Tracklet-based Explainable Video Anomaly Localization

Ashish Singh[1,2*] Michael J. Jones[2*] Erik G. Learned-Miller[1]
[1]CICS, University of Massachusetts Amherst [2]Mitsubishi Electric Research Labs
ashishsingh@cs.umass.edu mjones@merl.com elm@cs.umass.edu

Abstract

We take a scene-understanding approach to video anomaly localization (VAL) that leverages the rapid progress that has been made in training general deep networks for object detection, object recognition and optical flow. Our method uses each detected object's short-term trajectory, appearance embedding, size, and location as its representation. These high-level attributes provide rich information about the object types and movements that are found in nominal video of a scene. By efficiently comparing the high-level attributes of test objects to those of normal objects, our method detects anomalous objects and anomalous movements. In addition, the human-understandable attributes used by our method can provide intuitive explanations for its decisions. We evaluate our method on many standard VAL datasets (USCD Ped1/Ped2, CUHK Avenue, ShanghaiTech and Street Scene) using spatio-temporal evaluation criteria and demonstrate new state-of-the-art accuracy.

1. Introduction

We present a novel method for video anomaly localization (VAL) based on a high-level scene representation, allowing for intuitive explanations of decisions. Because of the practical applications to automatic monitoring of static-camera surveillance video, we focus on the single-scene, unsupervised version of VAL in which nominal video of a scene is provided to define only the normal activity in a scene. In contrast to much of the recent work in this area [16, 31, 43, 44], we focus on spatial localization of anomalies in addition to temporal localization. Most scenes include many different activities happening simultaneously and it is important to indicate to a user which of the activities within each frame is anomalous. Thus, we use the term ”localization” instead of ”detection” to emphasize our focus on both spatial and temporal localization.

Our approach is motivated from a scene understanding perspective that first tries to understand the scene in terms of the objects present and their motions/trajectories. This understanding is used to build a location-dependent model of the nominal video. Location dependence is important because activities that are normal in some locations (such as people walking on a sidewalk) may be abnormal in other locations (such as people walking on a rooftop). Our scene understanding approach yields a model that is human-interpretable which allows our system to provide human-understandable explanations for its decisions.

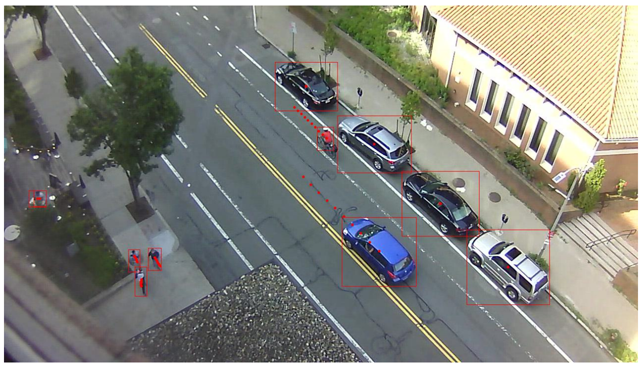
Figure 1. Visualization of all tracklets computed for a snippet of video from Street Scene.

We leverage state-of-the-art deep networks for object detection, object recognition and optical flow to yield a high-level understanding of the activity in a scene. Objects are detected using a combination of a deep network object detector and simple motion segmentation (using optical flow) which allows for the detection of unknown moving objects that the object detector was not trained on. Short-term trajectories for each detected object are then generated over T frames from optical flow to provide rich information about the movement of objects. Figure 1 shows an illustration for a frame of Street Scene [25] of the detected objects and their 10-frame trajectories which are represented by red dots indicating the location of the middle of the object's bounding box for the 10 frames. Additionally, object appearance is modeled using the feature vector embedding provided by an object recognition deep network. Putting all of these high-level attributes together yields an object-centric repre-

[*]equal contribution, work done while A.S. was an intern at MERL

sentation where, for each object, we store the object's size and location in the image, its trajectory over T frames, and its appearance embedding. We call this combination of attributes a *tracklet*. An exemplar-based model [26, 32] is then built for the nominal video of the scene which stores representative exemplar tracklets that cover the variety of normal tracklets found in the scene. Anomalous tracklets are detected by computing the distance to the nearest exemplar for each tracklet found in test video.

There are many novel aspects to our method. We present an alternative to the majority of recent approaches to video anomaly detection that rely on training a specialized deep network for each scene using a frame reconstruction or frame prediction objective function. Our method is the first to combine object appearance, size, location and trajectory into an object-centric representation that also uses a straightforward exemplar-based model which does not require any deep network training for each new scene. Because of the use of human-understandable attributes in our model, our method can provide explanations for its decisions such as "This is anomalous because it is an unusual object class with an unusual trajectory." Furthermore, we demonstrate new state-of-the-art accuracy for localizing anomalies both spatially and temporally on standard datasets.

2. Related work

The field of video anomaly detection/localization has grown rapidly. For recent surveys please see [22, 28, 42]. Much of the recent work has leveraged either the idea of using a form of autoencoder to reconstruct normal frames [3, 10, 12, 16, 18, 23] or the idea of predicting unseen normal frames from nearby ones [15, 38, 44]. While these approaches achieve good results on many standard VAD datasets, one drawback is that they do not generally lead to interpretable decisions. In contrast, our tracklet-based method provides intuitive explanations for its decisions.

The idea of using trajectories for video anomaly localization has been used in past works in ways that are distinct from our method. The early work of Stauffer and Grimson [33] modeled long-term tracks of objects and could detect anomalous trajectories as outliers from clusters of trajectories. The work of [39], [24] and [46] also focused on clustering long term trajectories and could detect outlier trajectories, but did not evaluate anomaly detection accuracy on current datasets. Short-term tracks of super-pixels along with Hidden Markov Models were used by [2] to detect video anomalies. More recent approaches [13, 21, 30] used the idea of learning to predict normal human skeleton trajectories (trajectories of human pose coordinates) to detect anomalous human skeleton trajectories. Unlike our method, these approaches are specialized to detecting only anomalous trajectories of human poses. Ours is a method for general anomaly detection.

