Grenze International Journal of Engineering and Technology, Jan Issue

Review of Various Object Detection and Anomaly Detection Techniques

Amrutasagar Kavarthapu[1], Dr. Pokkuluri Kiran Sree[2] and Dr. D Rajya Lakshmi[3]
[1]Seshadri Rao Gudlavalleru Engineering College/CSE, Gudlavalleru, India
Email: sagar.chintu89@gmail.com
[2]Shri Vishnu Engineering College for Women(A)/CSE, Bhimavaram, India
Email: drkiransree@gmail.com
[3]Jawaharlal Nehru Technological University Gurajada Vizianagaram /CSE, Vizianagaram, India
Email: jrajyalashmi@gmail.com

***Abstract*— Object detection in surveillance videos is challenging task whenever the objects are similar to other objects and overlapped with each other. Sometimes objects may be masked or cover with any transparent or non transparent material. Due to coverage of objects with opaque materials, object detection becomes difficult. Once object have been identified, it is require to recognize such object leads to any anomaly in that particular environment or not. There is huge demand for intelligent systems with the capability of automatically identifying anomalous situations or objects in an environment present in videos. Due to this, a wide range of approaches have been proposed to build an effective system that would ensure public security. Intension of this article is to specify various Object detection and Anomaly detection techniques.**

***Index Terms*— Object, Frame, Same-Class, Region of Interest, Pooling, Anomaly.**

I. INTRODUCTION

A. Object Detection

Object detection in videos is a significant task, and is used in a large number of application areas, including event detection, environment of railway stations, airports and some other crowd and sensitive areas. Object detection and tracking can be achieved using different Machine Learning techniques [1]. Basically different kind of objects can be recognized as movable and non movable objects. Non movable objects with respect to the current working environment will be assumed as background and movable objects can be consider as foreground objects. Real-time videos consist of compound objects.

Deep learning-based object detectors use deep CNNs as a basis to extract features from the input video and classify the object(s) based on these features. Object detectors are basically categorized into (i) one stage object detectors [3] and (ii) two stage object detectors [2]. In one stage object detectors, bounding boxes are calculated over the input video frames without suggesting a region over the frames. In two stage object detectors, approximate regions of the object are suggested first. Then, features are mined from the proposed regions and these regions are used for classification of objects and bounding box regression for object identification. The commonly used methods for proposing regions are based on superpixel grouping [4] and sliding windows.

Grenze ID: 01.GIJET.10.1.222

B. Anomaly Detection

An event(s) deviating from usual behavior can be defined as anomalies [11], e.g., running in busy streets, pedestrians walking on busy roads, occurrence of accident on roads, usage of harmful devices by a person in a group of peoples in sensitive areas, etc. The purpose of anomaly detection is to recognize early detection of human or objects behaviours in an environment. Early detection of anomalies will helpful in preventing of crime, identifying abnormal situation and alert the peoples in the particular area. Generally less quantity of anomalous situation will occur with respect to whole environment. i.e. for example, less than 1% of anomalous situations will occur to overall actions present in the environment. So to identify such less quantity of anomalous situations with human effort require more time and effort. To identify anomalous situations in time with less effort and more accuracy, anomaly detection techniques were using in the current trend.

To work with anomaly detection mechanisms, first learn the regular patterns, and then mine representations of usual cases. If any events diverge from the usual cases representations, such event leads to abnormal. In anomaly detection mechanisms, still there are some challenges are present in video surveillance and human activity recognition. These challenges are associated to the extracting features of objects present in video, when overlapping of objects are occur in video, objects are merging with backgrounds, low visualization, will impact on performance of the a system's [12].

C. Process of anomaly detection

Thewhole process of Object Detection and Anomaly Detection will be described briefly in the following figure 1.

