Anomaly Detection In Videos Using Deep Learning

Ashwini Jadhav[1], M. U. Inamdar[2]
[1] Dept of E&TC
[2]Assistant Professor, Dept of E&TC
[1, 2] Siddhant College of Engineering, Pune

Abstract- *Efficient anomaly detection in surveillance videos across diverse environments represents a major challenge in Computer Vision. Analysis of the information captured using these cameras can play effective roles in event prediction, online monitoring and goal-driven analysis applications including anomalies and intrusion detection. Nowadays, various Artificial Intelligence techniques have been used to detect anomalies, amongst them convolutional neural networks using deep learning techniques improved the detection accuracy significantly. This work proposes a background subtraction approach based on the recent deep learning technique of residual neural networks capable of detecting multiple objects. The goal of this article is to propose a new method based on deep learning techniques for anomaly detection in video surveillance cameras. The proposed method has been evaluated in the UCSD dataset, and showed an increase in the accuracy of the anomaly detection.*

Keywords- Deep Learning, Anomaly Detection, surveillance cameras

I. INTRODUCTION

Surveillance cameras are increasingly being used in public places e.g. streets, intersections, banks, shopping malls etc. to increase public safety. However, the monitoring capability of law enforcement agencies has not kept pace. The result is that there is a glaring deficiency in the utilization of surveillance cameras and an unworkable ratio of cameras to human monitors. One critical task in video surveillance is detecting anomalous events such as traffic accidents, crimes or illegal activities.

Anomaly detection with the use of unsupervised machine learning techniques is still an open debate in the field of machine learning. Anomaly means the occurrence of events or behaviors which are unusual, irregular, unexpected and unpredictable and thus different from existing patterns [1]. Detecting anomalies by learning from normal data can have important and different applications [2]. And also, an anomaly detection process is completely dependent on the environment, context and anomaly scenario [3, 4]. In different scenarios, anomalies will accordingly be different [1]. Existing supervised methods for anomaly detection such as simple CNN based methods require labels which are difficult to attain due to the video high dimension information. High dimension of video affects representation and creation of a model [5]. In this work, anomaly detection is based on videos of surveillance cameras. It should be noted that detection in videos is more difficult than in other data since it involves detection methods and also requires video processing as well [6].

The processing of surveillance cameras information in crowded scenes poses serious challenges and difficulties. If this process is online, the complexity will even increase. One of the best approaches for processing this information and consequently achieving the goal-oriented pattern is the use of advanced machine learning techniques such as deep learning approaches. The advantage of these types of processes, which usually have a high dimensional data, can be traced back to the existence of an end-to-end system. End-to-end systems automate feature extraction [7]. One of the main purpose of using deep learning is to extract information from high dimension data [8].

This work introduces an anomaly detection method based on deep learning techniques. The architecture of this method has two main phases which are called train network and detection classifier. The first phase aims for feature extraction and is consisted of five components with a deep structure. The aim of the second phase is detection. This phase is consisted of five deep neural network classifiers and reconstruction network. Each component in detection phase produces a detected class and a score. At last, by these detection classes and scores, the ensemble classifier performs the final detection and announces it.

The main contribution of this work is the use of deep learning techniques in all phases of anomaly detection. In other sections of this work, at first, a brief description and background of video anomaly detection based on deep learning methods is provided, at section II, related work is presented, in section III, our proposed new method is described in detail and in the final section evaluations are conducted to demonstrate improvements and advantages of the proposed method in comparison with previous methods.

II. BACKGROUND

Due to the existence of rich and analytical information in videos and their easy accessibility, scientific researchers have been interested in the analysis and processing of these kinds of data. One of the challenges in analyzing video data is objects detection in video frames [9]. Also, video anomaly detection has been one of the controversial research topics within the recent years. In the last few years, deep learning approaches have also been introduced for the implementation of anomaly detection methods. In all anomaly detection approaches, learning is achieved solely through normal data. Another important point regarding the anomalies is that abnormal events are usually rare events that occur comparatively less than other normal incidents [2].