The idea of explainable video anomaly detection has gained interest in recent years with many new papers published [4, 6, 11, 32, 36, 37, 40]. The earliest approach that provided some type of explanations was due to Hinami et al. [11]. They modeled the likelihood of high-level features computed by networks for object recognition, action recognition and attribute recognition computed from a single frame. For objects detected as anomalies, high-level attributes with the lowest likelihood were reported as the reason for the anomaly. Following this work there were a number of approaches that modeled nominal video using scene graphs [4, 6, 34]. The basic idea is to model the objects in a scene as nodes of a graph and their relationships as edges between nodes. The scene graph of [4] modeled each frame independently with no temporal or tracking information. Later, [34] improved on this by building scene graphs with temporal connections between nodes in different frames. The work of [6] did not use temporal information in their scene graphs, but added a separate branch that models the skeletal trajectories of detected people. In a similar vein, [36] proposed an explainable video anomaly localization method that used human-object interaction (HOI) vectors to model interactions between people and objects. A Gaussian mixture model is then learned from nominal HOI vectors and low probability test HOI vectors are detected as anomalous. Their method does not include any temporal modeling. Another method that provides explanations is [37]. They use a traditional frame prediction approach to detect anomalies and then in a separate stage provide explanations by extracting object and action classes for anomalous regions. Wu et al. [40] use pretrained networks to provide high-level features such as background segmentation, object detections, and tracks. Then they learn an autoencoder model to predict normal high-level feature vectors. Features that are poorly reconstructed are considered the anomaly explanations.

Singh et al. [32] present one of the few interpretable approaches to build location-specific models of nominal video which recognizes that what is normal in one region of the scene (a car driving on the street) may be anomalous in another region (a car driving on a sidewalk). Our approach is similar in some ways to theirs, but replaces their high-level motion features with object trajectories that provide richer motion information. Detecting and tracking objects removes the need for the fixed grid of spatial regions used in their work. Singh et al.'s method is like viewing the world through a small, fixed window. Fast-moving objects are only seen for a few frames before they move out of the fixed window. The size and location of each fixed window will not be appropriate for every object, resulting in only seeing parts of many objects. Furthermore, Singh et al.'s method of estimating motion features is difficult to do from raw pixels and is their main source of errors. Our approach of using tracklets, addresses all of these problems. Our method is like viewing the world through a variable-sized window (according to the object size) that moves with the object. Objects are tracked so that bounding boxes are appropriately sized and keep the object centered which results in more accurate representations of objects and their motions. We also leverage existing deep networks for object detection, object recognition and optical flow estimation instead of training specialized networks for motion features. Our approach allows future improved models to be swapped in. Our richer tracklet representation results in significantly fewer exemplars which means smaller models. Like [32], we build a location-specific model which allows us to detect anomalies involving normal activities occurring in unusual locations. In contrast to other work on explainable VAD (again with the exception of [32]), we build a model of nominal video using an exemplar selection algorithm that does not require any deep network training on new scenes. Finally, we achieve significantly improved results using spatio-temporal evaluation criteria on standard test sets.

3. Our Method

Our method consists of two stages. The first stage learns a model of normal activity from the nominal video of a scene. An illustration of this stage is shown in Figure 2. The second stage detects anomalies in test videos by looking for differences from the model of normal activity (see Figure 3). Both of these stages are based on tracklets which represent the objects present in each frame and their movements. Tracklets are computed using pre-trained deep networks for optical flow, object detection and object recognition.

3.1. Computing tracklets

Tracklets are formed by first detecting objects using both a deep network-based object detector and a motion segmentation algorithm. We use the DINO object detector [45] to detect 80 different object classes in each frame of video. Because pretrained object detectors cannot detect object classes for which they were not trained, and because video anomaly localization can involve unexpected objects, we also detect any moving objects (regardless of object class) using a motion segmentation algorithm. Our motion segmentation algorithm uses pixelwise optical flow. Details are given in Section 4.1. If an object is detected by both the object detection network and the motion segmentation algorithm (judged as having an intersection over union value of 0.5 or greater) then we discard the motion segmentation detection. We found DINO's bounding boxes to be more accurate than motion segmentation when objects are close together (crowds of people) and to be more robust to shadows. Object detection outputs a four-element vector for each object: $[x, y, w, h]$ where (x, y) are the coordinates of the center of the bounding box, w is the width, and h is the height.

For each detected object in the scene, we use a pretrained deep network object recognizer to compute an appearance embedding for the RGB image patch within the object detection bounding box. We tried two different object recognition networks: a custom-trained AppearanceNet (details in Section 4.2) based on ResNext-50 [41] and OpenCLIP [5]. We present results using each recognizer in the experimental section. The appearance embedding is a vector, f, of length N where $N = 128$ for AppearanceNet or $N = 768$ for OpenCLIP.

To represent the motion of each detected object, we use a simple short-term tracking algorithm based on optical flow. Because we are only tracking for a few frames (10 frames is used in most experiments), a simple tracking method suffices to produce good results. Details of our tracking algorithm are given in Section 4, but the main idea is to use optical flow to follow the displacement of a small sample of moving pixels within the object's bounding box from frame t to frame $t + 1$. The median displacement of the sample of pixels is used as the consensus displacement for that frame. This simple tracking algorithm works well in practice, but any tracking algorithm could be used instead. Our goal here is not to advance the state-of-the-art in tracking but rather to show that a simple tracker is enough to provide a very useful motion representation for video anomaly localization.

Tracking yields a trajectory, denoted C, for each detected object which is a sequence of T coordinates, $C = \{(x_1, y_1), (x_2, y_2), ..., (x_T, y_T)\}$, specifying the (x, y) image coordinates of the center of the bounding box in each of the T frames of the track. The (x_1, y_1) of trajectory C and the (x, y) of the center of the bounding box from object detection are the same.

Each detected object is thus represented as a *tracklet*, $\Theta = [C, w, h, f]$, containing the length T trajectory of 2D coordinates, the scalar width and height, and the length N appearance embedding.

3.2. Model of nominal video

Given nominal video of a scene, we compute tracklets for every 5th frame. These tracklets give a high-level representation of all the objects that appear in the scene in terms of their appearance (represented by the appearance embedding vectors f), their size (represented by each object's bounding box width (w) and height (h)), their movement (represented by their trajectories, C), and their locations (represented by the first coordinate in C).

