Deep Reinforcement Learning-based Anomaly Detection for Video Surveillance

Sabrina Aberkane and Mohamed Elarbi-Boudihir
E-mail: s.aberkane@esi-sba.dz, m.elarbiboudihir@esi-sba.dz
ESI-SBA, High school of Computer Sciences, Sidi Bel-Abbès, Algeria

Student paper

Keywords: deep reinforcement learning, anomaly detection, video surveillance

Received: June 22, 2021

The anomaly detection in automated video surveillance is considered as one of the most critical tasks to be solved, in which we aim to detect a variety of real-world abnormalities. This paper introduces a novel approach for anomaly detection based on deep reinforcement learning. In recent years, deep reinforcement learning has been achieving a significant success in various applications with data of a high degree of complexity such as robotics and games, by mimicking the way humans learn from experiences. Generally, the state-of-the-art methods classify a video as normal or abnormal without pinpointing the exact location of the anomaly in the input video due to the unlabeled clip-level data in training videos. We focus on adapting the prioritized Dueling deep Q-networks to the anomaly detection problem. This model learns to evaluate the anomaly in video clips by exploiting the video-level label to obtain a better detection accuracy. Extensive experiments on 13 cases class of real-word anomaly show that our DRL agent achieved a near optimal performance with a high accuracy in the real world video surveillance system compared to the state-of-the-art approaches.

Povzetek: Razvita je nova metoda globokega spodbujevalnega učenja za prepoznavanje anomalij pri videonadzoru.

1 Introduction

In video surveillance systems, the ability to recognize actions can be used to detect and prevent abnormal or suspicious events. Such intelligent systems would be greatly helpful for providing security to people. Indeed, the surveillance cameras also make some people feel more safe, knowing that the culprits are being watched. Generally, these kinds of systems are powered by different algorithms[1, 2, 3], which are action recognition, object tracking and object classification. The conception of such algorithms is typically addressed in computer vision research which works on how to make machines gain some human understanding of data from digital images and videos.

In this work, we focus on designing an intelligent visual surveillance system, which aims to detect abnormalities in urban places. The anomaly detection task has been one of the most talked about issues for decades, and is still a very hot topic due to the broad real-world applications including visual surveillance. To address the abnormality detection problem, some researchers attempted to give a general definition, which covers all existing normal/abnormal motions in daily life. Otherwise, many workers considered the task as an activity classification problem. All these researches share one main purpose which is to build an intelligent machine imitating the human capability of interpreting complex human behaviors in a cluttered environment. Is it possible that a machine could perform the recognition task at the same level as humans?

In our paper, we try to answer the question above by demonstrating that a machine can be as efficient as a human as long as it succeeds in reproducing the human's native learning mechanism. Indeed, we consider building an agent able to learn from the environment through a sequence of trial/error. The video analytics framework takes a video clip as input, then the pre-trained agent will provide two principal elements separately: the first is an estimation of the existence of an abnormal content in the video. The second, indicates the anomaly score for each segment in the video. The system architecture is inspired by a trending approach called deep reinforcement learning, which is a branch of machine learning based on the concept that an agent learns from interacting with an environment. The agent training was done through a new large-scale dataset of 1900 videos, 128 hours long, untrimmed real-world surveillance footage, with 13 cases of realistic abnormalities.

The organization of this paper is given as follows: After the introduction in this section, we will present the state-of-the-art anomaly detection approach in section 2. Subsequently, we introduce how to implement the system using the Dueling DQN as well as the anomaly localization in section 3. Section 4 will present the results and conclu-

System	Techniques	Scene	Localization	Dataset(s)	Accuracy
Schuldt et al [4]	Patterns Represtation SVM classification	Outdoor Indoor Uncrowded	Disable	Action database (available on request)	71,70 %
Hu et al [6]	Modeling trajectories Cluster-based	Outdoor crowded traffic	Disable	Action database (available on request)	80%
Qiao et al [7]	Modeling Optical flow Deep autoencorder	Outdoor Indoor	Disable	Lawn indoor plaza	98.33%
Khaleghi et al [8]	deep learning	Outdoor Indoor	Enable	UCSD dataset	88.1%
Shean Chong et al [9]	Spatiotemporal architecture Convolutionnel network Autoencoder	Outdoor Crowded	Enable	UCSD dataset	89.9%
Hasan et al [10]	Learning pattern model Autoencoder	Outdoor Indoor Crowded	Enable	CUHK Avenue UCSD Ped1 UCSD Ped2 Subway Entrance Subway Exit	70,2% 81.0% 90.0% 94.3% 80.7%
Sultani et al [11]	Multiple instance learning deep learning	Outdoor Crowded	Enable	UCF Dataset	75.41%
Oh et al [12]	Reinforcement learning	-	Disable	GeoLife GPS TST	35% 93%

Table 1: The comparison of properties between state-of-the-art approaches.

sions are finally made in Section 5.

