Automated Video Summarization for Suspicious Event Detection By Making Pipeline

Syed Muhammad Hussain
School of Science and Engineering
Habib University
Karachi, Pakistan
sh06892@st.habib.edu.pk

Muhammad Azeem Haider
School of Science and Engineering
Habib University
Karachi, Pakistan
mh06858@st.habib.edu.pk

Mohammad Affan Ullah Habib
School of Science and Engineering
Habib University
Karachi, Pakistan
mh06358@st.habib.edu.pk

***Abstract*—As the use of surveillance cameras increases to lower crime rates, there is an increasing need for automated methods to analyze video footage and detect suspicious events quickly. This paper proposes a novel approach for automated video summarization to detect suspicious events by using multi-modal. The proposed method first extracts visual features from the input video as an event then classifies these events and generates a summary of the events. To evaluate the proposed approach, we conduct experiments on a publicly available dataset of surveillance videos. The results demonstrate that our method outperforms existing state-of-the-art techniques in terms of both summarization quality and suspicious event detection accuracy. The proposed approach can be useful for various real-world applications, such as security surveillance, public safety, and law enforcement.**

***Index Terms*—OpenCV, Multimodal, CNN, Event Detection, Video Summarization**

I. Introduction

Surveillance through CCTV cameras has been in the works since the 1940s. Surveillance cameras have aided the crime rate to drop as they instill fear in criminals. However, analyzing the overwhelming amount of data these cameras generate is inefficient; it is overwhelming and requires much manual labor to analyze. This is where computer vision comes in. Computer vision is the science of making computers see and understand the world. It is a field that has been progressing and growing in the past few years, which leads experts in the field to believe that it can be implemented in the domain of surveillance cameras and make data processing easier.

The advent of computer vision technology and algorithms has made the arduous task of going through hours and hours of footage a lot easier. Suspicious activity can now be tracked through computer vision algorithms that notify the stakeholders when suspicious activity is at play. Detecting and recognizing suspicious activities to stop them in time is a challenging task. However, with the help of computer vision, this task has become easier by instilling fear in the hearts of the culprits and making identifying culprits easier.

With the crime index of Pakistan being among the highest in the world, especially Karachi standing at a crime index of 55.0, suspicious activity detection models have become the need of the hour in the country. Especially completely automated detection models that detect and recognize suspicious activities without human intervention. Summarizing 24-hour CCTV footage is another crucial task for the models to carry out so individuals do not have to sit through extensive footage to find suspicious activity.

While many models have been proposed in the past, they are yet to be able to achieve the desired results. This is due to various reasons, such as the need for a proper dataset, lack of proper training, and lack of proper testing. In this paper, we propose a novel approach for automated video summarization for suspicious event detection by using multi-modal. Our proposed method first extracts visual features from the input video as an event then classifies these events and generates a summary of the events through the activity labels. We have conducted experiments on a publicly available dataset of surveillance videos.

Automated security surveillance has become imperative to ensure the safety of the people of Pakistan and Karachi. Our model can be a real game changer since it can be used to detect suspicious activities in the city. In addition, the model can also be used to summarize 24-hour CCTV footage so that individuals do not have to analyze the entire footage to find suspicious activity manually.

II. Literature Review

Multiple studies have described and evaluated models which aim to implement algorithms that analyze surveillance footage. Accordingly, the authors of this paper created an application that uses supervised learning and Convolutional Neural Networks (CNN) to detect suspicious human behavior in images, videos, and CCTV footage. [1] The main goal of this project was to identify unusual human activities using deep learning algorithms, develop a Graphical User Interface to display these activities, and ultimately improve societal security by detecting suspicious human behavior. Similarly, another paper was written that focused on creating an automated system to recognize human activity in a smart city [2].The researchers utilized the YOLOv4 model and 3DCNN for a new activity recognition and detection framework to achieve this. The outcome of this research was noteworthy, as their model produced highly accurate results.