We could store every tracklet found in the nominal video and use that set as our model. However, this would be wasteful since many of the tracklets are very similar. Instead, we use the exemplar selection algorithm of [25, 32]

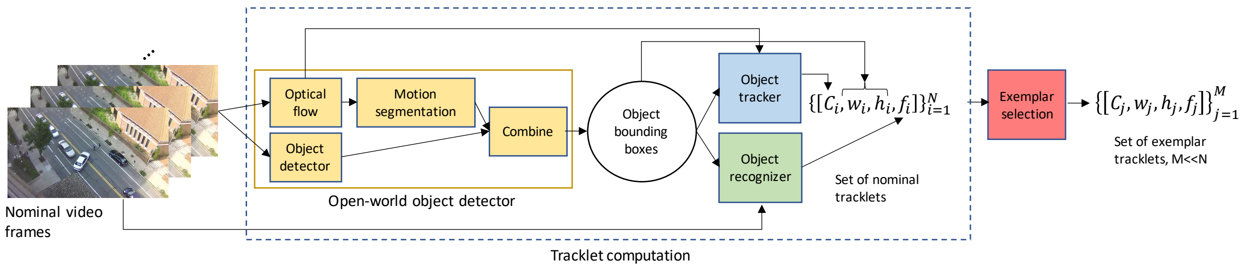

Figure 2. Computational pipeline for building a model of nominal video. See details in Sections 3.1, 3.2
.

to select a small subset of tracklets (called exemplars) that cover the variety of different tracklets found in the nominal video. Given the set of all nominal tracklets, the exemplar selection algorithm proceeds as follows:

1. Initialize the exemplar set to NULL
2. Add the first tracklet to the exemplar set.
3. For each subsequent tracklet, find its distance to the nearest tracklet in the exemplar set. If this distance is above a threshold, th, then add it to the exemplar set.

Exemplar selection yields an exemplar set $E = \{\Theta_1, \Theta_2, ..., \Theta_M\}$ of tracklets from the nominal video representing the variety of objects and their movements found in different locations of the scene. To use this simple exemplar selection algorithm we need to define the distance between two tracklets. Given two tracklets, $\Theta_1 = [C_1, w_1, h_1, f_1]$ and $\Theta_2 = [C_2, w_2, h_2, f_2]$, the distance, $D(\Theta_1, \Theta_2)$ is defined as:

$$D(\Theta_1, \Theta_2) = max(\frac{L(\Theta_1, \Theta_2) - \mu_L}{\sigma_L}, \frac{P(\Theta_1, \Theta_2) - \mu_P}{\sigma_P}, \frac{A(\Theta_1, \Theta_2) - \mu_A}{\sigma_A}, \frac{S(\Theta_1, \Theta_2) - \mu_S}{\sigma_S}), \quad (1)$$

where $L()$ is the location distance, $P()$ is the trajectory (path) distance, $A()$ is the appearance distance, $S()$ is the size distance, and the μ and σ scalars are normalization constants which make each distance function comparable.

The location distance is the Euclidean distance between the centers of the object bounding boxes in the first frame of the tracklets:

$$L(\Theta_1, \Theta_2) = \sqrt{(x_{1,1} - x_{2,1})^2 + (y_{1,1} - y_{2,1})^2}, \quad (2)$$

where $C_1 = \{(x_{1,1}, y_{1,1}), ..., (x_{1,T}, y_{1,T})\}$ and $C_2 = \{(x_{2,1}, y_{2,1}), ..., (x_{2,T}, y_{2,T})\}$ are the trajectories of each tracklet. The trajectory distance is the sum of the differences between the displacements of the first tracklet and the displacements of the second tracklet normalized by the minimum displacement:

$$P(\Theta_1, \Theta_2) = \sum_{t=1}^{T-1} \frac{|dx_{1,t} - dx_{2,t}|}{max(min(dx_{1,t}, dx_{2,t}), 1)} + \frac{|dy_{1,t} - dy_{2,t}|}{max(min(dy_{1,t}, dy_{2,t}), 1)}, \quad (3)$$

where $dx_1(t) = x_{1,t} - x_{1,t+1}$, $dx_2(t) = x_{2,t} - x_{2,t+1}$, $dy_1(t) = y_{1,t} - y_{1,t+1}$, $dy_2(t) = y_{2,t} - y_{2,t+1}$. The max function in the denominator is used to avoid division by zero.

The appearance distance is the Euclidean distance between appearance embedding vectors, f_1 and f_2:

$$A(\Theta_1, \Theta_2) = \sqrt{\sum_{i=1}^{N} (f_{1,i} - f_{2,i})^2}. \quad (4)$$

The size distance is the Euclidean distance between each tracklet's width and height normalized by the minimum width and height:

$$S(\Theta_1, \Theta_2) = \sqrt{\frac{(w_1 - w_2)^2}{min(w_1, w_2)} + \frac{(h_1 - h_2)^2}{min(h_1, h_2)}}. \quad (5)$$

Note that because $D(\Theta_1, \Theta_2)$ is a maximum over location distance as well as appearance, trajectory and size distances, different tracklets will be stored as exemplars for different locations in the frame. This is the desired behavior since what is normal in one location may not be normal in another.

3.3. Detect anomalies in test video

After the first stage of building a tracklet-based model of the nominal video is done, the second stage consists of detecting anomalies in testing video of the same scene. Anomaly localization follows much of the same pipeline as model building. The first step is to detect all objects in the testing video (using a combination of deep network object detection and motion segmentation from optical flow) and then

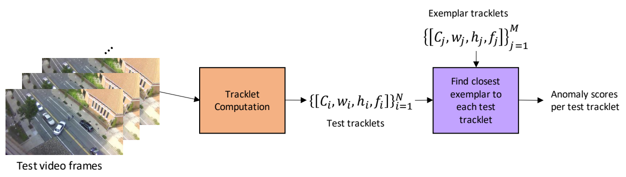
Exemplar tracklets
$\{[C_j, w_j, h_j, f_j]\}_{j=1}^{M}$
Tracklet Computation
$\{[C_i, w_i, h_i, f_i]\}_{i=1}^{N}$
Test tracklets
Find closest exemplar to each test tracklet
Anomaly scores per test tracklet
Test video frames

Figure 3. Pipeline for detecting anomalies in test video.

compute tracklets to represent the appearance, size, trajectory and location of each object. This is done using the same methods used in the model building stage.