2 Related work

The initial studies on anomaly detection have been reported in [4, 5, 6, 7], where the systems model the normal motion of individuals as trajectory, the anomaly is detected as a deviation from that normal trajectory. More recently, the following works used deep learning, which achieves competitive performances in video data. In this paper [8] a deep learning-based technique is used on both features extraction phase and rare events detection phase. The authors in [9] employ a spatiotemporal autoencoder to design a framework for events detection, which is composed of both spatial feature representation and the learning temporal evolution of the spatial features. Hasan et al [10] also used deep learning with autoencoders to present a fully connected autoencoder to learn the model of anomaly detection. To learn anomalous events, Waqas et al [11] constructed a new framework based on a deep multiple instance learning which leverages weakly labeled training videos. The authors in [12] applied the inverse reinforcement learning (IRL) for sequential anomaly detection, the system captures the sequence of actions of a target agent as input data, to return observation and evaluate whether it follows a normal pattern or not. The proposed approach works with a reward function which is inferred via IRL.

Table 1 compares the properties between previous systems. 'Scene' indicates where the anomaly occurs and the number of individuals on-site(crowded, uncrowded); 'Localization' specifies the property to locate where the anomaly is occurring. The 'scene' column of [12] is indicate as (-), because the used dataset is represented by a sequence of time-stamped points, each of which contains the information of latitude, longitude and altitude.

3 System modelisation

In this paper, we formulate the anomaly detection as a sequential decision-making process, then we propose the concept of a deep anomaly detection network to estimate the probability of covering an abnormality for each video segment. We assume that for a given video only a small number of segments contain the anomaly. Hence, we employ the reinforcement learning approach to train our detection network, which encourages high scores anomalous video segments as compared to normal segments. The latter is the equivalent of going through the process of finding the N segments with the highest abnormality scores from an input video.

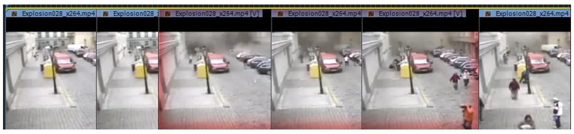

Figure 2: A sample of a distribution of anomaly segments in an abnormal video(red).

Deep reinforcement learning offers two different struc-

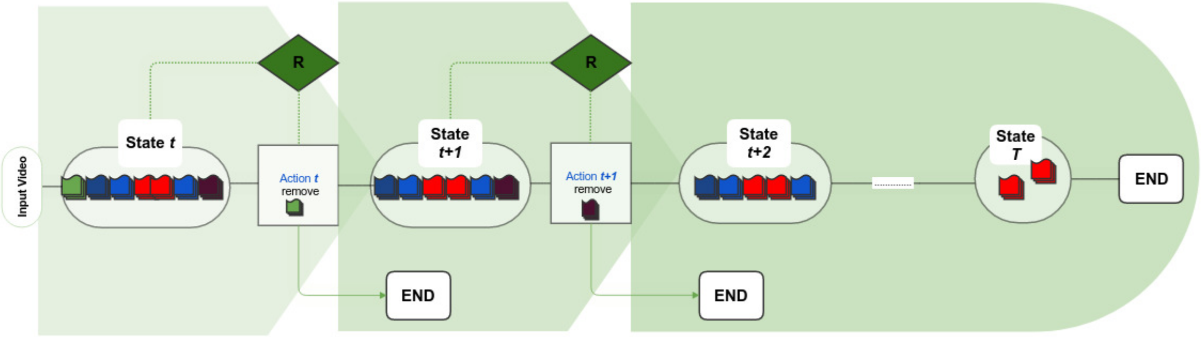

Figure 1: The system selection process.

tures where a machine can teach itself based on the results of its actions. One is the Deep Q-network DQN structure which only relies on the evaluation of actions to make decisions, while the other is the Deep Dueling Q-network DDQN that unlike the first, it takes advantage of both action value and environment information.