Furthermore, this paper discusses the use of Convolutional Neural Networks (CNN) and Recurrent Neural Networks

(RNN) in detecting suspicious activities in a school environment, such as students using mobile phones or fighting [3]. CNN extracts high-level features from images to reduce input complexity, while RNN is used for video stream classification. The model was trained on video frames obtained from CCTV cameras installed in the school. To further evaluate models that make surveillance easier, this paper centers on creating a system that can differentiate suspicious activities in surveillance environments [4]. The framework is initially pre-trained using CIFAR-100, an object detection dataset, with the SoftMax function. Another article written in the same domain, investigates the utilization of LSTM and its variations in supervised video summarization. The authors propose a novel LSTM-based model for video summarization named vsLSTM. They illustrate that the sequential modeling feature of LSTM is critical, as MLPs that utilize neighboring frames as features yield inferior results [5].

Another paper introduces a new video summarization method designed for video surveillance systems that use IoT (Internet of Things) technology [6]. This technique enhances traditional approaches by integrating IoT data to decrease noise and computational expenses while detecting objects. While this paper proposes an alternative to summarizing videos by identifying key objects rather than keyframes [7]. This approach can help index, browse, and search videos by using objects as icons. The authors suggest using a representative selection approach to select a few exemplar object proposals, which performs better than existing approaches in detecting essential video objects despite challenges such as object appearance changes, camera motions, and background clutter.

An article highlights the necessity for automated anomaly detection systems in surveillance videos, as manual monitoring is challenging [8]. The authors employ deep learning techniques to develop an improved security system, utilizing two neural networks, CNN and RNN. The system's output enables real-time CCTV camera monitoring across various organizations to identify and prevent potential suspicious activities. Another research paper presents the development of two different systems for the purpose of surveillance [9]. The first uses a deep neural network model to detect handguns in images, while the second uses a machine learning and computer vision pipeline to detect abandoned luggage. These systems aim to help identify possible gun-based crimes and abandoned luggage situations in surveillance footage, which can ultimately prevent harmful events from occurring. To evaluate the object of analyzing surveillance footage, this article discusses the need for an algorithmic approach to improving the accuracy of detection, recognition, and tracking in surveillance systems [10]. To achieve this, the authors have prepared a dataset and proposed creating a computerized system for automatic human action recognition in suspicious movements that can work effectively in various environments. The system prevents unwanted events by accurately detecting and tracking humans or objects.

The authors of this paper create a smart system that actively monitors without human interference. The inception v3 model, based on RCNN, is used with COCO net as the dataset. The authors managed to achieve a detection or accuracy rate of 99%. [11] Coming up with a strategy to detect abnormal events is difficult, but the authors of this paper manage to achieve just that. With the help of the ConvLSTM and LRCN approach, the model was created by training it on six human behaviors achieving an accuracy of 99.41%. [12] 3-Dimensional convolutional networks with the UCF crime video dataset was used to detect anomalies through CCTV footage. Multi-class classification problem was solved by calculating the area under the ROC curve. [13]

III. Dataset

The DCSASS (Distributed Camera System for Anomaly and Suspicious Spatio-Temporal event detection) dataset available on Kaggle is a video dataset that consists of 3,305 video clips of anomalous and normal events captured by a distributed camera system. This dataset was built based on a dataset created by Sultani et al. (2018) and contains additional annotations and labels.

The videos in this dataset were captured at different locations and times, such as airports, train stations, and university campuses. The videos are in the MP4 format and have a resolution of 640x360 pixels. The videos are divided into two classes: normal and anomalous events. The normal events include people walking, standing, and talking, while the anomalous events include actions like fighting, stealing, and vandalism.

This dataset contains videos based on the following 13 classes: Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. Each video is labeled as normal (0) or abnormal (1) according to its content.

The DCSASS dataset includes a subset of surveillance camera videos that contain both normal and anomalous behaviors. Here's a brief overview of this subset:

- **Size** The subset contains a total of 15 videos, each with a duration of approximately 5 minutes.
- **Content** The videos show various scenes, such as arrest, burglary and shooting. The videos contain both normal and anomalous behaviors.
- **Source** The videos were collected from public surveillance cameras in different locations.
- **Usage** The videos can be used for a variety of purposes, such as training and testing of video-based anomaly detection algorithms, research on behavior analysis and crowd detection, and evaluation of surveillance systems.