Next we compute anomaly scores for each test tracklet by finding the nearest exemplar tracklet. (Techniques for speeding up the nearest neighbor search are discussed in the supplemental material.) The anomaly score, AS, for a tracklet, Θ, is the distance (Equation 1), to the nearest exemplar tracklet:

$$AS(\Theta, E) = \min_{\Theta_e \in E} D(\Theta, \Theta_e), \tag{6}$$

where $E = \{\Theta_1, \Theta_2, ..., \Theta_M\}$ is the exemplar set learned in the model building stage (Section 3.2).

4. Implementation details

In this section we give more details on various modules in our pipeline.

4.1. Motion segmentation

Our motion segmentation algorithm takes as input pixelwise optical flow (from RAFTv2 [35]) between successive frames. We first compute a motion mask by thresholding the L_2 flow magnitude (using a threshold of 0.5) to yield a binary mask of moving pixels. We then run a clustering algorithm (DBSCAN) on the moving pixels to find clusters of pixels with similar flows. For each cluster, we find all the connected components each of which corresponds to a connected set of pixels with similar flow, i.e. a motion segment. Each motion segment correlates strongly to a different object. We discard motion segments with fewer than 120 pixels which removes most erroneous object detections. The thresholds were chosen based on experiments with Ped1 and Ped2 which were used as our validation set (Section 5.1).

4.2. AppearanceNet

We train a slightly modified ResNext-50 [41] network to recognize five different object classes given a 128x128x3 pixel RGB image patch as input. The five classes are person, car, bicycle, dog and fire hydrant. These classes were chosen because they are commonly found in surveillance video as well as the standard datasets for VAD. Furthermore, labeled training examples are available for these classes. The ResNext-50 architecture was modified by adding an extra linear layer that maps the usual 2048 length penultimate layer to a new 128 length penultimate layer. This 128-length feature vector is the embedding we use to represent appearance. We collected training data for these classes from MS-COCO [14] and VOC2012 [7] as well as publicly available surveillance video that we annotated. In total we used 41629 person examples, 33889 car examples, 5075 bicycle examples, 2481 dog examples, and 520 fire hydrant examples. Our training set also included 59221 patches with all zero labels cropped from images that did not contain any of these five object classes. More details about this training set are in the supplemetal material. Binary cross entropy was used as our loss function which allows separate probabilities for each object class as output. The output can have high probability for more than one object class or none of them since an image patch can contain multiple objects or none of the known classes.

4.3. Tracking details

Object trajectories are formed by tracking an object detection bounding box for a small number (T) of frames (10 in most of our experiments) using pixelwise optical flow. An object trajectory consists of a set of absolute (x, y) coordinates for the center of the bounding box for each frame in the track. The initial coordinate comes from the center of the detected bounding box. From the initial bounding box, we sample k moving pixels and follow the optical flow for these points for $T - 1$ more frames. For each frame, the displacement of each point is found from the optical flow field of that frame. The median displacement of the k points is taken as the displacement of the tracked bounding box for that frame. This procedure is continued for the remaining frames in the track. Because optical flow will usually result in non-integer coordinates for the k points, bilinear interpolation of the flow field is used to compute each displacement. We use the median displacement for a small sample of moving pixels to give robustness to noisy flow fields. In our experiments we choose $k = 9$.

4.4. Computing distance normalization constants

The four components of the distance function in Equation 1 need to be comparable, i.e. need to have similar scales so that one does not dominate the others. To insure this, each distance is normalized by subtracting the mean and dividing by the standard deviation. We use pairs of tracklets from the nominal video of a dataset to compute each distance's distribution for that dataset. More details are given in the supplemental material. Normalization results in distances less than 0 if two tracklets are very similar (raw attribute distance less than the mean), and distances greater than 1 if two tracklets are significantly different (raw attribute distance greater than the mean plus standard deviation).

5. Experiments

5.1. Datasets and evaluation criteria

To validate our proposed method, we evaluate our method on the following datasets: UCSD Ped1 and Ped2 [20], CUHK Avenue [17], Street Scene [25] and ShanghaiTech [19]. We use UCSD Ped1 and Ped2 for hyperparameter tuning and test our method on CUHK Avenue, Street Scene and ShanghaiTech datasets.

UCSD Ped1 & Ped2: The UCSD Ped1 and Ped2 datasets consists of 14,000 (Train=6800 / Test=7200) and 4560 (Train=2550/ Test=2010), respectively, video frames of pedestrian walkways. The nominal videos contain only pedestrians while the anomalies consist of (1) non-pedestrian entities (bikers, skaters and cars) and (2) unusual pedestrian motion patterns.

CUHK Avenue: The Avenue dataset contains 30,652 video frames (Train=15,328 / Test=15,324) of a college campus. The abnormal events include unseen actions (running), similar actions with unseen motion patterns (walking in wrong direction) or unseen actions (throwing a backpack).

Street Scene: The Street Scene dataset is a large dataset consisting of 203,257 video frames (Train=56,847 / Test=146,410) representing a scene of a two-lane street with bike lanes and pedestrian sidewalks. Prominent examples of anomalies include loitering, vehicles driving outside their lanes and jaywalking.

ShanghaiTech: The ShanghaiTech dataset is a multi-scene benchmark for video anomaly localization. It consists of 296,690 video frames (Train=255,899 / Test=40,791) recording different scenes in a college campus. Major categories of anomalies include people riding bikes or skating in pedestrian zones, fighting, jumping and chasing. We evaluate our method on ShanghaiTech to highlight robustness and applicability of our method to multi-scene benchmarks. Specifically, the main focus of our framework is on the single scene video anomaly localization task, which includes location-dependent anomalies [27]. Even though the location-dependent aspect of our model is not necessary for a multi-scene dataset, we evaluate our method on ShanghaiTech without any modification to highlight its robustness and wide applicability. We expect further improvements in accuracy if we specialize our model to multi-scene datasets.