We adopted the dueling structure that was introduced by Wang et al [13], which explicitly sets apart the representation of state value and the state-dependent actions advantage via two separate streams.

$$Q(s, a) = V(s) + A(s, a) \tag{1}$$

$A(s, a)$ denotes the advantage stream that outputs a vector having dimensionality equal to the number of actions, representing the value of selecting an action a_i at state s_t. $V(s)$ denotes the value stream which outputs a scalar to represent the value of state s_t . The Value of a state is independent of actions. Both streams are combined at the end to produce the Q-function estimate through the combining module that can simply aggregate the value and the advantage streams as in [14]. The final output is a set of Q values $Q(s, a)$, one for each action.

As known, The markov decision process MDP is the underlying basis of any Reinforcement Learning model. Thus, to train the reinforcement learning agent for detecting abnormal events or behaviors on a video, it's very crucial to structure the environment of the processed video to the agent in some way to satisfy the Markov property, which is defined by a tuple (S, A, P, R, γ) such as states S, actions A, etc.

State: we consider that an input video V contains N segment x_i where $V = \sum_{i=1}^{N} x_i$. The state of the environment is a set of video segments $S = [x_1^V, x_2^V, x_3^V, ..., x_N^V]$. Where the initial state S_0 is composed of $N = 32$ segments, and for the next states, the initial number of segments decreases until it reaches the minimum number allowed $N = 5$, which is defined as the terminal state.

Action: An action $a_i \in A$ is every action executed by the RL agent to achieve its final objective, which is finding the set of segments x_i that are covering the anomaly to trigger an alarm.

Reward: In the literature, we find reward/penalization functions that are quite simple, as used in games [15], others much less simple like the one defined in [16]. Generally, in more complex environments, the reward function must be designed in such a way to suit the agent's environment to reflect what the agent has actually learned from the last episodes.

We can describe our environment as real, complex and unpredictable. In this context we based the rewarding/penalizing scheme on two axes:

– **Actions A related to segments X from normal videos V_n:** The reward value $r(V_n, a_i)$ reached the maximum when $Q(s_t, a_i) = 0$. The agent is penalized when $Q(s_t, a_i) = 1$, this means that the action is judged corresponding to the database annotations only.

– **Actions A related to segments X from abnormal videos V_{ab}:** The reward value $r(V_{ab}, a_i)$ reached the maximum when $Q(s_t, a_i)$ is similar to Expert E_c evaluation $Q(s_t, a_i) = E_c(s_t, a_i)$, and the minimum when $|Q(s_t, a_i) - E_c(s_t, a_i)| = 1$.

3.1 Selection process

Two different schemes are assumed to isolate the accurate anomaly representative segments. One is to directly score each segment and then consider the most representative one that is judged by the highest anomaly value from each video. The other is to remove the worst segment that is judged by the lowest anomaly value gradually, and the remaining segments are the most representative ones. still, due to the shortage of data (annotated videos) for the learning process, its preferable to maximize the iterations within one video, to increase the learning results even with using less resources. Additionally, it is not easy to select the right segments from the first run. It is obvious that finding the worst segment in a video is less complex and more rewarding than directly finding the segments of interest. Thus, we adopt the second plan, where an agent performs an action a_i by removing a segment x_i at state s_t. Therefore the action space is limited, and not the same at each step t.

The state s_t is represented by the remaining segments after t moves, the action a_i is represented by excluding a segment x_i at move t. Excluding a segment may lead to two states: s_{t+1} and termination, where termination means that we already found a set of segments that contain an anomaly.

The estimation feedback from environment r_i for (s_t, a_i) is not only provided by video level annotations, but also by video segment level annotations provided by a nominated expert E_c in anomaly recognition. The expert teaches our network as its recognition performance indicates the qualities of input segments x_i. To force the RL agent to learn the environment dynamics by itself. The environment does not provide any other feedback to the agent apart from the state and the reward.

3.2 Deep dueling-based anomaly detection

The system is mainly based on the dueling structure, which relies on two different sub-networks. They share the same features extractor layer [17]. The inputs is a video that is subsequently fragmented to 32 segments x_i, and considered as the initial state s_0 at t_0.

$$V = S_0 = \{x_i \mid 1 \leq i \leq 32\} \tag{2}$$

To transform the raw video segment data into an understandable format to the artificial agent, for each video segment x_i, we extract the visual features using the C3D Feature Extractor [17], then we obtain the corresponding final state format as follows:

$$V = S_0 = \{f_i^{x_i} \mid 1 \leq i \leq 32\} \tag{3}$$

Technically, each state s_t is a set of visual features representation $f_i^{x_i}$, which encapsulates all the data about the 3D convolutional features of the video segments ${X_i}^{Video}$.

