VNU Journal of Science: Com. Science & Com. Eng., Vol. 39, No. 2 (2023) 90-100

VNU Journal of Science:
Computer Science and Communication Engineering

Journal homepage: http://www.jcsce.vnu.edu.vn/index.php/jcsce

Original Article

Noisy-label Propagation for Video Anomaly Detection with Graph Transformer Network

Do Thu Uyen, Ta Viet Cuong*

VNU University of Engineering and Technology,144 Xuan Thuy, Cau Giay, Hanoi, Vietnam

Received 05 January 2023
Revised 03 April 2023; Accepted 15 May 2023

Abstract: In this paper, we study the efficiency of Graph Transformer Network for noisy label propagation in the task of classifying video anomaly actions. Given a weak supervised dataset, our methods focus on improving the quality of generated labels and use the labels for training a video classifier with deep network. From a full-length video, the anomaly properties of each segmented video can be decided through their relationship with other video. Therefore, we employ a label propagation mechanism with Graph Transformer Network. Our network combines both the feature-based relationship and temporal-based relationship to project the output features of the anomaly video to a hidden dimension. By learning in the new dimension, the video classifier can improve the quality of noisy, generated labels. Our experiments on three benchmark dataset show that the accuracy of our methods are better and more stable than other tested baselines.

Keywords: Anomaly Detection, Graph Transformer Network, Weak Supervised, Noisy Labeling.

1. Introduction

Detecting anomaly action from video data plays an important role in video-based surveillance systems [1, 2]. The definition of anomaly actions in a surveillance system is highly associated with the semantic meaning of the scene, which is to identify the unusual action through unusual appearance or motion attributes in unsealing locations or times [3]. One of the main challenges for detecting an anomaly event in video data is the low frequency of anomaly events against normal events. The detectors are required to learn the distribution of the temporal and spatial patterns to be able to separate unusual actions from normal ones. Popular approaches include supervised, weak-supervised, and unsupervised methods. In the supervised approach, the context in which happens an abnormal action is well-defined.

* Corresponding author.
E-mail address: cuongtv@vnu.edu.vn

https://doi.org/10.25073/2588-1086/vnucsce.659

Therefore, instead of learning the action distributions, it is possible to treat the abnormal action as a class label and employ standard video classification models to classify the unusual actions [4]. The unsupervised approaches work with a broader context, which classifies the anomaly actions as rare occurrence events. Because of the resemblance to real-world settings, this setting isa more active research area with popular works such as [5, 6]. One of the main advantagesof unsupervised approaches is that the numberof unlabeled, normal actions in the video is larger than the abnormal ones. Therefore, those approaches are less restricted to the scared of training data. However, it is required that the detector can extract useful features for modeling the distribution, which is a challenging task given that the surveillance video is a rich source of information. The weak-supervised method balances the supervised and unsupervised approaches. Employs several abnormal data in training. By combining a small number of abnormal data with normal data in training, the detector can be directed to learning the unusual behavior in the videos [7].

In a video-based classification approach, one way to build the weak label is to mark a long video with a normal or abnormal label without explicitly stating the start and end of the abnormal actions. The classifier would try to identify the small segment from the long videos which correspond to the abnormal actions (Fig. 1). This setting is the resemblance to several benchmark datasets such as CUHK Avenue [5] or USCD Ped1/USCD Ped2 [6]. To effectively train the video classifier, noisy labels for each small segment can be generated from the weak label and then used to train the classifier [8]. In the domain of video anomaly detection, Zhong et al. [9] employthe graph representation of video segments within a long complete video for filtering the noisy labels. They use the Graph Convolution Network (GCN) [10] for propagating the label from the high-confidence nodes to the low- confidence ones. Two types of neighborhoods are considered for label propagation including features-based adjacency and time-based adjacency. After filtering the noisy labels, popular architectures for video action classifiers are used to learn, such as C3D [11] or TSN[12]. Other works use graph structures to mine the similarity between segmented videos suchas [13]. In our approaches, we improve the drawback of convolution operations in GCN by incorporating the self-attention mechanism of Graph Transformer Network (GTN) [14]. The usage of attention layers which are built upon the convolution layers is two-fold. By employing the global self-attention operators, far nodes can be connected through their similarity structures. In addition to that, the global properties of GTN allow multiple types of connection can be modeled and projected into the same latent space. Our experiments on three benchmark datasets including USCD Ped1, USCD Ped 2 and Avenue dataset in a weak-supervised setting show that the GTN architecture could improve the anomaly detection AUC score substantially and be more stable than the baselines. In the remainder of the paper, we give related works in Section II; our methods are described in Section III; Section IV illustrates the experiments and results on several benchmark datasets; the conclusion and future works are present in Section V.