Evaluation criteria. We use the Region-Based Detection Criterion (RBDC) and the Track-Based Detection Criterion (TBDC) as proposed in [25] as our evaluation criteria and report the area under the curve (AUC) for false positive rates per frame from 0 to 1. These criteria measure both temporal and spatial localization accuracy. We do not use the pixel-level criterion [20] due to its well known flaws [25] or the popular frame-level criterion [20] because it does not measure spatial localization. In addition to not caring where an anomaly detection is located, the frame-level criterion also does not take into account the number of false positive regions when computing false positive rates.

Hyper-parameter tuning. For exemplar selection, we select the threshold value th by using the Ped1 and Ped2 datasets as our validation set. We chose Ped1 and Ped2 as our validation set because performance on these datasets is saturated and hence they are not as useful for comparative study. Furthermore, as noted by previous works [26, 32] ground-truth annotations of Ped1 and Ped2 do not label every location-specific anomaly. Thus, we follow [32] in using the new ground-truth labels that include all location-specific anomalies. We show results on Ped1 and Ped2 with different threshold values for exemplar selection in Table 1. From the table, we can see that the choice of threshold has a relatively small effect on accuracy. It mainly effects the number of exemplars chosen to model nominal tracklets. We select $th = 0.25$ as our final threshold value for evaluation on other benchmarks as it provides a good trade-off between computational efficiency and test accuracy.

5.2. Results

We provide comparative results of our method with other recent approaches on Avenue, ShanghaiTech and Street Scene in Tables 2 and 3. From the results, we can see that our framework outperforms other methods in accurately localizing anomalies. Specifically, we achieve state-of-the-art results on Avenue with respect to RBDC and TBDC criteria, while for ShanghaiTech our method achieves best RBDC score and second-best TBDC scores (Table 2). On the Street Scene dataset, which is more challenging than other benchmarks due to large and diverse set of anomalous events, we outperform the previous best performing method by more than 6% and 8% with respect to RBDC and TBDC metrics. The strong results demonstrated across five benchmarks with diverse sets of anomalies highlight the generality and robustness of our framework.

5.3. Ablation study

We perform the following ablation experiments to gain further insight into the impact of some of the design choices in building our proposed framework.

Importance of individual attributes. To evaluate the importance of individual components of our tracklet representation, we perform a comparative study by building location-dependent models with only a single other component (appearance, trajectory or size). To do this, we simply use only the location distance and one other component's distance in the maximum function of Equation 1. This is done both to select exemplars and to compute anomaly scores for test tracklets. We present our results in Table 4. We see that all components encode useful information in detecting anomalies in different situations. It is interesting to see that a location-based appearance model does very well

th	UCSD Ped1			UCSD Ped2		
	RBDC	TBDC	NUM	RBDC	TBDC	NUM
2	40.8	87.1	252 / 14269	80.3	96.6	108 / 7050
1.5	43.2	88.8	380 / 14269	82.5	96.9	150 / 7050
1	44.0	89.0	644 / 14269	84.1	98.3	267 / 7050
0.75	44.4	89.8	961 / 14269	82.3	96.4	344 / 7050
0.5	44.9	89.7	1382 / 14269	83.3	96.6	461 / 7050
0.25	**45.3**	**89.4**	**2133 / 14269**	**84.2**	**97.9**	**673 / 7050**
0.0	45.5	89.6	3525 / 14269	83.4	98.2	1047 / 7050

Table 1. RBDC, TBDC and frame-level scores (in %) of our method for different exemplar thresholds (th) on UCSD Ped1 and Ped2. NUM denotes the number of exemplar tracklets selected vs the total number of tracklets present in the training set. Bold row indicates th chosen for later experiments.

on Street Scene. This is most likely because many anomalies in Street Scene involve normal object classes in unusual locations (jaywalking, biker outside of a bike lane or a car u-turning). A location-based trajectory model does well on Ped2 and Avenue. This is probably due to the fact that many anomalies in those datasets involve unusual speeds such as people sprinting or fast-moving cyclists, skateboarders or golf carts. The best results are obtained when all components are used jointly, thus highlighting the importance of a variety of appearance and motion components to localize anomalies under many different scenarios.

Tracklet length. We evaluate the effect of change in length of tracklets, T, by computing anomaly localization results using different track lengths on Ped1 and Ped2 (Table 5). We find that tracklets with length $T = 10$ give us best results. Using only 5 frames hurts accuracy due to not enough information about the object's trajectory. Using 15 frames also hurts accuracy a little probably due to the added difficulty of tracking over more frames.

Appearance features. We further perform an ablation study on UCSD Ped1 and Ped2, Avenue and Street Scene datasets to compare our AppearanceNet model for representing object appearance versus using OpenCLIP [5]. For the OpenCLIP pre-trained model, we specifically used the ViT-L/14 model pretrained on DataComp-1B dataset and used the image encoder output as the appearance feature embedding. We present our results in Table 6 which shows that AppearanceNet is more accurate in all cases although the difference is relatively small in most cases. The superiority of AppearanceNet is most likely due to its focus on the small number of object classes that are most common in outdoor surveillance video in contrast to OpenCLIP which attempts to learn a vast number of object categories.

Method	Avenue		ShanghaiTech	
	RBDC	TBDC	RBDC	TBDC
Ionescu *et al.*[12]	15.8	27.1	20.7	44.5
Ramachandra *et al.* [25]	35.8	80.9	-	-
Ramachandra *et al.* [26]	41.2	78.6	-	-
Georgescu *et al.* [8]	57.0	58.3	42.8	83.9
Liu *et al.* [15]	19.6	56.0	17.0	54.2
Liu *et al.* [16]	41.1	86.2	44.4	83.9
Georgescu *et al.* [9]	65.1	66.9	41.3	78.8
Liu *et al.*[15] + Ristea *et al.* [29]	20.1	62.3	18.5	60.2
Liu *et al.*[16] + Ristea *et al.* [29]	62.3	**89.3**	45.5	84.5
Georgescu *et al.*[9] + Ristea *et al.* [29]	66.0	64.9	40.6	83.5
EVAL [32]	**68.2**	87.56	**59.2**	**89.4**
SSMTL++v1 [1]	40.9	82.1	43.2	84.1
SSMTL++v2 [1]	47.8	85.2	47.1	85.6
Our Method	**69.6**	**89.6**	**59.6**	**87.6**