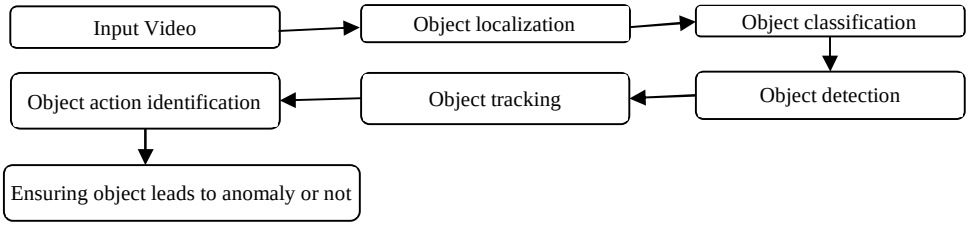

Figure 1. Process of Object Detection and Anomaly Detection in videos

First the input video is divided into different frames, from the frames you can obtain images, from the obtained images object localization is performed based of region of interest and then object are classified. Once the objects are classified, object will be detected. Now object are going to tack with respect to their motion, behaviour and appearance. After tracking of object(s) present in the video, object action will be identified. Based on action of object(s), you can predict anomaly situation will occur or not in that environment.

The remaining part of the paper present as, Section II specifies various object detection mechanisms, Section III specifies various anomaly detection mechanisms, and Section IV includes comparative studies and finally Section V gives the conclusion of the present study.

II. Object Detection Techniques

A. Region based Convolutional Neural Network

Region-based Convolutional Neural Networks (RCNN) [6] or regions with CNN features (R-CNNs) are approaches that apply deep models to object recognition. R-CNN models first select multiple proposed regions from an image (e.g., anchor boxes are one type of selection method) and then label their kinds and bounding boxes. These labels are created based on predefined classes that are given to the program and then use a convolutional neural network for forward computation to mine features from each domain.

In R-CNN [6], the input image is first divided into almost two thousand regions and then a convolutional neural network is applied to each region. The size of the regions is calculated and the correct region is inserted into the neural network. The training time is significantly higher compared to YOLO, as the bounding boxes are classified and created individually and the neural network is applied to one region at a time. R-CNN uses a selective search method to localize RoIs in the input images and uses a Deep Convolutional Neural Network based region-wise classifier to classify the RoIs independently.

B. Fast RCNN

Fast R-CNN [7] [9] is an improved version of RCNN that deals with the mining of RoIs from feature maps. It has been found to be much faster than the conventional R-CNN architecture. In 2015, Fast R-CNN was developed

with the aim of significantly reducing the training time for identification of objects in the video. The R-CNN autonomously computed the features of neural network on each of the two thousand regions of interest, where as Fast R-CNN runs the neural network once on the entire image. Fast R-CNN is very similar to the architecture of YOLO, but YOLO remains a faster substitute to Fast R-CNN due to the simplicity of the code. But YOLO comes under one stage object detector.
At the end of the network a method known as Region of Interest (ROI) with pooling was introduced. This ROI with pooling separate each region of interest from the network's output tensor, reshapes and classifies it. This mechanism makes Fast R-CNN more accurate than the original R-CNN. However, due to this recognition technique, less data input is required to train Fast R-CNN and R-CNN detectors.

C. Faster RCNN

Faster R-CNN [2] is an object recognition model that improves Fast R-CNN by using region-based object classification (ROI pooling). Faster R-CNN consists of two modules. The first module is a deep fully convolutional network that proposes regions, and the second module is the Fast R-CNN detector [9] that uses the proposed regions. The following figure 2 describes entire system is a single, unified network for object detection and it is an architectural diagram of Faster R-CNN.

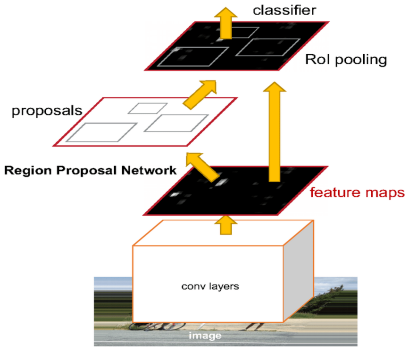

Figure 2. Faster R-CNN architecture

Faster R-CNN has been trained end-to-end by introducing RPN (Region Proposal Network). An RPN is a network that is used to generate RoIs by regressing the anchor boxes. The anchor boxes are then used for object recognition.