To estimate the anomaly value of states s_t/video, all the extracted segment features $f_i^{x_i}$ are combined together, only to be used by the advantage stream as an input, which gives a single output value $V(s_t)$, the latter is given by equation 4. The stream's output indicates the probability of a video containing an anomaly.

$$V(s_t, \theta, \beta) = V(\{f_i^{x_i} \mid 1 \leq i \leq N\}, \theta, \beta) \tag{4}$$

The action-dependent advantage function $A(s, a)$ computes the value of the advantage of selecting a particular action(or segment) over the base value of being in the current state (or video).

We estimate the advantage stream by using the C3D features of the N remaining segments separately as input where $N = 32$ at step t_0/state s_0 presented by the following formula:

$$A(s_t, a_t, \theta, \alpha) = A(\{f_i^{x_i} \mid 1 \leq i \leq N\}, f_i^{x_i}, \theta, \alpha) \tag{5}$$

After the state value $V(s_t)$ and the video segments' value $A(s_t, a_i)$ are calculated, the output values of these two streams will be combined by an aggregation layer to evaluate each video segment x_i, in accordance with the following equation:

$$Q(s_t, a_i, \theta, \alpha, \beta) = V(s_t, \theta, \beta) + A(s_t, a_t, \theta, \alpha) - \left|\tfrac{1}{A}\right| \sum_{a_{t+1}} A(s_t, a_t, \theta, \alpha) \tag{6}$$

$Q(s_t, a_i)$ corresponds to the conditional probability of executing action a_i, which represents the deletion of a segment x_i from a state s_{t+1} at step t, the criteria of deletion is defined as a minimum Q value among values.

$$a_i \mid s_t = \begin{cases} \text{not} \operatorname*{argmin}_a(s_t, a) & x_i \text{ is a part of } s_{t+1} \\ \operatorname*{argmin}_a Q(s_t, a) & x_i \text{ is not a part of } s_{t+1} \end{cases} \tag{7}$$

The features f_i of the selected segments x_i will in turn be extracted at state s_t to return the next state s_{t+1} as following:

$$S_{t+1} = \begin{cases} \sum_{i=1}^{N} f_i^{x_i} - f_b^{x_b} & \text{if } b = \operatorname*{argmin}_a Q(s_t, a) \\ Terminal & \text{if } N = 5 \end{cases} \tag{8}$$

Then, the system judges the agent's decision by reward r_t provided by two different functions. For the videos annotated as normal, and abnormal defined by equations 9 and 10 respectively. The reward functions are as follows:

$$r_n^V(s_t, a_i, s_{t+1}) = \begin{cases} +1 & \text{IF } Q(s_t, a_i) = 0 \\ -Q(s_t, a) & \text{ELSE} \end{cases} \tag{9}$$

$$r_{ab}^V(s_t, a_i, s_{t+1}) = \begin{cases} +1 & \text{IF } Q(s_t, a_i) = E_c(s_t, a_i) \\ +|1 - ((s_t, a_i) - E_c(s_t, a_i))| & \text{IF } Q, E_c < T_h \vee Q, E_c > T_h \\ -|1 - ((s_t, a_i) - E_c(s_t, a_i))| & \text{IF } Q|E_c < T_h \vee Q|E_c > T_h \end{cases} \tag{10}$$

T_h is a predefined value $2 < T_h < 5$ representing the threshold that is used to signal an anomaly. The goal is reached once the anomaly is located in abnormal videos, or all segments of a normal video are well-judged $Q = 0$.

3.3 Prioritized experience replay

It is an improvement [18] to the Experience Replay mechanism used in the DQN algorithm that outperformed humans in Atari games [15]. The basic Experience Replay samples the batch uniformly (selecting the experiences randomly for training) these relevant experiences that occur rarely have practically no chance of being selected. As the name suggests, in Prioritized Experience Replay, a buffer is created to store the transition tuples by changing the sampling distribution based on a criterion to define the priority of each tuple of experience. The replay buffer is a cache D of