Table 2. RBDC, TBDC and Frame AUC scores (in %) of various state-of-the-art methods on Avenue and ShanghaiTech datasets. The top score for each metric is highlighted in **red**, while the second best score is in **bold.**

Methods	RBDC	TBDC
Auto-encoder [10]	0.3	2.0
Dictionary method [17]	1.6	10.0
Flow baseline [25]	11.0	52.0
FG Baseline [25]	21.0	53.0
EVAL [32]	**24.3**	**64.5**
Our method	**30.9**	**72.9**

Table 3. RBDC and TBDC AUC scores (in %) of various baseline methods on Street Scene dataset. The top score for each metric is highlighted in **red**, while the second best score is highlighted in **bold.**

Attributes	Ped1	Ped2	Avenue	SS
App	35.8 / 71.4	74.4 / 92.1	48.9 / 70.1	29.4 / 69.1
Traj	34.9 / 85.9	76.4 / 97.6	57.3 / 89.5	20.0 / 59.6
Size	28.3 / 68.7	65.8 / 93.1	55.7 / 76.4	21.1 / 58.4
Tracklet	**45.3 / 89.4**	**84.2 / 97.9**	**69.6 / 89.6**	**30.9 / 72.9**

Table 4. RBDC / TBDC AUC scores (in %) of our method when using individual components of tracklet representation (appearance, trajectory and size) along with the location component compared with the full representation (Tracklet).

Track Length	Ped1	Ped2
5	40.0 / 84.2	81.5 / 94.6
10	**45.3 / 89.4**	**84.2 / 97.9**
15	44.4 / 86.3	82.0 / 94.4

Table 5. RBDC / TBDC AUC scores (in %) of our method when considering different maximum lengths for each tracklet.

6. Explainability

The explainability of our method comes from the fact that our model uses human-interpretable features to describe what is happening in a scene. Figure 4 shows a visualization of a test tracklet from Street Scene. The top, left image in the figure shows part of the first frame for the tracklet as well as the bounding box for the tracklet in that frame. The top, right image visualizes all elements of the tracklet. The image crop in the tracklet visualization shows the object's appearance and represents the appearance embedding vector. The trajectory is shown in the right side of the tracklet visualization. The initial coordinate of the trajectory is colored green and the final coordinate is colored red. It is

Attributes	Ped1	Ped2	Avenue	SS
App	**45.3 / 89.4**	**84.2 / 97.9**	**69.6 / 89.6**	**30.9 / 72.9**
OpenCLIP	43.4 / 87.8	82.0 / 97.6	62.8 / 87.2	28.9 / 66.4

Table 6. RBDC / TBDC AUC scores (in %) of our method when using our pre-trained model (App) versus OpenCLIP pe-trained model as appearance feature extractor.

scaled to fill the space to make it more visible. Without this scaling, tracklets with small total displacement would not be visible. This scaling for visualization essentially distorts the velocity inherent in the trajectory, so the average velocity is given below the trajectory to indicate the amount of distortion. The scaling factor for visualizing the trajectory is inversely proportional to the trajectory's average velocity. The height and width of the tracklet's bounding box is shown to the left and below the appearance patch. The top of the tracklet visualization shows the absolute (x, y) coordinates for the location of the tracklet. The visualization of the test tracklet in Figure 4 shows a cyclist moving downward near the middle of the frame. The exemplar tracklet with smallest distance to this test tracklet is shown at the bottom, right of Figure 4. It is a car moving down and to the right since this is the only object class occurring near this location in the scene. Exemplars with normal cyclists occur much further away and thus have large location distances. From these visualizations, we can immediately see that the test tracklet is a different object class (cyclist) from the closest normal tracklet and that it is moving with a very different trajectory. A simple textual explanation of the anomaly which is automatically generated is shown at the bottom, left of the figure along with the four distance scores for the location (L), trajectory (P), appearance (A) and size (S).

Figures 5 and 6 show more visualizations of an anomalous tracklet and the nearest exemplar tracklet. For the car making a u-turn in Figure 5 it is clear that the trajectory is what is anomalous and for the sprinting person in Figure 6, it is also the trajectory and especially the velocity of the test tracklet that is anomalous.

7. Conclusions

In this work, we have proposed a novel interpretable object-centric framework for detecting anomalies in a given video sequence. Our framework is based on (1) decomposing a scene into objects, (2) representing each object into human-understandable attributes (trajectory, appearance, size, and location), and (3) constructing an efficient and informative model of nominal video by identifying exemplar object instances. Importantly, learning a model of a new scene does not require any deep network training which makes it much more practical for real-world applications. It is also trivial to extend a model when new nominal video is available, making it ideal for continual learning. Our method

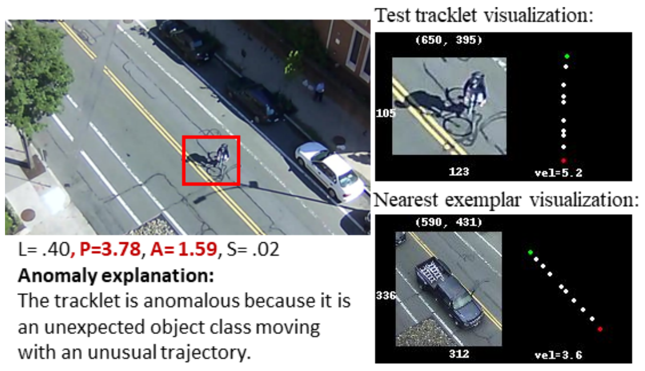
Test tracklet visualization:
(650, 395)
105
123
vel=5.2
Nearest exemplar visualization:
(590, 431)
336
312
vel=3.6
L= .40, P=3.78, A= 1.59, S= .02
Anomaly explanation:
The tracklet is anomalous because it is an unexpected object class moving with an unusual trajectory.

Figure 4. Visualization example for a region of Street Scene showing an explanation of the "biker outside lane" anomaly.