1. Region Proposal Networks

A Region Proposal Network (RPN) takes an image of arbitrary size as input and outputs a set of rectangular object proposals, each labelled with an objectness score. This RPN model process with a fully convolutional network [10]. Here RPN ultimate goal is to share computations with a Fast R-CNN object recognition network [9], here assumed that both networks use a common set of convolutional layers.
To generate region proposals, transfer a small network on top of the convolutional feature map output by the last common convolutional layer. This small network takes as input an n x n spatial window of the input convolutional feature map. Each sliding window is mapped to a lower-dimensional feature. This mini-network is shown in a single location shown in figure 3. Note that the mini-net works on the sliding window principle and the fully connected layers are shared at all spatial positions. This architecture is naturally implemented with an n x n convolutional layer, followed by two sibling 1 x 1 convolutional layers (for reg and cls, respectively).

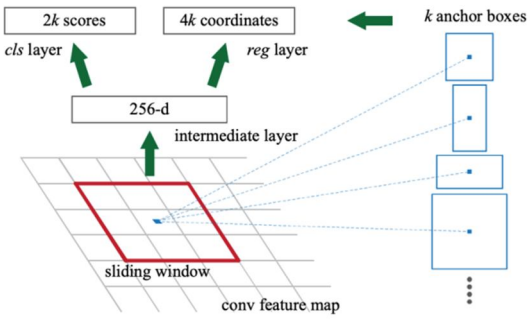

Figure 3: Architecture of Region Proposal Network

D. Granulated RCNN

Granulated RCNN (G-RCNN) [1] for multi-object detection is an improved version of the widely used Fast RCNN [7] and Faster RCNN [2], which provides better performance. Unlike Fast RCNN and Faster RCNN, G-RCNN takes the video directly as input and considers both spatial and temporal information. The inclusion of granularized layers with spatio-temporal information in the deep CNN architecture (e.g., AlexNet [3]) provides better localization of objects (RoIs). Since the classification task only refers to the objects present in the RoIs, the recognition accuracy increases considerably. In terms of speed, G-RCNN is superior to Fast RCNN, while it is comparable to Faster RCNN.

Granules means clusters that are formed when defining the RoIs via the pooling feature map instead of all feature values, here only the positive RoIs are used during training, instead of the entire RoI map. In G-RCNN only the objects in the RoIs are used instead of the entire feature map for object classification. All this feature of G-RCNN leads to an improvement in recognition speed and accuracy in real time.

Generally foreground regions that contain either single or multiple object modules, and are referred to as "granules". Processing with granules reduces the operation space (search space) for further processing and also reduces the cost complexity of video object recognition. In Granulated RCNN, the concept of optimal granulation is used in the pooling feature map.

Granulation is a process of formation and representation of granules that develops through the abstraction of information and the derivation of knowledge from data. A granule is defined as a clump of unrecognizable objects that are drawn together, for example, probability, similarity, proximity or functionality. G-RCNN has adopted the irregularly shaped neighborhood granules in the pooling feature map and uses both spatial and temporal similarity to localize the object region(s) in video frames.

G-RCNN use the commonality between the spatial and temporal maps is considered as the optimal map (called RoI map), which represents the foreground modules (i.e. object objects) in the video sequence. Now, class-specific SVMs are used to classify these object modules. After classification, these classified objects are used for tracking, which is a twofold task. First, a representation model of possible trajectories is calculated for each object recognized in a video. This model is then used in a global optimization problem to estimate the affinity between the detected objects in different frames of the same video. G-RCNN has two parts. First one Foreground Region Proposal Network (FRPN) and second one is Classification network. Architecture of G-RCNN was described in the figure 4.

1. Foreground region proposal network

Foreground region Proposal Network (FRPN) [1], a granularized deep CNN that provides RoIs for the foreground over the video frame. FRPN accepts the current video frame (f_t) with three channels (R, G and B) and its previous P frames (f_{t-1} to f_{t-P}) as inputs. The FRPN network passes these input frames through the first convolutional layer (*Conv1*) and the pooling layer (*Pool1*) to extract their reduced feature maps. Now, for each image in the Pool1 feature map in the *Granule1a* layer, a set of spatial neighborhood granules is computed and referred as step 1.