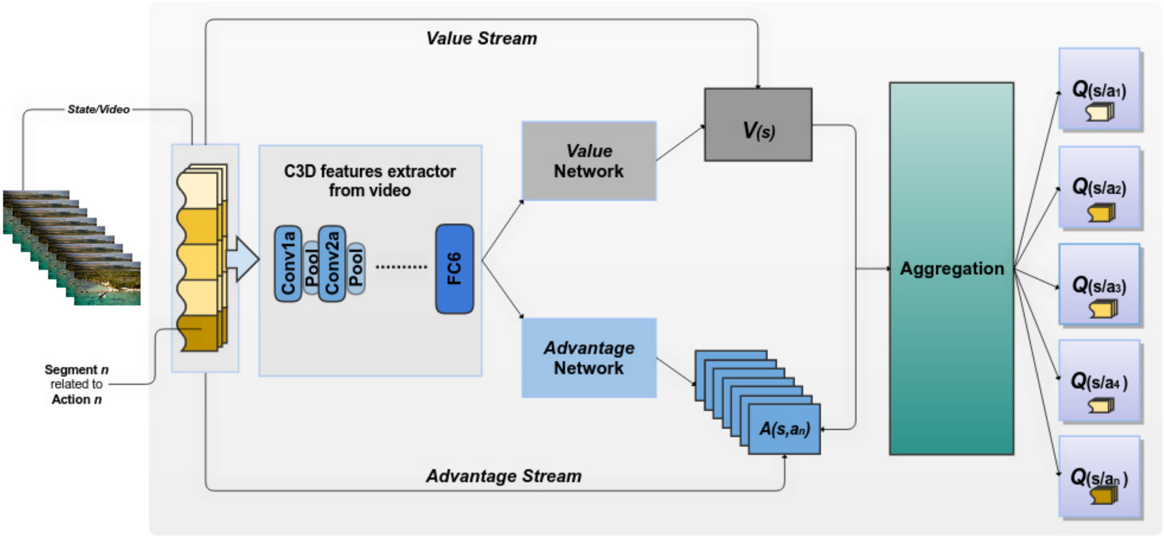

Figure 3: Deep dueling system for training a video surveillance agent.

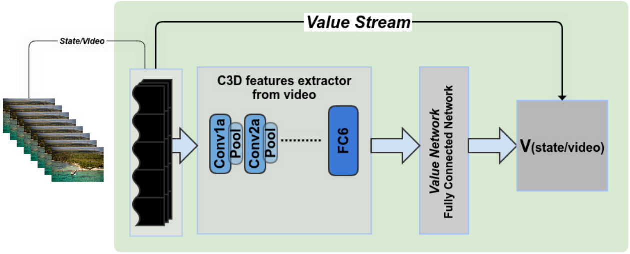
Value Stream
State/Video
C3D features extractor from video
Conv1a
Pool
Conv2a
Pool
FC6
Value Network Fully Connected Network
V(state/video)

Figure 4: The value stream.

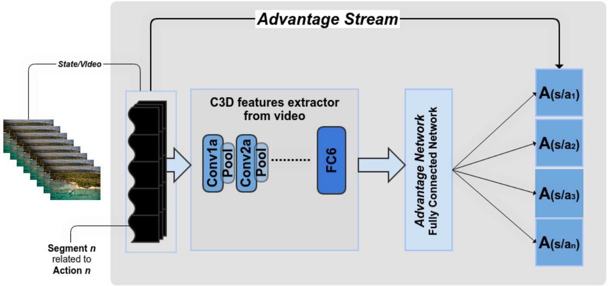
Advantage Stream
State/Video
C3D features extractor from video
Conv1a
Pool
Conv2a
Pool
FC6
Advantage Network Fully Connected Network
A(s/a1)
A(s/a2)
A(s/a3)
A(s/an)
Segment n related to Action n

Figure 5: The advantage stream.

finite size to feed the neural network. Each (s_t, a_t, r_t, s_{t+1}) transition relative to the tempral-difference (TD) error. The highest priority is given to samples that produced a larger TD error, plus some constant to avoid zero probability for an experience being chosen.

4 Experimental results

Unlike the dataset used in [12], which is based on GPS trajectory, we choose to train our system on a dataset including multiple anomaly classes that are similar to real-world anomalies, in order to get as close as possible to the context of surveillance videos. So, we perform experiments on a large-scale dataset named UCF-Anomaly-Detection-Dataset[11] to evaluate the performance of our DRL anomaly event detector agent. The dataset is composed of long untrimmed surveillance videos which cover 13 real-world anomalies, including Abuse, Arrest, Arson, Assault, Accident, Burglary, Explosion, Fighting, Robbery, Shooting, Stealing, Shoplifting, and Vandalism. The UCF-Anomaly-Detection-Dataset collected a 950 unedited real-world surveillance videos with clear anomalies as well as a 950 normal videos. The UCF dataset provides only video-level annotations, however, to train the system, we need segment-video level labels. For that purpose, we used an external expert system for video segment evaluation. In remainder of this section, we will describe the methods and the setup used for configuring and evaluating the learning video surveillance agent, and expose the details of the experiment results. Additionally, we compared our approach with state-of-the-art video anomaly detection.