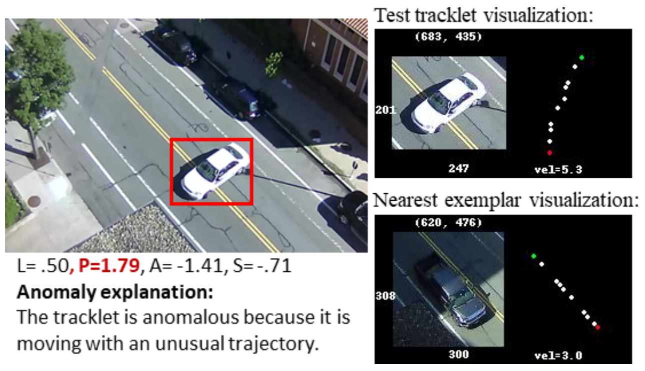
Test tracklet visualization:
(683, 435)
201
247
vel=5.3
Nearest exemplar visualization:
(620, 476)
308
300
vel=3.0
L= .50, P=1.79, A= -1.41, S= -.71
Anomaly explanation:
The tracklet is anomalous because it is moving with an unusual trajectory.

Figure 5. Visualization example for a region of Street Scene showing an explanation of the "car making u-turn" anomaly.

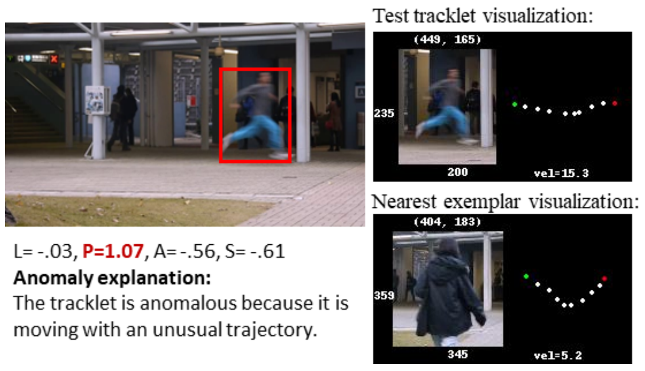
Test tracklet visualization:
(449, 165)
235
200
vel=15.3
Nearest exemplar visualization:
(404, 183)
359
345
vel=5.2
L= -.03, P=1.07, A= -.56, S= -.61
Anomaly explanation:
The tracklet is anomalous because it is moving with an unusual trajectory.

Figure 6. Visualization example for a region of CUHK Avenue showing an explanation of the "person sprinting" anomaly.

achieves strong results on five different benchmarks and, in particular, state-of-the-art results under RBDC and TBDC criteria on CUHK Avenue, Street Scene, and ShanghaiTech datasets, highlighting the precision and generalizability of our method. Furthermore, our method is interpretable; every anomaly classification decision can be explained in terms of an object's appearance and motion attributes that differ from objects observed in nominal video. Thus, our method exhibits a number of desirable properties for a robust, real-world video anomaly localization system.

References

[1] Antonio Barbalau, Radu Tudor Ionescu, Mariana-Iuliana Georgescu, Jacob Dueholm, Bharathkumar Ramachandra, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Ssmtl++: Revisiting self-supervised multi-task learning for video anomaly detection. *Computer Vision and Image Understanding*, 229:103656, 2023. 7

[2] Sovan Biswas and R. Venkatesh Babu. Anomaly detection via short local trajectories. *Neurocomputing*, 242, 2017. 2

[3] Yunpeng Chang, Zhigang Tu, Wei Xie, and Junsong Yuan. Clustering driven deep autoencoder for video anomaly detection. In *European Conference on Computer Vision*, pages 329–345. Springer, 2020. 2

[4] Nicholas F.Y. Chen, Zhiyuan Du, and Khin Hua Ng. Scene graphs for interpretable video anomaly classification. In *NIPS*, 2018. 2

[5] Mehdi Cherti, Romain Beaumont, Ross Wightman, Mitchell Wortsman, Gabriel Ilharco, Cade Gordon, Christoph Schuhmann, Ludwig Schmidt, and Jenia Jitsev. Reproducible scaling laws for contrastive language-image learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2818–2829, 2023. 3, 7

[6] Keval Doshi and Yasin Yilmaz. Towards interpretable video anomaly detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2023. 2

[7] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, 2015. 5

[8] Mariana-Iuliana Georgescu, Antonio Barbalau, Radu Tudor Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. Anomaly detection in video via self-supervised and multi-task learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12742–12752, 2021. 7

[9] Mariana Iuliana Georgescu, Radu Ionescu, Fahad Shahbaz Khan, Marius Popescu, and Mubarak Shah. A background-agnostic framework with adversarial training for abnormal event detection in video. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021. 7

[10] Mahmudul Hasan, Jonghyun Choi, Jan Neumann, Amit K Roy-Chowdhury, and Larry S Davis. Learning temporal regularity in video sequences. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 733–742, 2016. 2, 7

[11] Ryota Hinami, Tao Mei, and Shin'ichi Satoh. Joint detection and recounting of abnormal events by learning deep generic knowledge. In *Proceedings of the IEEE international conference on computer vision*, pages 3619–3627, 2017. 2

[12] Radu Tudor Ionescu, Fahad Shahbaz Khan, Mariana-Iuliana Georgescu, and Ling Shao. Object-centric auto-encoders and dummy anomalies for abnormal event detection in video. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7842–7851, 2019. 2, 7

[13] Asiegbu Miracle Kanu-Asiegbu, Ram Vasudevan, and Xiaoxiao Du. Leveraging trajectory prediction for pedestrian video anomaly detection. In *IEEE Symposium Series on Computational Intelligence*, 2021. 2

[14] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft coco: Common objects in context. In *Computer Vision–ECCV 2014: 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V 13*, pages 740–755. Springer, 2014. 5

[15] Wen Liu, Weixin Luo, Dongze Lian, and Shenghua Gao. Future frame prediction for anomaly detection–a new baseline. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6536–6545, 2018. 2, 7

[16] Zhian Liu, Yongwei Nie, Chengjiang Long, Qing Zhang, and Guiqing Li. A hybrid video anomaly detection framework via memory-augmented flow reconstruction and flow-guided frame prediction. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 13588–13597, 2021. 1, 2, 7