The challenges for detecting anomalies in videos include speed, online alerts, and localization. It should be mentioned that anomaly localization is very crucial and most of the existing systems and data lack it. In some approaches, the localization is performed in the pre-processing step which is usually based on video frames comparison. This will increase the accuracy [10, 11]. In other words, most of the existing approaches and available datasets only indicate the presence of anomalies and do not specifying their location [12]. The current methods also lack appropriate training data and correct anomaly description along with their high cost of extracting features which directly affect detection [6].

One of the widely used methods for detecting anomalies is the use of a binary classifier which has two normal and abnormal classes. The normal class contains data whose occurrence frequency is high, while the other class contains rare and unseen events in accordance with the data pattern [2].

III. PROPOSED METHODOLOGY

The proposed method of this work is based on deep learning techniques for detecting anomalies in video. Two main components are considered for this method. The first component is the extraction and learning of the feature and the second component is the detection of anomalies. Apart from these two components, there is a pre-processing step which is related to background estimation and removal. Like all machine learning approaches, this method also has two main train phase and test phase. In train phase, features are trained by train parts of dataset which contains only normal frames, and trained model in test phase is used by other parts of dataset which contain abnormal frames. Figure 3 illustrates the overall framework of the proposed method.

As can be seen in the figure, learning features are of four main types. For some types, feature extraction processes are performed on single frames, and others are based on patch frames in order to reduce cost and training time. The first feature is appearance which is related to object detection in each frame; and by comparing each frame with previous and next frames the detection score is generated. The second feature is density which is about density of objects in each frame; the final score is generated based on frames comparison and average speed.

The third feature is motion which is based on the flow of objects between patch frames and it generates optical flow and a sequence of video then used for another score on anomaly. The last feature is scene which is based on patch frames and reconstructing a scene from learned model. The combination of these features is also used for detection and creation of scores.

3.1 Pre-Processing

The first step before starting extracting and learning features is to estimate and remove the background. The background is indeed different for different scenarios as there are various methods for its removal. For instance, the background might include empty spaces or street borders. In this method, the background estimation is based on most occurrence of frequency (MOF) between video frame patches. For the background estimation steps at first, a histogram is generated for each frame of the video which is based on pixels and their location in the image. Then the histogram of the frames in each patch is compared with each other, and the maximum values per patch are identified as background and are thus grayed. Removing the background will reduce the cost of the computing and the processing time. This step is considered as a part of train network.

3.2 Feature Extraction and Learning Component

In addition to background estimation, train network has four main components. The deep network for extracting appearance feature uses a stacked denoising auto-encoder (SDAE) with 6 encode layer and the same structure of decode layer [17, 23]. Each frame is convolving to network with 1*1 window size and it includes stride and padding. All frames normalize in binary mode. This SDAE has 6 encode layers and 6 same structure in decode layer which is deeper than the existing methods. The output of this step is detected objects which are called appearance representation. This output is used in detecting phase and also is utilized as an input to density estimation component in order to increase the accuracy of estimation.

Density Estimation [25] is carried out by convolutional neural network with 8 * 8. Windows filter. The third component is motion feature extractor [17, 23]. It performs a feature extraction based on the direction of moving objects in the scene of video patches. This deep network also has a similar structure to appearance feature extractor but it is based on frames patches. After entering the patch frame into the network, computing optical flow will be done based on comparison of frames in a patch. The output of this step is Motion Representation which is used for future detection.

The last component is Scene Reconstruction which is based on reconstruction network [26]. The structure of this reconstruction network is based on convolutional Auto-Encoder with the same CNN generator and discriminator networks. Generator part regenerate the scene which has 10 layers to reconstruct frames based on the previous and the next Frame in same patch and the discriminator compares the generated scene with original one in order to compute the reconstruction error. It should be mentioned that discriminator part has the same structure as that of the generator. A high reconstruction error during test indicates anomalies. The reconstruction error in train network is low and this will be a measure for detecting anomalies. At the end of the training step, a set of learned and combined features is created in order to achieve anomaly detection.

In the detection component, learned features which are generated in train network are given to a classifier with two classes of normal and abnormal. Features are given as individual and combined feature to these networks. Reconstruction error and appearance features are given to network as a combined feature since the appearance feature or object detection with a reconstruction error can be a strong feature for the detection of anomalies. The lower reconstruction error for the corresponding frame will make the detection more accurate.