4.1 Hyper-parameters

In our experiments, the preprocessing of the video data is made through the extraction of the visual features from the fully connected (FC) layer FC6 of the C3D network provided by authors of [17]. Before computing features, we resize each video frame to 240x320 pixels and fix the frame rate to 30 fps. We compute C3D features for every 16-frame video clip followed by l2 normalization. The agent's network is implemented with a fully connected Feedforward neural network setup by the network configuration described in Table 2. The network includes 3 layers. The ReLU function [19] is used for the two first layers, the output layer takes the sigmoid function for activation to build the output.

Layer N	1^{st} layer	2^{nd} layer	Output layer
Type	Dense	Dense	Dense
Unit size	512	256	64
Activation	ReLU	ReLU	Sigmoid
Weight Regularizer	decay $l2(0.001)$	decay $l2(0.001)$	decay $l2(0.001)$

Table 2: The network configuration.

After splitting every video into 32 non-overlapping segments, the agent starts learning by playing one episode per video. During an episode, the agent is allowed to play a number of steps to find the segments that cover the anomaly to be reported. The number value of steps depends on the annotation of the processed video, if it is annotated as normal, the agent has 32 steps t to remove all segments with a low abnormality score, otherwise, the segments are removed as there is no anomaly. In the case of an episode of a video annotated as abnormal, the number 5 has been set as a minimum of segments per state s_t, so the agent has 27 chances to remove the segments with the weakest anomaly score, in other words, it keeps the segment with high predicted anomaly score. We built our agent on 1600 episodes. Training was performed on a 4GB NVIDIA GeForce RTX 2070 SUPER GPU. The methods are implemented using Python with the help of Keras. The training process works under Adagrad optimizer algorithm [20] with MSE loss function and a learning rate with a value of 0.01, it's remaining parameters are set as default.

4.2 Results and analysis

Firstly, we study the sample results shown in Figures 6 and 7. Figure 6 represent a successful abnormality detection on videos containing anomalous events. The localization of anomalies is highlighted by the red frames, corresponding to the highest predicted anomaly score, and green frames highlighted segments corresponding to anomaly score approaching zero. The figure 7 represents a successful abnormality detection in videos with normal events, all frames in the video are shown in green. Meaning that all segments in the video have a low anomaly score.

The false alarms are considered as the weak point of an artificial video surveillance system. Based on the observations, we have made several attempts to reduce the false anomaly detection in the system. During the first evaluation, we noticed a large number of false negative cases, then we deduced that this was due to the default value of the number of minimum final segments (equal to 5). So, we rebuilt our model based on a new criterion, which is the final minimum number of segments will be decide by the expert E_c. In others words, for a given video that contains an anomaly, the stop criterion of the episode is the number of segments whose anomaly value judged by the expert is higher than a given threshold. Across many evaluations, the threshold value with the best results are 3.2.

We also observed a high score of false alarms in case of sudden people assembly as it happened with our principal chosen expert E_c, to reduce this phenomenon, we defined this case as a very important experience through prioritized experience replay mechanism. We managed to reduce the error score by up to 60.1%. However, the system failed in many cases of very crowded scenes.

The goal was to completely automate the video surveillance system. However, there were some false alarms, so we decided to set criteria to trigger the alarm (such as calling the police, or locking all the doors, etc). Therefore, the predicted anomaly score should be greater than a threshold to trigger the alarm automatically. Otherwise, we propose to send the video segment to a human assistant to take the final decision.

4.3 Comparison with SOTA methods

Table 3 summarizes the comparison of the proposed approach with the existing state-of-the-art methods using two different datasets. For the case of UCSD-dataset, our approach demonstrates inferior performance compared to the methods including Qiao et al [7], Khaleghi et al[8], Shean Chong et al[9], Hasan et al[10]. On the other side, the proposed system produces superior performance compared to the machine learning-based methods as Lu et al[21].

Approach	References	UCSD Dataset	UCF Dataset
Machine Learning	[21]	63.8%	54.3%
Deep Learning	[7]	98.33%	-
	[8]	88.1%	-
	[9]	89.9%	-
	[10]	90.0%	65.5%
	[11]	-	75.41%
Proposed system	#	87.44%	83.12%

Table 3: AUC comparison of the proposed system with SOTA baseline models on both UCSD dataset and UCF-Anomaly-Detection-Dataset using machine learning and deep learning methods.