[17] Cewu Lu, Jianping Shi, and Jiaya Jia. Abnormal event detection at 150 fps in matlab. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2720–2727, 2013. 6, 7

[18] Yiwei Lu, Frank Yu, Mahesh Kumar Krishna Reddy, and Yang Wang. Few-shot scene-adaptive anomaly detection. In *European Conference on Computer Vision*, pages 125–141. Springer, 2020. 2

[19] Weixin Luo, Wen Liu, and Shenghua Gao. A revisit of sparse coding based anomaly detection in stacked rnn framework. In *Proceedings of the IEEE international conference on computer vision*, pages 341–349, 2017. 6

[20] Vijay Mahadevan, Weixin Li, Viral Bhalodia, and Nuno Vasconcelos. Anomaly detection in crowded scenes. In *2010 IEEE computer society conference on computer vision and pattern recognition*, pages 1975–1981. IEEE, 2010. 6

[21] Romero Morais, Vuong Le, Truyen Tran, Budhaditya Saha, Moussa Mansour, and Svetha Venkatesh. Learning regularity in skeleton trajectories for anomaly detection in videos. In *CVPR*, 2019. 2

[22] Rashmiranjan Nayak, Umesh Chandra Pati, and Santos Kumar Das. A comprehensive review on deep learning-based methods for video anomaly detection. *Image and Vision Computing*, 106, 2021. 2

[23] Trong-Nguyen Nguyen and Jean Meunier. Anomaly detection in video sequence with appearance-motion correspondence. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1273–1283, 2019. 2

[24] Claudio Piciarelli, Christian Micheloni, and Gian Luca Foresti. Trajectory-based anomalous event detection. *IEEE Transactions on Circuits and Systems for Video Technology*, 18(11):1544–1554, 2008. 2

[25] Bharathkumar Ramachandra and Michael Jones. Street scene: A new dataset and evaluation protocol for video anomaly detection. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2569–2578, 2020. 1, 3, 6, 7

[26] Bharathkumar Ramachandra, Michael Jones, and Ranga Vatsavai. Learning a distance function with a siamese network to localize anomalies in videos. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2598–2607, 2020. 2, 6, 7

[27] Bharathkumar Ramachandra, Michael Jones, and Ranga Raju Vatsavai. A survey of single-scene video anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020. 6

[28] Bharthkumar Ramachandra, Michael J. Jones, and Ranga Raju Vatsavai. A survey of single-scene video anomaly detection. *IEEE Transactions on Pattern Analysis and Machine Learning*, 2020. 2

[29] Nicolae-Catalin Ristea, Neelu Madan, Radu Tudor Ionescu, Kamal Nasrollahi, Fahad Shahbaz Khan, Thomas B Moeslund, and Mubarak Shah. Self-supervised predictive convolutional attentive block for anomaly detection. *arXiv preprint arXiv:2111.09099*, 2021. 7

[30] Royston Rodrigues, Neha Bhargava, Rajbabu Velmurugan, and Subhasis Chaudhuri. Multi-timescale trajectory prediction for abnormal human activity detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020. 2

[31] Chenrui Shi, Che Sun, Yuwei Wu, and Yunde Jia. Video anomaly detection via sequentially learning multiple pretext tasks. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 1

[32] Ashish Singh, Michael J. Jones, and Erik Learned-Miller. Eval: Explainable video anomaly localization. In *CVPR*, 2023. 2, 3, 6, 7

[33] Chris Stauffer and W. Eric L. Grimson. Learning patterns of activity using real-time tracking. *IEEE TPAMI*, 22(8):747–757, 2000. 2

[34] Che Sun, Yunde Jia, Yao Hu, and Yuwei Wu. Scene-aware context reasoning for unsupervised abnormal event detection in videos. In *ACMMM*, 2020. 2

[35] Deqing Sun, Charles Herrmann, Fitsum Reda, Michael Rubinstein, David J Fleet, and William T Freeman. Disentangling architecture and training for optical flow. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXII*, pages 165–182. Springer, 2022. 5

[36] Stanislaw Szymanowicz, James Charles, and Roberto Cipolla. X-man: Explaining multiple sources of anomalies in video. In *CVPR*, 2021. 2

[37] Stanislaw Szymanowicz, James Charles, and Roberto Cipolla. Discrete neural representations for explainable anomaly detection. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2022. 2

[38] Chenxu Wang, Yanxin Yao, and Han Yao. Video anomaly detection method based on future frame prediction and attention mechanism. In *IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2021. 2

[39] Xiaogang Wang, Kinh Tieu, and Eric Grimson. Learning semantic scene models by trajectory analysis. Technical Report MIT-CSAIL-TR-2006-008, MIT Computer Science and Artificial Intelligence Laboratory (MIT CSAIL). 2

[40] Chongke Wu, Sicong Shao, Cihan Tunc, Pratik Satam, and Salim Hariri. An explainable and efficient deep learning framework for video anomaly detection. *Cluster Computing*, pages 1–23, 2021. 2

[41] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1492–1500, 2017. 3, 5

[42] Rajesh Kumar Yadav and Rajiv Kumar. A survey on video anomaly detection. In *IEEE Delhi Section Conference (DELCON)*, 2022. 2

[43] Cheng Yan, Shiyu Zhang, Yang Liu, Guansong Pang, and Wenjun Wang. Feature prediction diffusion model for video anomaly detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2023. 1

[44] Zhiwei Yang, Jing Liu, Zhaoyang Wu, Peng Wu, and Xiaotao Liu. Video event restoration based on key frames for video anomaly detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2023. 1, 2

[45] Hao Zhang, Feng Li, Shilong Liu, Lei Zhang, Hang Su, Jun Zhu, Lionel M Ni, and Heung-Yeung Shum. Dino: Detr with improved denoising anchor boxes for end-to-end object detection. *arXiv preprint arXiv:2203.03605*, 2022. 3

[46] Tianzhu Zhang, Hanqing Lu, and Stan Z. Li. Learning semantic scene models by object classification and trajectory clustering. In *CVPR*, 2009. 2