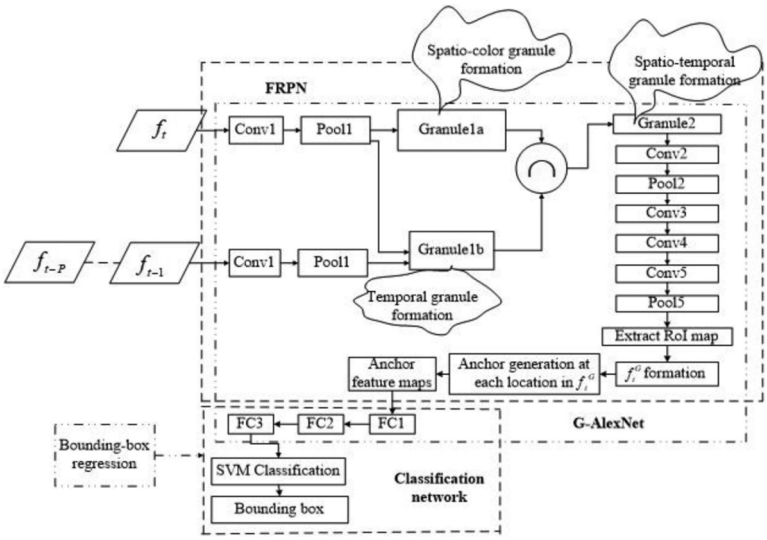

Figure 4. Architecture of G-RCNN

The network also computes temporal granules for the image sequence f_t, f_{t-1}, ..., f_{t-P} on the *Pool1* map in the *Granule1b* layer. Now these two types of granules are combined while maintaining their similarities, resulting in spatio-temporal granules are referred as step 2. This is done using the Pool1 feature map of f_t in the *Granule2* layer. This spatio-temporally granulated feature map now passes through several convolutional layers (e.g. Conv2, Conv3, Conv4 and Conv5) and pooling layers (e.g. Pool2 and Pool5) to generate the reduced map with the RoIs is referred as step 3. RoIs characterize the estimated object region(s) in f_t. The anchoring process is then performed on each RoI pixel location to obtain the approximate suggestion for the region is referred as step 4. In this method, several spatial windows with different scales and aspect ratios are drawn over each RoI pixel location. These spatial windows are called anchors.

2. Classification network

The primary focus is on categorizing objects within each Region of Interest (RoI). The classification network receives a series of anchor feature maps as input and produces a classification score and corresponding bounding box for the identified object. The classification process involves three distinct steps as Anchor classification, Fitting bounding-box and object recognition.

E. MCD-SORT

MCD-SORT is an efficient version of deep SORT [8] in terms of both accuracy and speed for tracking multiple objects, as it involves frame-by-frame association only within the same-class objects, instead of taking all classes together as one class. In order to address the aforementioned problem, a multi-class deep SORT algorithm, known as MCD-SORT [8] performs frame-by-frame association for both motion and appearance features exclusively for target objects belonging to the same class. The identification of these same-class objects is based on the granulated foreground modules (RoI map) obtained from G-RCNN. By focusing on same-class objects rather than all-class objects for frame-by-frame association, the searching space and computation time are reduced, leading to improved tracking accuracy.

MCD-SORT is specifically designed for tracking purposes. With the capability to handle occlusions of varying durations, Deep SORT proves its effectiveness in both short and long-term scenarios. The tracking algorithm MCD-SORT [8] builds upon the foundations of deep SORT [8], offering advanced features for improved performance.

The functioning of Deep SORT involves computing all possible assignments between the target object and existing track-lets in previous frames, followed by predicting the most suitable assignment for each target object. However, this approach increases the searching space and reduces tracking speed for all classes. MCD-SORT addresses this issue by limiting the computation of possible assignments only between the target object and existing track-lets of the same class. The set of trajectories containing different numbers of track-lets is denoted as $\{t : t_1, t_2, ..., t_n\}$.

III. Anomaly Detection Techniques

A. Anomaly Detection using Video-Level Labels

The components of anomaly detection using Video-Level Labels [5] of the overall framework are described below, visualized in figure 5:

1. Group of Video Fragments

The video V_i is partitioned into fragments, denoted as F(i,j), where each fragment contains k non-overlapping frames. Here, $i \in [1, n]$ represents the index of the video in a dataset of n videos, and $j \in [1, m_i]$ represents the index of m_i fragments in V_i. A single binary label, either normal (0) or anomalous (1), is assigned to each video.