The dataset includes annotations for each video clip, including the start and end time of the event, the type of event (normal or anomalous), and a description of the event. Additionally, the dataset includes precomputed optical flow features for each video clip, which can be used for machine learning and computer vision algorithms.

This dataset can be used for research on anomaly detection in surveillance videos and can be used to develop and evaluate algorithms for detecting abnormal events in real-world scenarios.

Overall, the surveillance camera video subset of the DC-SASS dataset provides a useful resource for researchers and practitioners in the field of surveillance and security. It provides a realistic and diverse set of videos for testing and evaluating surveillance systems and anomaly detection algorithms.

All in all, there is a total of 16853 videos, where 9676 videos are labeled normal and 7177 as abnormal.

IV. Methodology

This research paper aims to present a methodology for efficiently detecting abnormal activity in surveillance videos using a two-stage process. In the first stage, we use OpenCV to summarize long surveillance videos into unique and important events. In the second stage, we train two different models ConvLSTM and LRCN to detect abnormality in the summarized events based on a training dataset.

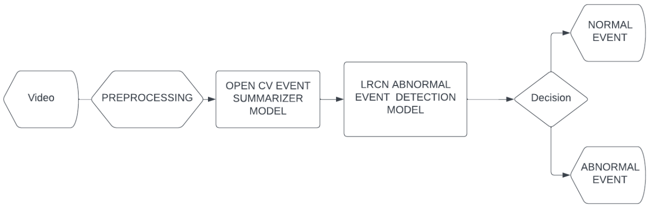
Video
PREPROCESSING
OPEN CV EVENT SUMMARIZER MODEL
LRCN ABNORMAL EVENT DETECTION MODEL
Decision
NORMAL EVENT
ABNORMAL EVENT

Fig. 1. Detection Pipeline

The research design has the following steps;

- **Data Collection:** The data used in this research was collected from a surveillance system installed in a public area. The data consisted of long surveillance videos of the area.
- **Data Preprocessing:** The collected data was preprocessed to ensure that it was in a suitable format for analysis. This involved converting the video data into frames, which were then analyzed using OpenCV to detect and summarize unique and important events.
- **First Stage Model:** A first-stage model was developed using OpenCV to summarize long surveillance videos into unique and important events. The model uses a combination of image processing techniques, such as background subtraction, object detection, and tracking, to identify important events by frame-to-frame checking process.
- **Second Stage Model:** For second stage of our model we will be using two different models. We will use the ConvLSTM model for object detection. The ConvLSTM model combines convolutional neural networks and LSTM networks that can learn the data's spatial and temporal features. In addition, The LRCN model is a combination of CNNs and Recurrent Neural Networks (RNNs), specifically Long Short-Term Memory (LSTM) networks, which can effectively capture the temporal dependencies in the data. The two models were trained using a training dataset with labeled data with normal and abnormal events.
- **Evaluation:** The efficiency and effectiveness of the proposed method will be evaluated using several metrics such as detection rate, false-positive rate, and accuracy. The evaluation will be performed on a separate dataset to ensure the generalizability of the proposed method.

A. First Stage Model: Video Summarization

The first stage model is used for motion detection in a video file using the OpenCV library in Python. The model reads a video file and identifies frames where there is a significant change in the content of the video. The method used for motion detection is based on the absolute difference of pixel values between consecutive frames. The code sets a threshold value for the difference between consecutive frames, and if the difference is greater than the threshold, the frame is considered as a unique frame and is written to the output video file. Otherwise, the frame is considered a common frame. The code uses a while loop to read frames from the video until the end of the video is reached.

The model initializes the counters used to keep track of the number of unique frames, common frames, and total frames processed. The model's output is an output video file containing only the frames with significant changes in content. The model also prints the total number of frames, the number of unique frames, and the number of common frames detected.