2. Related Works

Detecting anomalies in the video is a challenging task [1, 2]. Several pioneers work as described in [15]. Recent works rely on the power of deep neural networks in video analysis to detect anomaly actions [16]. On the weak-supervised based approach, multi-instance learning could be used used as a template for motion modeling in the weak monitoring problem [9, 17]. For detecting anomaly videos, there is some popular dataset benchmark such as CUHK Avenue [5] and USCD Ped1/USCD Ped2 [6]. The work in [5] employs k-nearest neighbors clustering with a motion-field shape description on USCD Ped1 to get an AUC of 86.9% on frame-level detection. However, their approaches highly

depend on several sensitive hyper-parameters such as the size of the motion field. On the Avenue dataset, the DMAD framework [18] can reach 92.8% AUC. Nevertheless, the framework requires learning an encoder-decoder network to memorize the semantics of the scene which takes a long time to train. For video analysis by using deep neural networks, three dimensions convolutional - C3D architectures [19] are widely used in video recognition applications, especially in action recognition problems. It is used as a technique for feature extraction. 3D convolutional layers extract both spatial and temporal components related to the motion of objects, human actions, person-to-object interactions, and the appearance of objects, people, and that scenes. In addition to that, Temporal Segment Network takes a different approach by extracting features from random videos through convolutional operators in 2D and 3D. There are several variants built on TSN such as [19, 20]. The TAM architecture [19] (Temporal Aggregation Module) employs the aggregation module to synthesize features from TSN module. The approach in TAM is designed to capture the temporal information in the video through a hierarchical setting. At a lower layer, for recognizing features at different imagescales, TAM uses the Big-Little Network [19] for learning video features. The relationship between different frames in the video is then aggregated by a depth-wise convolution. The design ofTAM in extracting temporal features is as efficient as the 3D convolution in 3D but with fewerparameters. In general, the TAM architecture can be considered a lightweight module to extract the representation of the video. Recently, due to the effective performance of Graph Neural Networks (GNN), several works on video analysis domains employ graph structures to model the spatio-temporal relationship of video. For example, the graph is used to improve the object tracking results in [21]. In [13], graph structures are used to group similar video segments. The authors in [22] learn a GNN for video retrieval with text. Standard GNN approaches include composing feed-forward layers [23] and message-passing methods like in [24]. More advanced graph learning architecture attempt to adapt successful deep network modules from other domains such as image and text into the graph domains. Graph Convolution Network (GCN) [10] is a promising approach that extends the standard convolution layer from the image into the graph learning domain. The GCN design is based on spectral graph theory [25, 26], which decomposes the graph signal over the spectral domain and defines a series of parameterized filters for convolution. The Graph Transformer Network (GTN) [14] and other related variants [27, 28] extend the well-known attention architecture from the text domain into the graph. The main usage of GTN for video analysis is its ability to workwith multiple relations between video elements. While standard approaches on GCN can only join the heterogeneous graphs at the later embedding layers, the works of Yun et. al. can transform the multiple-relationship graphs into a meta-path graph and employ the attention mechanism to improve the learning embedding process. Nevertheless, using GNN-based approaches in anomaly detection tasks is challenging due to the smoothness effect when a noisy signal is propagated through the graph [29, 30].

3. Video Anomaly Detection with Graph Transformer Network

3.1. Overview

To be able to detect the video part containing anomalies from a long video with the weak label, we rely on a video classifier to assign the noisy label to each segment video. In our works, we employ the setup that each segment video is extracted from the long video by using a fixed number of frames. Fig. 1 illustrates our approach.

The long, full-length video is given using weak labels, *Normal* or *Anomaly*. Given a full-length video, the part colored red is marked with an anomaly labeled. However, while labeling such data is take little time, it is not practical to work all the full length of the video.