As far as we know, deep learning is a dominant source nowadays due to achieving high performance in many fields. the table shows that deep learning -based methods achieved better results than our approach. The obtained performance is due to the volume of data which is considered insufficient as it includes only 50 video samples for

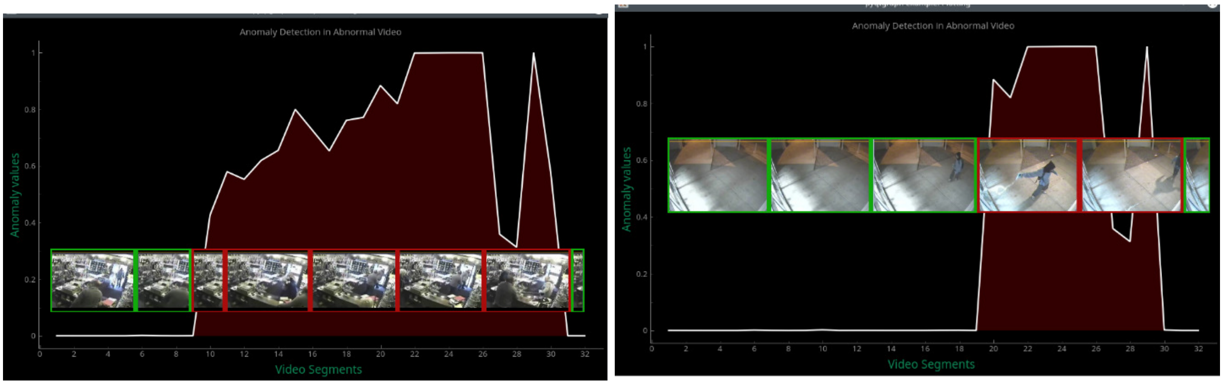

Figure 6: Examples of anomaly detection in Abnormal Videos.

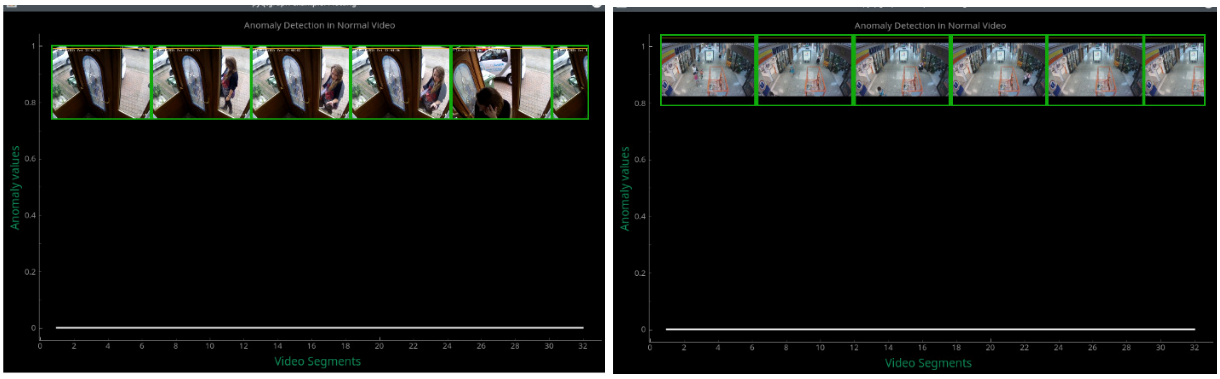

Figure 7: Example of anomaly detection in Normal Videos.

training, we assume that it surely affects the agent's learning process.

For UCF-Anomaly-Detection-Dataset that is considerably larger than the previous one, our system outperforms the deep learning and machine learning techniques. We conclude that the proposed system requires a big amount of training data for optimal performance.

[11] provides a framework to detect suspicious events in video surveillance by combining the two techniques of multiple instance learning with deep learning, resulting in an anomaly scores for each video.

Expert system E_c	Basic results	Proposed model results
Deep MIL[11]	75.41 %	83.12 %
Dictionary[21]	54.3 %	65.2 %
Deep auto-encoder[10]	65.5 %	71.09 %

Table 4: AUC comparison of multiples methods used as Expert system.

Our agent learned faster since in [11], the system started to predict the right anomalous score after 3000 iterations, but in our system, it was just after 1450 iterations. it is due to the fact that we have strengthened the learning phase with segment-level labels and the increasing of the exploration time. We were also able to surpass the compared method [11] in the reduction of false alarms as it generates a score of 1.9 for normal videos versus a score of 1.02 generated by our system. Additionally, by comparing our anomaly detection approach to two anomaly detection models: dictionary based approach [21] and deep auto-encoder based approach [10], the settings used for the comparison of both models are exactly the same ones set by [11].

Table 4 shows the comparison results for [11, 21, 10] frameworks, while simultaneously using the said approaches as the anomaly detection expert E_c, as well as a provider of the video-segments level annotations.