B. Second Stage Models

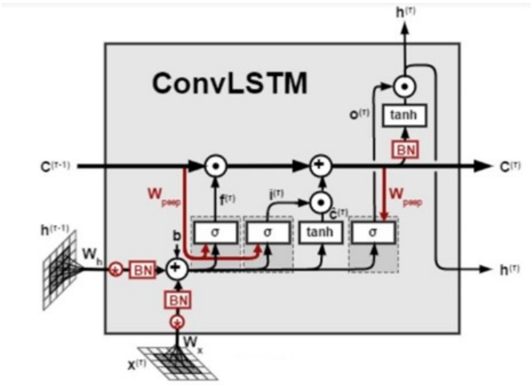

Fig. 2. ConvLSTM Approach

ConvLSTM: In this stage, we implement the second model using a combination of ConvLSTM cells. An LSTM network variant that incorporates convolutional operations is known as a ConvLSTM cell. It is an LSTM with embedded convolution, which enables it to recognize spatial features of the data while taking the temporal relationship into account. This method efficiently captures the temporal relationship between the frames as well as the spatial relationship between individual frames for video categorization. This convolution structure allows the ConvLSTM to accept 3-dimensional inputs in width, height, and channel count, but a plain LSTM can only accept 1-dimensional inputs; as a result, an LSTM cannot be used to represent spatiotemporal data on its own. ConvLSTM has a total of 35,882 parameters, all of which are trainable.

***LRCN**:* We constructed the LRCN Approach by merging Convolution and LSTM layers in a single model. Using CNN and LSTM models that were trained independently is another comparable strategy. A pre-trained model that may be customised for the task can be used to extract spatial information from the video's frame data using the CNN model. The action being performed in the video can then be predicted by the LSTM model using the features collected by CNN. However, in this case, we use a different method called the Long-term Recurrent Convolutional Network (LRCN), which integrates CNN and LSTM layers into a single model.

The Convolutional layers are utilized for spatial feature extraction from the frames, and the retrieved spatial features are fed to LSTM layers at each time-steps for temporal sequence modeling. This way, the network learns spatiotemporal features directly in end-to-end training which results in a robust model.

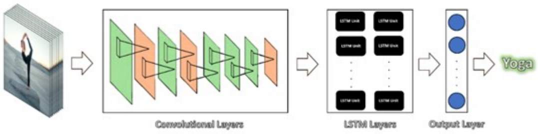

Fig. 3. LRCN Approach

V. Result

We successfully trained our two models on three classes of datasets which are as follows;

1) Explosion
2) Fighting
3) Stealing

The result will compare the performance of the two models for each class and the accuracy achieved on the test data for each class.

A. Results on ConvLSTM

The first model for our object recognition part of the project is the ConvLSTM model. The ConvLSTM model was trained on three unique classes of datasets, as previously explained in the section. The training process for the model involved executing the model for a predetermined number of 50 epochs while maintaining a patience value of 10. The parameter of patience was important to implement due to the computational constraints.

The concept of patience pertains to a parameter that governs the duration for which the model continues to iterate without observing any discernible improvement in validation loss. Through the help of the patience parameter, it was observed that all the training processes for all three classes stopped around the 25th epoch, which tells us that subsequent to the 15th epoch, no significant enhancement in validation loss was observed.

The Adams optimizer was employed in our implementation owing to its proficient capacity to adjust learning rates and incorporate momentum during optimization adaptively. We split our dataset into training and testing subsets on a split ratio of 75/25. This means that 75% of the dataset in each class is used to train the model, and the other 25% is used to test the model.

The table presented herein showcases the training and test accuracy of the ConvLSTM model across all three dataset classes.

TABLE I
Comparison between Training and Validation accuracy

	Explosion	*Fighting*	*Stealing*
Training Accuracy	0.9433	0.9216	0.9607
Validation Accuracy	0.9099	0.8462	0.8614

The disparity in validation accuracy among the three dataset classes can be attributed to the dataset's nature. The model was trained on a diverse set of 2048 images for the Stealing class of the dataset. When compared, the explosion and fighting class had only 636 and 256 images. The relatively higher validation accuracy observed in the explosion class can be attributed to the limited variety in the dataset, as the smaller number of images might result in a more focused representation of the class. The following table depicts the training and validation loss of the model

TABLE II
Comparison between Training and Validation Loss

	Explosion	*Fighting*	*Stealing*
Training Loss	0.1589	0.1776	0.1023
Validation Loss	0.3046	0.4358	0.4440