2. Feature Extractor

The computation of spatio-temporal features for each fragment F(i,j) is achieved by utilizing a pre-trained feature extractor model, such as Convolution 3D (C3D) introduced in [13]. This framework is versatile and can utilize any spatio-temporal feature extractor.

3. Fully Connected Base Network

Figure 5(d) illustrates network architecture, which comprises of two FC (Fully Connected**)** layers, each accompanied by a ReLU activation function and a dropout layer. The input layer accepts a spatio-temporal feature vector, while the output layer generates an anomaly regression score ranging from 0 to 1 using a Sigmoid activation function. A score of 0 indicates a normal fragment, whereas a score of 1 indicates an anomalous fragment.

4. Clustering Based Self-Reasoning

Clustering algorithms like k-means can be utilized to divide all fragments into two clusters, considering that anomaly detection is a binary problem. These clusters are formed based on the feature representations of each fragment obtained from the output of FC-1 layer. The reason behind this choice is that FC-1 layer contains more information due to its larger size compared to FC-2. Instead of using FC-1 for pre-training, this method employs clustering for two purposes: (i) generating fragment-level pseudo annotations from video-level labels, and (ii) encouraging the network to separate both clusters in the case of an anomalous video and bring them closer in the case of a normal video.

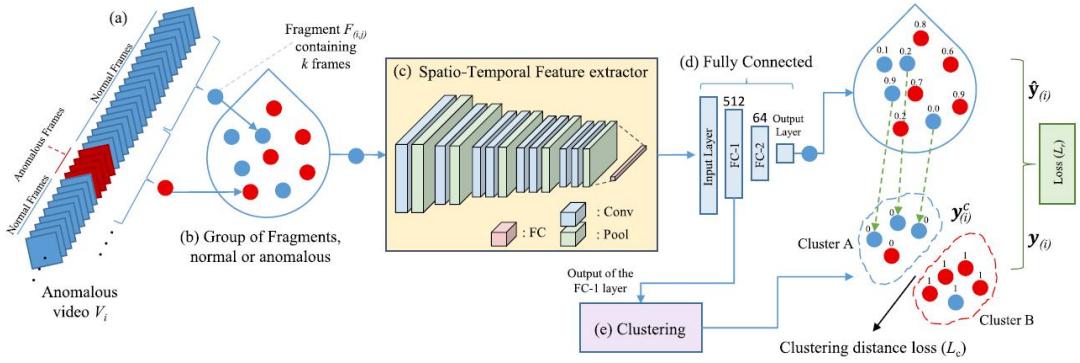

Figure 5. Architecture for anomaly detection in self-reasoning weakly supervised setting

B. Anomaly Detection with Transformers

The original proposal for the Transformer [14] focused on its application in sequence-to-sequence tasks within natural language processing (NLP), such as language translation. The key concept behind this model is the utilization of self-attention, which allows the model to effectively capture dependencies across the entire sequence. In this case, this system recognize that a video inherently represents a temporal sequence with spatial content. As a result, interpret a video as a series of tubelets and employ a Transformer encoder to capture long-term spatiotemporal dependencies. Additionally, this system incorporate a 3-D convolutional decoder to predict the subsequent frame based on the acquired spatiotemporal relations. Figure 6 provides an overview of the model.

The tokenization process in the Vision Transformer [15] involves dividing an image into smaller patches to form a sequence. However, in the case of videos, this system tokenize them by extracting non-overlapping, spatiotemporal tubes. Additionally, it includes a learnable embedding x_{cls} at the beginning of the tubelet embeddings sequence. This embedding also serves as the output feature p of the Transformer encoder. Moreover, to incorporate the original spatiotemporal position information into this model, this system introduce learnable spatiotemporal position embeddings $\boldsymbol{E}_{pos} \in R^{(N+1)\times K}$ to the tubelet embeddings.

