmann, "CADP: a novel dataset for CCTV traffic camera based accident analysis," *Proceedings of AVSS 2018 - 2018 15th IEEE International Conference on Advanced Video and Signal-Based Surveillance*. IEEE, 2019, doi: 10.1109/AVSS.2018.8639160.

[20] M. Abadi *et al.*, "Tensor flow: large-scale machine learning on heterogeneous distributed systems," *arXiv:1603.04467*, 2016, doi: 10.48550/arXiv.1603.04467.

[21] K. B. Lee and H. S. Shin, "An application of a deep learning algorithm for automatic detection of unexpected accidents under Bad CCTV monitoring conditions in tunnels," *Proceedings - 2019 International Conference on Deep Learning and Machine Learning in Emerging Applications, Deep-ML 2019*, pp. 7–11, 2019, doi: 10.1109/Deep-ML.2019.00010.

[22] V. D. Gowda, A. Sharma, A. S. Naik, R. S. Meena, J. M. Kudari, and S. Purushotham, "Design and implementation of a system for vehicle accident reporting and tracking," *7th International Conference on Communication and Electronics Systems, ICCES 2022 - Proceedings*. IEEE, pp. 349–353, 2022, doi: 10.1109/ICCES54183.2022.9835896.

[23] L. Cao, Q. Jiang, M. Cheng, and C. Wang, "Robust vehicle detection by combining deep features with exemplar classification," *Neurocomputing*, vol. 215, pp. 225–231, 2016, doi: 10.1016/j.neucom.2016.03.094.

[24] A. Arinaldi, J. A. Pradana, and A. A. Gurusinga, "Detection and classification of vehicles for traffic video analytics," *Procedia Computer Science*, vol. 144, pp. 259–268, 2018, doi: 10.1016/j.procs.2018.10.527.

[25] H. S. Lee and Y. Kim, "Development of a deep-learning based tunnel incident detection system on CCTVs," in *Development of a deep-learning based tunnel incident detection system on CCTVs*, 2017, vol. 19, no. 6, pp. 915–936, doi: 10.9711/KTAJ.2018.20.6.1161.

BIOGRAPHIES OF AUTHORS

Amsaveni Avinashiappan is working as a professor in the Department of Electronics and Communication Engineering, Kumaraguru College of technology, Coimbatore. She obtained her Ph.D. degree from Anna University, Chennai in the year 2016. She has around 24 years of academic experience. Her areas of research interest include Antennas, information security, and signal processing. She has published around 60 research papers in international journals and conference proceedings. She can be contacted at email: amsaveni.a.ece@kct.ac.in.

Harshavarthan Thiagarajan has received his B.E degree in Electronics and Communication Engineering in 2022 from, Kumaraguru College of technology, Coimbatore. His areas of interest include IoT, embedded systems, and software technologies. He is working in IoT related field. He can be contacted at email: tharshavarthansvhv@gmail.com.

Harshwarth Coimbatore Mahesh has received his B.E degree in Electronics and Communication Engineering in 2022 from, Kumaraguru College of Technology, Coimbatore. His areas of interest include IoT, cloud computing, DevOps tools and technologies. He has experience with CI/CD and automation projects. He can be contacted at email: harshwarth01@gmail.com.

Rohith Suresh has received his B.E degree in Electronics and Communication Engineering in 2022 from, Kumaraguru College of Technology, Coimbatore. His areas of interest include networking systems, embedded systems, and operating systems. He is working in networking-related field. He can be contacted at email: rohithsuresh183@gmail.com.