Two other combination features are Motion Feature and density map. These are two complementary features and the direction of motion must be equal to the transfer of density direction.

The classifiers used in this method are simple deep classifiers which used the softmax function. As can be seen in Figure 4, five classifiers with the same structure are used in the detection step. There are 5 hidden layers in these networks in order to reduce the computing cost overhead. The last layer of these networks is fully connected. Each of these classifiers finally detects anomaly or normal situation and produces a score for the percentage of anomalies presence. This score ranges between [0 – 1].

In addition to the classifiers, there is also an Auto-Encoder reconstruction network which has the same structure as the Auto-Encoder in train network. But it is pre-trained and no generator exists in this component. This network uses the previous generator. A comparison is made between the test data and the trained network and according to the difference between reconstruction error in train and discriminator the score and the result of the detection of the anomaly are determined.

The last component is final decision-making (ensemble) which determines the final detection result. This classifier is a simple linear classifier that declares the final result based on the percentage of votes and the score of other classifiers. The structure of this component is defined in a way that if four out of six classifiers vote for anomalies, the detection is declared as anomaly and the score is announced as the average of other classifier scores.

3.3 Convolutional Neural Network

In machine learning, a network (CNN, or ConvNet) is a class of deep, feed-forward artificial neural networks, most commonly applied to analyzing visual imagery. CNNs use a variation of multilayer perceptrons designed to require minimal preprocessing.[1] They are also known as shift invariant or space invariant artificial neural networks (SIANN), based on their shared-weights architecture and translation invariance characteristics. Convolutional networks were inspired by biological processes[4] in that the connectivity pattern between neuronsresembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. CNNs use relatively little pre-processing compared to other image classification algorithms. This means that the network learns the filters that in traditional algorithms were hand-engineered. This independence from prior knowledge and human effort in feature design is a major advantage.

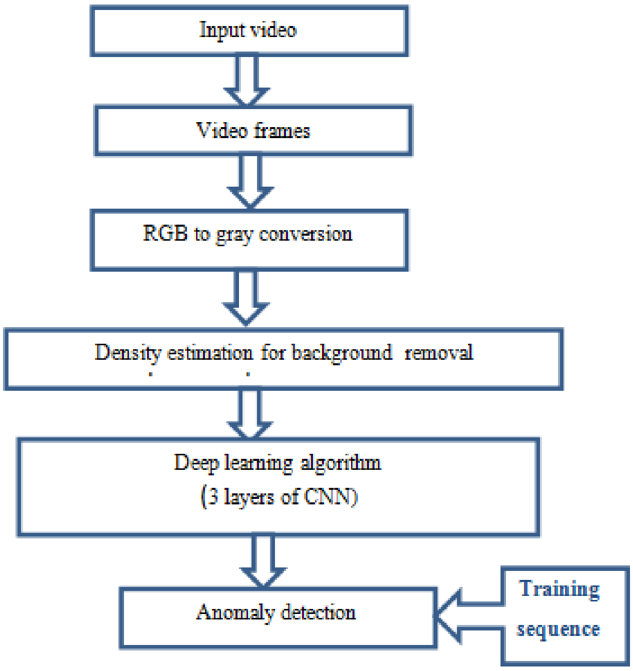


Figure 1. Flow diagram of proposed system

They have applications in image and video recognition, recommender systems and natural language processing. A CNN consists of an input and an output layer, as well as multiple hidden layers. The hidden layers of a CNN typically consist of convolutional layers, pooling layers, fully connected layers and normalization layers. Description of the process as a convolution in neural networks is by convention. Mathematically it is a cross-correlation rather than a convolution. This only has significance for the indices in the matrix, and thus which weights are placed at which index.

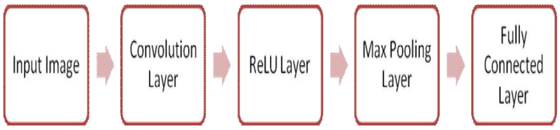


Fig 2 CNN architecture

3.3.1 Convolutional Layer

Convolutional layers apply a convolution operation to the input, passing the result to the next layer. The convolution emulates the response of an individual neuron to visual stimuli.