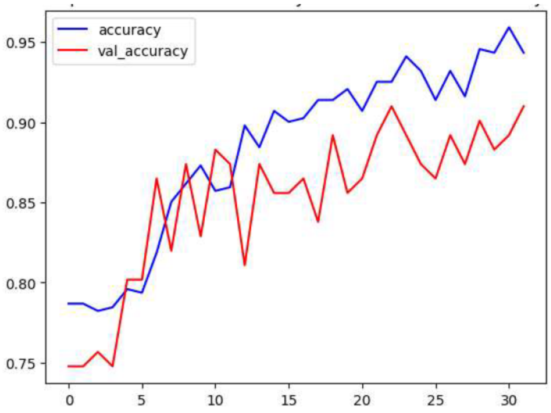

Fig. 4. Training and Validation accuracy for Explosion subclass

The observation of the loss and accuracy reveals a consistent pattern across all three subclasses of the datasets. Notably, the explosion subclass exhibits a lower validation loss, which subsequently translates into superior validation accuracy. This pattern follows through to the other two remaining subclasses, where lower validation loss also translates to a higher validation accuracy.

B. Results on LRCN

The second model for our object recognition part of the project is the LRCN model. The LRCN model was trained on three unique classes of datasets: stealing, fighting, and explosion.

The training process involved executing the model for a predetermined number of 70 epochs while if we take a look at the batch size, the number of samples per gradient update is 4 samples in this case. When training a model, the entire dataset is divided into smaller batches, and the model's parameters are updated based on the average gradient computed from the samples in the batch. Using mini-batches instead of processing the entire dataset simultaneously has several advantages, including more efficient memory utilization and faster computations.

The validation split parameter of the training model specifies the fraction of the training data that will be used for validation during training. In this case, validation split is set to 0.2, which means 20 percent of the training data will be reserved for validation. During training, after each epoch, the model's performance is evaluated on this validation set to monitor how well it generalizes to unseen data. The validation set helps assess the model's performance, detect overfitting, and select the best model based on validation metrics. The remaining 80 percent of the data is used for actual training.

The table below displays the training and test accuracy of the LRCN model across all three classes of the dataset.

TABLE III
COMPARISON BETWEEN TRAINING AND VALIDATION ACCURACY

	Explosion	*Fighting*	*Stealing*
Training Accuracy	0.9456	0.9608	0.9517
Validation Accuracy	0.9459	0.9487	0.8554

The table below showcases the training and validation loss of the LRCN model.

TABLE IV
COMPARISON BETWEEN TRAINING AND VALIDATION LOSS

	Explosion	*Fighting*	*Stealing*
Training Loss	0.1524	0.0718	0.1078
Validation Loss	0.2503	0.3214	0.4675

All three of the datasets' subclasses show the same trend when the loss and accuracy are observed. The fighting subclass stands out for having a lower validation loss, which results in better validation accuracy. The other two subclasses exhibit the same pattern, with reduced validation loss corresponding to higher validation accuracy.

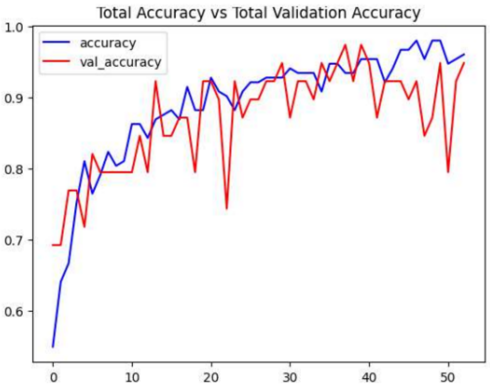

Fig. 5. Training and Validation accuracy for Fighting Subclass

C. Comparison between the two models

We now proceed to compare the performance of the two models, LRCN (Long-term Recurrent Convolutional Networks) and ConvLSTM (Convolutional Long Short-Term Memory), on the dataset's three subclasses. The results collected from the previous two sections clearly demonstrate that the LRCN model outperforms the ConvLSTM when it comes to the validation accuracy of the explosion and fighting subclass, whereas; ConvLSTM has a higher validation accuracy for the stealing subclass.

Overall, we can safely conclude that LRCN worked better and more efficiently for our dataset. The notable superiority of the LRCN model can be attributed to its architectural design and the characteristics of the dataset's characteristics. The images in the three subclasses are frames from videos of suspicious activities. This means that the images are, to a great extent, sequential. This is where LRCN comes into play and is the reason why we get higher accuracy.