The system utilizes a convolutional decoder to make predictions for the next frame, denoted as $\hat{I}_{i+1}$, based on the learned spatiotemporal feature p. Initially, employ two fully connected layers to enhance the dimension of p, which is then reshaped into a 3-D tensor of 8 × 8 × 512. This specific size corresponds to the number of convolutional layers in the decoder. To strike a balance between computational complexity and reconstruction accuracy, this system opt for a decoder consisting of five convolutional layers and upsampling layers. This decoder progressively reconstructs the next frame, with dimensions of 256 × 256 × 3, from the encoded feature tensor of 8 × 8 × 512.

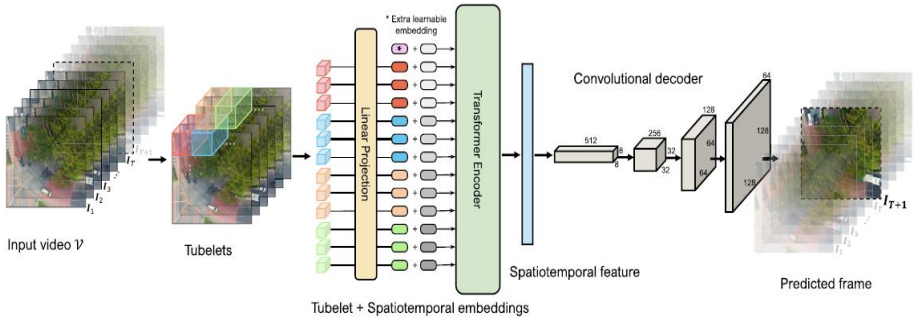

Figure 6. Overview of ANDT

C. Localizing Anomalies from Weakly-Labeled Videos (WSAL)

The initial step of this system [16] involves providing a fundamental formulation for anomaly localization. Here elaborate on the core modules of Weakly Supervised Anomaly Localization (WSAL). In this scenario, given a video sequence X and its corresponding video-level annotation y ∈ {0, 1}. Here, 'y=1' signifies the presence of an anomaly in the sequence, while 'y=0' indicates the absence of any anomaly in X. To tackle the computational burden caused by repetitive video frames, this system divide the entire video into segments of equal lengths, denoted as $X = (X_1, X_2, \bullet \bullet \bullet, X_m)$. The objective of video segmenting is to alleviate the computation load by extracting features using a classical convolutional network for each frame within the i^{th} segment X_i. By aggregating the features of all frames within the segment, here obtain the segment feature x_i. Consequently, the sequence X can be represented by the m-tuple features $(x_1, x_2, \bullet \bullet \bullet, x_m)$. Using this m-tuple, this system can now determine whether the current video contains any anomaly or not. This is achieved by assigning each segment in the video an anomaly score, which indicates the probability of it being anomalous.

A novel function is derived to formally estimate the anomalous margin within a video, enabling the prediction of its state as either normal or abnormal. By estimating the anomalous margin within a video, this system was developed a novel function that allows for the formal description of the video, enabling the prediction of its state as either normal or abnormal. Through the estimation of the anomalous margin within a video, a novel function is derived to formally describe the video, facilitating the prediction of its state as either normal or abnormal.

$$S(X) = \max_{i,j=1,\ldots,m} f(\psi(x_{i-k}, \ldots, x_i, \ldots, x_{i+k}), \psi(x_{j-k}, \ldots, x_j, \ldots, x_{j+k})), \quad (1)$$

Where ψ is a high-order function that encodes a segment with an anchor, along with its surrounding 2k segments in the temporal context.

Here the metric f calculates the margin distance, which measures the anomaly score margin between segment positions i and j. As the predicted anomaly scores become closer, the distance decreases. By utilizing the metric f, it can determine the margin distance between segment positions i and j, which measures the anomaly score margin. As the predicted anomaly scores approach each other, the distance diminishes. The anomaly score margin between segment positions i and j is quantified by the margin distance metric f. As the predicted anomaly scores converge, the distance decreases accordingly. The video score, denoted as S(•), represents the maximum relative distance between pairwise positions. It is anticipated that normal videos will have lower scores compared to anomalous videos. Consequently, the maximum-distance strategy imposes a smoother constraint on entire normal videos in contrast to anomaly videos, aligning with the conventional assumption.