Each convolutional neuron processes data only for its receptive field. Although fully connected feedforward neural networks can be used to learn features as well as classify data, it is not practical to apply this architecture to images. A very high number of neurons would be necessary, even in a shallow (opposite of deep) architecture, due to the very large input sizes associated with images, where each pixel is a relevant variable. For instance, a fully connected layer for a (small) image of size 100 x 100 has 10000 weights for *each* neuron in the second layer. The convolution operation brings a solution to this problem as it reduces the number of free parameters, allowing the network to be deeper with fewer parameters. For instance, regardless of image size, tiling regions of size 5 x 5, each with the same shared weights, requires only 25 learnable parameters. In this way, it resolves the vanishing or exploding gradients problem in training traditional multi-layer neural networks with many layers by using backpropagation

3.3.2 ReLU Layer

ReLU is the abbreviation of Rectified Linear Units. This layer applies the non-saturating activation function f(x)=max(0,x). It increases the nonlinear properties of the decision function and of the overall network without affecting the receptive fields of the convolution layer.

3.3.3 Pooling

Convolutional networks may include local or global pooling layers[which combine the outputs of neuron clusters at one layer into a single neuron in the next layer. For example, *max pooling* uses the maximum value from each of a cluster of neurons at the prior layer. Another example is *average pooling*, which uses the average value from each of a cluster of neurons at the prior layer.

Another important concept of CNNs is pooling, which is a form of non-linear down-sampling. There are several non-linear functions to implement pooling among which *max pooling* is the most common. It partitions the input image into a set of non-overlapping rectangles and, for each such sub-region, outputs the maximum. The intuition is that the exact location of a feature is less important than its rough location relative to other features. The pooling layer serves to progressively reduce the spatial size of the representation, to reduce the number of parameters and amount of computation in the network, and hence to also control overfitting. It is common to periodically insert a pooling layer between successive convolutional layers in a CNN architecture. The pooling operation provides another form of translation invariance.

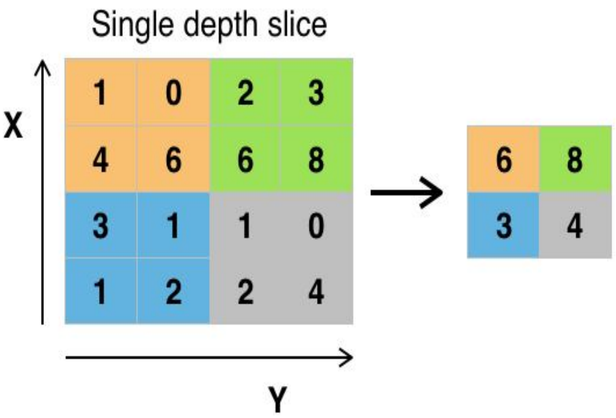


Fig 3. Maximum Pooling

The pooling layer operates independently on every depth slice of the input and resizes it spatially. The most common form is a pooling layer with filters of size 2x2 applied with a stride of 2 downsamples at every depth slice in the input by 2 along both width and height, discarding 75% of the activations. In this case, every max operation is over 4 numbers. The depth dimension remains unchanged.

In addition to max pooling, the pooling units can use other functions, such as average pooling or L2-norm pooling. Average pooling was often used historically but has recently fallen out of favor compared to max pooling, which works better in practice.

Due to the aggressive reduction in the size of the representation, the trend is towards using smaller filters[42] or discarding the pooling layer altogether

3.3.4 Fully connected

Fully connected layers connect every neuron in one layer to every neuron in another layer. It is in principle the same as the traditional multi-layer perceptron neural network.

In neural networks, each neuron receives input from some number of locations in the previous layer. In a fully connected layer, each neuron receives input from *every* element of the previous layer. In a convolutional layer, neurons receive input from only a restricted subarea of the previous layer. Typically the subarea is of a square shape (e.g., size 5 by 5). The input area of a neuron is called its *receptive field*. So, in a fully connected layer, the receptive field is the entire previous layer. In a convolutional layer, the receptive area is smaller than the entire previous layer.