5 Conclusion

In this paper, an automatic video surveillance system including an anomaly detection based on deep reinforcement learning technique is proposed. In order to accelerate the agent's learning process and achieve a higher accuracy, this approach is relying not only on the annotations of the videos-level, but also on a video segment-level score provided by an expert system.

The system is trained on a variety of real-world anomalies to make it as efficient as possible in real life situations. The described method has achieved a very competitive performance that has surpassed some expert performances. Based on those results, we concluded that segment level annotations would greatly increase the system's performance if the annotations were done by a humans.

We employed many techniques of reinforcement learning such as prioritized replay and dueling architecture, though, there are still more recent improvements such as Rainbow model or NROWAN-DQN for network noise reduction.

References

[1] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. *IEEE Trans*, Pattern Analysis and Machine Intelligence, 2013. https://doi.org/10.1109/tpami.2012.59.

[2] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. *NIPS*, 2014.

[3] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. *CVPR*, 2014. https://doi.org/10.1109/cvpr.2014.223.

[4] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A ¨local svm approach. in IEEE ICPR, 2004. https://doi.org/10.1109/icpr.2004.1334462.

[5] M. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. *ICCV*, pp. 1593–1600, 2009. https://doi.org/10.1109/iccv.2009.5459361.

[6] W. Hu, D. Xie, Z. Fu, W. Zeng, and S. Maybank. Semantic-based surveillance video retrieval Image Processing. *IEEE Transactions*, vol. 16, no. 4, pp. 1168–1181, 2007. https://doi.org/10.1109/tip.2006.891352.

[7] Meina Qiao, Tian Wang, Jiakun Li, Ce Li, Zhiwei Lin, and Hichem Snoussi. Abnormal event detection based on deep autoencoder fusing optical flow. *Control Conference (CCC) 36th*, IEEE, Chinese, pages 11098-11103, 2017. https://doi.org/10.23919/chicc.2017.8029129.

[8] Ali Khaleghi and Mohammad Shahram Moin. Improved anomaly detection in surveillance videos based on a deep learning method. *2018 8th Conference of AI & Robotics and 10th RoboCup Iranopen International Symposium (IRANOPEN)*, IEEE, pages 73–81, 2018. https://doi.org/10.1109/rios.2018.8406634.

[9] Yong Shean Chong and Yong Haur Tay. Abnormal event detection in videos using spatiotemporal autoencoder. *International Symposium on Neural Networks*, Springer, pages 189–196, 2017. https://doi.org/10.1109/ascc.2015.7244871.

[10] M. Hasan, J. Choi, J. Neumann, A. K. Roy-Chowdhury, and L. S. Davis. Learning temporal regularity in video sequences. *CVPR*, 2016. https://doi.org/10.1109/cvpr.2016.86.

[11] Waqas Sultani, Chen Chen, and Mubarak Shah. Real-world anomaly detection in surveillance videos. *Center for Research in Computer Vision (CRCV)*, 2018. https://doi.org/10.1109/cvpr.2018.00678.

[12] Min-hwan Oh, Garud Iyengar. Sequential Anomaly Detection using Inverse Reinforcement Learning. *arXiv:2004.10398v1 [cs.LG]*, 2020.

[13] Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot and Nando de Freitas. Dueling Network Architectures for Deep Reinforcement Learning, *arXiv:1511.06581v3[cs.LG]*, 2016.

[14] V. Mnih, K. Kavukcuoglu, D. Silver, A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning, *Nature*, 518(7540):529–533, 2015. https://doi.org/10.1038/nature14236.

[15] Mnih, V. Kavukcuoglu, K. Silver, D. Graves, A. Antonoglou, I. Wierstra, D. Riedmiller, M. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

[16] X. Lan, H. Wang, S. Gong, and X Zhu. Deep reinforcement learning attention selection for person re-identification. *BMVC*, 2017. https://doi.org/10.5244/c.31.121.

[17] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. *ICCV*, 2015. https://doi.org/10.1109/iccv.2015.510.

[18] Tom Schaul, John Quan, Ioannis Antonoglou and David Silver. Prioritizied Experience Replay. *arXiv:1511.05952v4 [cs.LG]*, 2016.

[19] Xavier Glorot, Antoine Bordes, Yoshua Bengio.Deep Sparse Rectifier Neural Networks. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*. PMLR 15:315-323, 2011.

[20] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res*, 2011.

[21] C. Lu, J. Shi, and J. Jia. Abnormal event detection at 150 fps in matlab. *ICCV*, 2013. https://doi.org/10.1109/iccv.2013.338.