D. Real-world Anomaly Detection in Surveillance Videos

The approach was depicted in Figure 7, which initiates by dividing surveillance videos into a fixed number of segments during the training process [17]. These segments serve as instances within a bag. Through the utilization of positive (anomalous) and negative (normal) bags, this system proceed to train the anomaly detection model using the proposed deep MIL ranking loss.

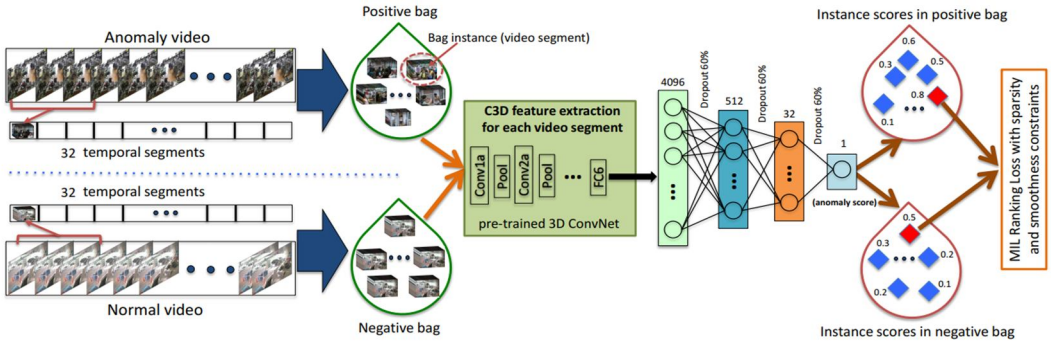

Figure 7. The flow diagram of Real-world Anomaly Detection

1. Multiple Instance Learning

When utilizing support vector machine for supervised classification, the labels for both positive and negative examples are typically provided. The classifier is then learned through an optimization function. However, Multiple Instance Learning (MIL) takes a different approach by not requiring precise temporal annotations for anomalous events in videos. Instead, only video-level labels indicating the presence or absence of an anomaly are

necessary. Videos containing anomalies are labeled as positive, while those without any anomalies are labeled as negative. Then, represent a positive video as a positive bag B_a, where individual instances in the bag are different temporal segments ($p_1, p_2, \ldots, p_m$), with m being the number of instances in the bag. It is assumed that at least one of these instances contains the anomaly. Similarly, the negative video is denoted by a negative bag, B_n, where the temporal segments in this bag form negative instances ($n_1, n_2, \ldots, n_m$). None of the instances in the negative bag contain an anomaly. Since the exact information (i.e. instance-level label) of the positive instances is unknown, the objective function can be optimized with respect to the maximum scored instance in each bag [18].

2. Deep MIL Ranking Model

Accurately defining anomalous behavior can be a challenging task, as it is highly subjective and can vary significantly from person to person. Additionally, determining how to assign labels of 1 or 0 to anomalies is not straightforward. Furthermore, due to the limited availability of sufficient anomaly examples, anomaly detection is often approached as a low likelihood pattern detection rather than a classification problem. In this approach, consider anomaly detection as a regression problem. This system goal is to assign higher anomaly scores to anomalous video segments compared to normal segments. One possible approach is to utilize a ranking loss that promotes higher scores for anomalous video segments in comparison to normal segments.

$$f(V_a) > f(V_n), \quad (2)$$

If the segment-level annotations are known during training, the ranking function mentioned above should perform effectively. In this context, V_a and V_n denote anomalous and normal video segments, while $f(V_a)$ and $f(V_n)$ represent the predicted anomaly scores for these segments, ranging from 0 to 1 respectively.

IV. Summary Of Various Object Detection Methods

The following table describe comparison of various Object detection methods with respect to various aspects.

TABLE I. COMPARISON OF VARIOUS OBJECT DETECTION METHODS

Method	Extraction	Training	Classifier
Region-based CNN (RCNN)	Computes the neural network features	Training time is significantly higher compared to YOLO	DCNN
Fast RCNN	Use RoIs from feature maps	Less data input is required to train Fast R-CNN, reduce training time	Fast R-CNN runs the neural network once for the entire image
Faster RCNN	It is a single, unified network for object detection	Trained end-to-end by introducing Region Proposal Network	Region-based object classification (ROI pooling)
Granulated RCNN	Granules are formed using pooling feature map	During training only the positive RoIs are used	Only the objects in the RoIs are used instead of the entire feature map
MCD-SORT	Involves frame-by-frame association only within the same-class objects	Focusing on same-class objects rather than all-class objects for frame-by-frame association	Same class track-lets by computing the similarity using metric Q.