Each neuron in a neural network computes an output value by applying some function to the input values coming from the receptive field in the previous layer. The function that is applied to the input values is specified by a vector of weights and a bias (typically real numbers). Learning in a neural network progresses by making incremental adjustments to the biases and weights. The vector of weights and the bias are called a *filter* and represents some feature of the input (e.g., a particular shape). A distinguishing feature of CNNs is that many neurons share the same filter. This reduces memory footprint because a single bias and a single vector of weights is used across all receptive fields sharing that filter, rather than each receptive field having its own bias and vector of weights.

IV. SYSTEM IMPLEMENTATION

For the simulation of system implementation MATLAB software have been used. MATLAB is higher level technical and scientific programming language which is capable of visualization and faster computation. Using MATLAB data can be analyzed, processed, synthesized and visualized.

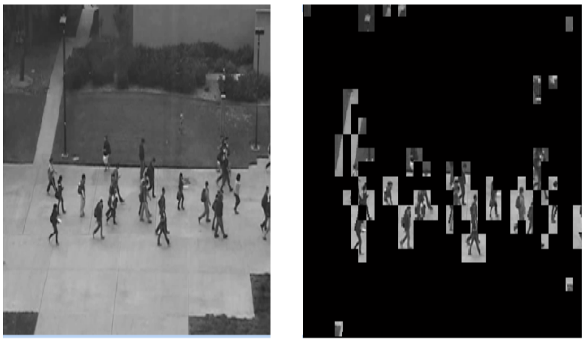
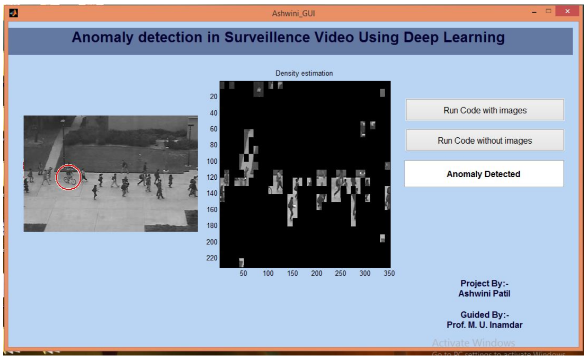


Figure 4. Density Estimation for dynamic data in video



Ashwini_GUI

Anomaly detection in Surveillence Video Using Deep Learning

Density estimation

Project By:-
Ashwini Patil

Guided By:-
Prof. M. U. Inamdar

Figure 5. Anomaly Detected in Video

Table 1. Cross validation accuracy for UCSD Database

UCSD database	% Cross Validation Accurac
Normal Activity	92.00 %
Anomaly Activity	91.50 %

V. CONCLUSION

In this work, a new deep learning based for anomaly detection of video surveillance cameras is introduced. One advantage of this method is the use of deep learning techniques in all train and detection components. The two main components of this method are evaluated based on some metrics and with UCSD dataset which is the most famous anomaly detection dataset. Another benefit of this method is the isolation of train network phase. So it can use as a pre-train network in similar works.

For further improvement, it is possible to add a component which can add descriptions to each detection classifier or to the last one; or it is possible to add a component in the detection phase which can localize the anomaly accurately.

REFERENCES

[1] Dinesh Kumar Saini, Dikshika Ahir and Amit Ganatra, “Techniques and Challenges in Building Intelligent Systems: Anomaly Detection in Camera Surveillance”, Proceedings of First International Conference on Information and Communication Technology for Intelligent Systems, Springer International Publishing Switzerland, 2016

[2] B Ravi Kiran, Dilip Mathew Thomas, Ranjith Parakkal, “An overview of deep learning based methods for unsupervised and semi-supervised anomaly detection in videos”, MDPI Journal of Imaging, arXiv:1801.03149v1, 2018

[3] yota Hinami, Tao Mei, and Shin’ichi Satoh, “Joint Detection and Recounting of Abnormal Events by Learning Deep Generic Knowledge”, arXiv:1709.09121v1, 2017

[4] M. Ribeiro, A.E.L., and H. S. Lopes, “A study of deep convolutional auto-encoders for anomaly detection in videos”, Pattern Recognition Letters, ELSEVIER, 2017

[5] Yong Shean Chong, Yong Haur Tay, “Abnormal Event Detection in Videos Using Spatiotemporal Autoencoder”, International Symposium on Neural Networks, Springer International Publishing AG, 2017