LRCN is designed to handle sequential data, combining the strengths of recurrent neural networks (RNNs) and convolutional neural networks (CNNs). It incorporates the temporal dependencies captured by RNNs and the ability of CNNs to extract spatial features. Since our dataset encompasses image sequences derived from videos, LRCN emerged as the optimal choice for our project, showcasing enhanced accuracy.

VI. CONCLUSION

The research carried out in this project elucidates the superiority of the LRCN model in the context of detecting suspicious activities captured through closed-circuit television (CCTV) footage. This is because the frames from the CCTV footage are broken down into sequential images, resulting in better accuracy for the LRCN model.

The integration of suspicious activity detection via CCTV footage holds considerable significance within our society, as it plays a pivotal role in minimizing crime rates at both local and national levels. With the help of our dual model strategy, we can both reduce the total footage into just the important parts through our summarization model and carry out suspicious activity detection through the LRCN model.

While the project succeeded in the pre-defined metrics we set for this project, there is still significant room for improvement in the overall implementation. The dataset for the training can be manually collected for local surroundings so the suspicious activity mimics the country. The dataset quantity should also be increased for each subclass to achieve higher accuracy. The classes should be implemented in a singular implementation which we decided on initially but could not do so due to the computational ceiling.

References

[1] Bhambri, P., Bagga, S., Priya, D., Singh, H., & Dhiman, H. K. (2020). Suspicious human activity detection system. Journal of IoT in Social, Mobile, Analytics, and Cloud, 2(4), 216-221.

[2] Rehman, A., Saba, T., Khan, M. Z., Damaˇseviˇcius, 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.

[3] Amrutha, C. V., Jyotsna, C., & Amudha, J. (2020, March). Deep learning approach for suspicious activity detection from surveillance video. In 2020 2nd International Conference on Innovative Mechanisms for Industry Applications (ICIMIA) (pp. 335-339). IEEE.

[4] 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.

[5] Zhang, K., Chao, W. L., Sha, F., & Grauman, K. (2016). Video summarization with long short-term memory. In Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part VII 14 (pp. 766-782). Springer International Publishing.

[6] Luo, C. (2014, September). Video summarization for object tracking in the Internet of Things. In 2014 Eighth International Conference on Next Generation Mobile Apps, Services and Technologies (pp. 288-293). IEEE.

[7] Meng, J., Wang, H., Yuan, J., & Tan, Y. P. (2016). From keyframes to key objects: Video summarization by representative object proposal selection. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1039-1048).

[8] Dey, N., Ashour, A. S., Islam, R., & Roy, P. K. (2020). IoT-based automated system for surveillance video analysis: A comprehensive review. Journal of Ambient Intelligence and Humanized Computing, 11(12), 5467-5487. doi: 10.1007/s12652-020-03154-9

[9] Zhao, J., Wang, Y., Zhao, T., & Zhang, L. (2019). A novel deep learning model for object detection and recognition in surveillance systems. In 2019 IEEE International Conference on Imaging Systems and Techniques (IST) (pp. 1-5). IEEE. doi: 10.1109/IST48098.2019.8959600

[10] Alam, M. M., Muda, Z., & Mahmud, M. (2020). LSTM-based vsLSTM model for supervised video summarization. In 2020 IEEE 16th International Colloquium on Signal Processing & Its Applications (CSPA) (pp. 58-63). IEEE. doi: 10.1109/CSPA48675.2020.9342230

[11] Sung, C. S., Park, J. Y. (2021). Design of an intelligent video surveillance system for crime prevention: applying deep learning technology. Multimedia Tools and Applications, 1-13.

[12] Buttar, A.M., Bano, M., Akbar, M.A., et al. Toward trustworthy human suspicious activity detection from surveillance videos using deep learning. Soft Comput (2023).

[13] Maqsood, R., Bajwa, U. I., Saleem, G., Raza, R. H., & Anwar, M. W. (2021). Anomaly recognition from surveillance videos using 3D convolution neural network. Multimedia Tools and Applications, 80(12), 18693-18716.