V. Conclusion

Object detection in surveillance videos is tricky task whenever the objects are analogous to other objects and overlapped with each other. Sometimes objects may be masked or cover with any transparent or non transparent material. Due to coverage of objects with opaque materials, object detection becomes difficult. Each object detection techniques have their own metrics and every object detection technique will be able to identify object with respect to mirror image only. We are going to propose object detection mechanism, which will be able to detect object even in inclined direction. Similarly anomaly detection mechanism is going to propose to identify anomaly occurrence in that particular environment based on action of objects behaviour present in the environment. The entire study represents the comparison of various object and anomaly detection techniques with respect to Extraction mechanism, Classifiers used and Training mechanisms.

References

[1] Anima Pramanik et al., "Granulated RCNN and Multi-Class Deep SORT for Multi-Object Detection and Tracking", IEEE Transactions on Emerging Topics in Computational Intelligence, Vol. 6, No. 1, February 2022, pp. 171-181

[2] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks” in Proc. Adv. Neural Inf. Process. Syst., 2015, pp. 91–99
[3] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2016, pp. 779–788
[4] J. R. Uijlings, K. E. Van De Sande, T. Gevers, and A. W. Smeulders, “Selective search for object recognition” Int. J. Comput. Vis., vol. 104, no. 2, pp. 154–171, 2013
[5] Muhammad Zaigham Zaheer et al. “A Self-Reasoning Framework for Anomaly Detection using Video-Level Labels”, IEEE Signal Processing Letters, VOL. 27, 2020, pp. 1705–1709
[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2014, pp. 580–587.
[7] R. Girshick, “Fast R-CNN,” in Proc. 2015 Int. Conf. Comp. Vis., Santiago, Chile, Dec. 7–13, 2015, pp. 13–16.
[8] N.Wojke, A. Bewley, and D. Paulus, “Simple online and realtime tracking with a deep association metric” in Proc. IEEE Int. Conf. Image Process., 2017, pp. 3645–3649
[9] R. Girshick, “Fast R-CNN,” in IEEE International Conference on Computer Vision (ICCV), 2015
[10] J. Long, E. Shelhamer, and T. Darrell, “Fully convolutional networks for semantic segmentation,” in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015
[11] Chandola, V.; Banerjee, A.; Kumar, V. Anomaly detection: A survey. ACM Comput. Surv. 2009, 41, 1–58.
[12] Sabokrou, M.; Fayyaz, M.; Fathy, M.; Klette, R. Deep-cascade: Cascading 3d deep neural networks for fast anomaly detection and localization in crowded scenes. IEEE Trans. Image Process. 2017, 26, pp. 1992–2004.
[13] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, “Learning spatiotemporal features with 3D convolutional networks,” in Proc. IEEE Int. Conf. Comput. Vis., 2015, pp. 4489–4497
[14] Pu Jin et al. “Anomaly Detection in Aerial Videos with Transformers”, IEEE Transactions on Geoscience And Remote Sensing, VOL. 60, 2022.
[15] Dosovitskiy et al., “An image is worth 16×16 words: Transformers for image recognition at scale” in Proc. Int. Conf. Learn. Represent. (ICLR), 2020, pp. 1–22.
[16] Hui Lv et al. "Localizing Anomalies from Weakly-Labeled Videos”, IEEE Transactions on Image Processing, VOL. 30, 2021, pp. 4505-4515
[17] Waqas Sultani et al., “Real-world Anomaly Detection in Surveillance Videos”, Computer vision foundation, pp. 6479-6488
[18] S. Andrews, I. Tsochantaridis, and T. Hofmann. “Support vector machines for multiple-instance learning”. In NIPS, pp. 577–584, Cambridge, MA, USA, 2002. MIT Press.