[6] Hung Vu, “Deep Abnormality Detection in Video Data”, Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017

[7] M. Sabokroua, M.F., M. Fathyc, Z. Moayedd and R. Kletted, “Deep-Anomaly: Fully Convolutional Neural Network for Fast Anomaly Detection in Crowded Scenes”, Journal of Computer Vision and Image Understanding, 2017

[8] Qaisar Abbas, Mostafa E. A. Ibrahim, M. Arfan Jaffar1, “Video scene analysis: an overview and challenges on deep learning algorithms”, Multimedia Tools and Applications, Springer, 2017

[9] Revathi, A. R., Kumar, Dhananjay “An efficient system for anomaly detection using deep learning classifier”, Signal, Image and Video Processing, Springer, 2016

[10] Hung Vu, Tu Dinh Nguyen, Anthony Travers, Svetha Venkatesh andDinh Phung, “Anthony Travers, Energy-Based Localized Anomaly Detection in Video Surveillance”, Springer International Publishing AG, 2017

[11] Siqi Wanga, E.Z., Jianping Yin, “Video anomaly detection and localization by local motion based joint video representation and OCELM”, Neurocomputing, 2017

[12] M. Sabokrou, M. Fathy, M. Hoseini., “Video anomaly detection and localisation based on the sparsity and reconstruction error of auto-encoder”, ELECTRONICS LETTERS, IEEE, 2016.

[13] Yong Shean Chong, Yong Haur Tay, “Modeling Video-based Anomaly Detection using Deep Architectures: Challenges and Possibilities”, Control Conference (ASCC), IEEE, 2015

[14] Shean Chong, Yong Haur Tay, Yong, “Modeling Representation of Videos for Anomaly Detection using Deep Learning: A Review”, arXiv:1505.00523v1, 2015.

[15] Muhammad Umer Farooq, Najeed Ahmed Khan and Mir Shabbar Ali, “Unsupervised Video Surveillance for Anomaly Detection of Street Traffic” International Journal of Advanced Computer Science and Applications(IJACSA), 2017

[16] Robert P. Loce , Raja Bala, Mohan Trivedi, “Computer Vision and Imaging in Intelligent Transportation Systems”, John Wiley & Sons Ltd. , 2017

[17] Dan Xu, E.R., Yan Yan, Jingkuan Song, Nicu Sebe, “Learning Deep Representations of Appearance and Motion for Anomalous Event Detection”, arXiv:1510.01553, 2015

[18] Sorina Smeureanu, Radu Tudor Ionescu, “Deep Appearance Features for Abnormal Behavior Detection in Video”, Springer ICIAP, 2017

[19] Medhini G. Narasimhan, S.K.S., “Dynamic video anomaly detection and localization using sparse denoising autoencoders”, Multimedia Tools and Applications, Springer, 2017

[20] Tianlong Bao, C.D., Saleem Karmoshi, Ming Zhu, “Video Anomaly Detection Based on Adaptive Multiple Auto-Encoders”, Springer International Publishing, 2016

[21] Yachuang Feng, Y.Y., Xiaoqiang Lu, “Learning Deep Event Models for Crowd Anomaly Detection”, Elsevier Neurocomputing, 2016

[22] Jiayu Sun, J.S., “Abnormal event detection for video surveillance using deep one-class learning”, Multimedia Tools and Applications, Springer, 2017

[23] Dan Xu, Y.Y., Elisa Ricci, Nicu Sebe, “Detecting Anomalous Events in Videos by Learning Deep Representations of Appearance and Motion”, Computer Vision and Image Understanding, 2016

[24] Bo Zong, Q.S., Martin Renqiang Min, Wei Cheng, Cristian Lumezanu, Daeki Cho, and Haifeng Chen, “Deep Autoencoding Gaussian Mixture Model For Unsupervised Anomaly Detection”, Iclr, 2018

[25] Yingying Zhang, D.Z., Siqin Chen, Shenghua Gao, Yi Ma, “Single-Image Crowd Counting via Multi-Column Convolutional Neural Network”, IEEE Conference on Computer Vision and Pattern Recognition, 2016.

[26] Antonia Creswell, T.W., “Generative Adversarial Networks: An Overview”, IEEE-SPM, 2017