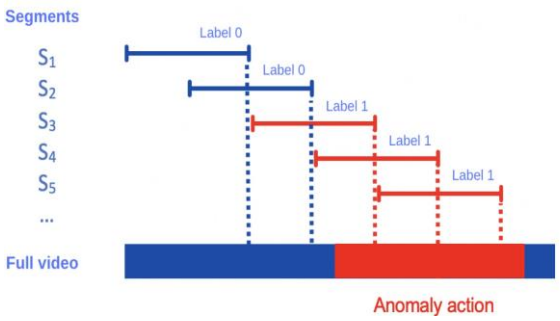

Figure 1. Using a weak label from the full video to extract segmented samples for training an abnormal action detection.

To be able to train a video-action classifier to detect the anomalies, we split the data into smaller segments with a fixed window length W and a steps length s: S_1, S_2, ..., S_N with Nbeing the number of segmented videos. The S_i islabeled anomaly (Label 1) if it contains anomalyaction, otherwise, the label of S_i is normal (Label0). The weak-supervised classifier method can only access the weak label, which is whether thefull-length video contains anomaly actions or not.

In this section, we will present the overview of our self-supervised method for working with weak-label video. It includes two separate components: a video classifier to train the anomaly detector and generate the noisy labels based on the weak label; a filtering step to improve the noisy labels. The two main components are shown in Fig. 2, which include:

Action classifier component: C_θ where θ serves as the parameters of the trainable deep learning model based on the input data. C_θ acts as the main learning unit used to generate noise labels from the split videos S_1, S_2, ..., S_N and to learn about noise labels. For each S_i, the output of C_θ would be the noise label γ_i, the probability P_i, and the feature X_i, where i is the sample index. For the selection of C_θ, deep network-based video classifiers such as C3D [11], TSN [12], and their variants such as TAM [19] can be considered as a suitable candidate.

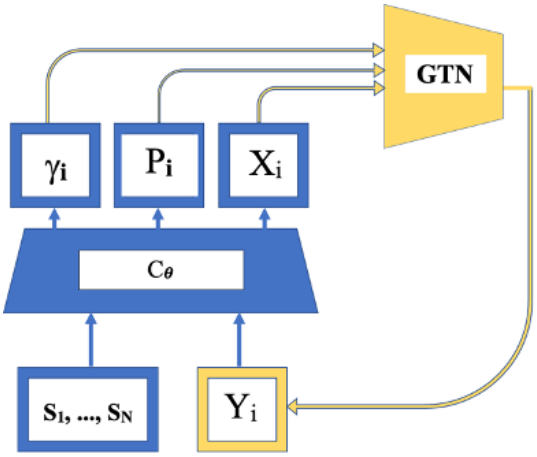

Figure 2. Overview of our methods for filtering the noisy labels and training an action classifier to detect anomaly segments in the video.

The noise filter component: the action classifier C_θ can be used to generate the noisy labels Y_i and trained on the generated label as in a traditional self-supervised approach. However, it usually makes the noisy signal generated from the action classifier C_θ suppress the useful features for differing normal and anomaly actions. Therefore, we improve a filtering phase based on graph transformer architecture to filter out irrelevant signals.

In our work, we employ label propagation based on GTN architecture. The learning GTN has the main effect of spreading noisy labels based on the relationship of time, which is the order of frames in the video, and the relationship of features X_i, which is extracted from the action classifier C_θ.

3.2. Noisy labeling for training video anomaly detection from a weak label

Firstly, from the weak label L of the input video S, we apply a self-supervised approach to generate the noisy label for each segmented S_1, S_2, ..., S_N with N being the number of segmented videos which are extracted from S . For each filtering step from 1 to K, the action classifier would be used to generate the label, Y_i. Firstly,

we assume that C_θ is pre-trained on the action classifier video dataset and capable of separating the usual and unusual segmented video S_i. Let P_i is the probabilities of a given S_i containing anomaly actions with label 1:

$$P_i = P(Y_i|S_i, \theta) \quad (1)$$

A threshold T is applied to split the output P_i into the label Y_i, which is 0 - for normal label and 1 - for anomaly videos. And then, the generatedY_i is used to train the classifier C_θ. The detailsof this approach are illustrated in Algorithm 1 below:

Algorithm 1 Extracting and training with noisylabels in a self-supervised approach

Input:
$S_1, S_2, ..., S_N$: Segmented videos from S with weak label L
C_θ: Action classifier deep learning model with parameters θ
T: Weak signal filter threshold
K: Number of filtering steps
Output:
- Return a trained C_θ model with noise labels
Algorithm:
1. If label L=0, assign label $Y_i = 0$
2. If label L=1, for each step k from 1..K
a. Generate $P_i = P(Y_i = 1|S_i, \theta)$
b. Update the noise label $Y_i = 1$ if $P_i \geq T$, otherwise $Y_i = 0$
c. Train θ with the new list of (S_i, Y_i) data

The noisy labels in Step 2 are only valid for the anomaly videos. With the videos containing only normal action, all the generated Y_i is assigned 0. In step 2c, the algorithm will synthesize all received interference labels and proceed to learn and retrain the C_θ model. In this simple approach, the interaction between normal and abnormal segmented S_i is only through the C_θ, which is dependent heavily on the first filter step at $k = 1$. In addition to that, the number of label anomalies is much lower than the normal label. Hence, the filtering process usually results in the skew to predict normal labels when the number of filtering processes K is increased.

3.3. Noisy label propagation with Graph Transformer Network

To improve the interaction between the normal and abnormal labels in the self-supervised learning steps, we construct a graph based on the video-based representation and temporal relationship between each sample S_i. Several works in [9] and [13] propose a similar graph-based structure to model the relationship of samples or events in a long video. Given a set of segmented video $S_1, S_2, ..., S_N$ extracted from a weak label sample S with anomaly action, we build two undirected graphs with the nodes are the set $S_1, S_2, ..., S_N$ as follows:

The feature-based graph: We construct the graph as an undirected, no-weight graph A^F by comparing the feature representation X_i of each S_i:

$$A_{ij}^F = \begin{cases} 1, & \text{if } X_i * X_j \geq \delta \\ 0, & \text{if } X_i * X_j < \delta \end{cases} \quad (2)$$

In our works, we derive X_i directly fromthe representation layer of C_θ, which is a d-dimensional vector and is normalized to have a unit length value. Therefore, the value of $X_i * X_j$ can be cast as the cosine similarity between two feature representations of the input videos. The constant δ is used to control the number of edges in the graph.

The temporal-based graph: For modeling the temporal relationship between two nodes representing the segment S_i and S_j, we define the adjacency graph A^T as follows:

$$A_{ij}^T = \begin{cases} 1, & \text{if } S_i \text{ and } S_j \text{ overlapped} \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

In the basic graph-based label propagation problem, the graph $G = (V, E, X)$, where V is the node set of the graph, E is the set of edges that contain both the A^F and A^G and X is the features of the nodes. The GTN architecture proposed by Yun et. al. [14] employs the concept of meta-path which connect through multiple relationships in graph G. The meta-path is

combined from different adjacency matrixes with attention-based softmax:

$$Q = \phi\left(\sum_l a^l A^l * \sum_k \beta^k A^k\right) \quad (4)$$

Algorithm 2 Noisy labels propagation by GTN

Input:
- S_1, S_2, ..., S_N: Segmented videos from S with weak label L
- C_θ: Action classifier deep learning model with parameters θ
- T: Weak signal filter threshold
- K: Number of filtering steps
- $lowT$, $highT$: The low and high threshold

Output:
- Return a trained C_θ model with noise labels

Algorithm:
1. If label L=0, assign label $Y_i = 0$
2. If label L=1, for each step k from 1..K
a. Generate $P_i = P(Y_i = 1|S_i, \theta)$
b. Compute noisy labels: $\gamma_i = 1$ if $P_i \geq highT$, $\gamma_i = 1$ if $P_i \leq lowT$
c. Label propagation with GTN by (S_i, γ_i)
d. Update new P_i with output O_i of GTN: $P_i = (P_i + O_i)/2$
e. Compute Y_i using P_i with a threshold of T
f. Train θ with the new list of (S_i, Y_i) data

More precisely, ϕ denotes the channel attention pooling over a convex combination of the based relationship A^l and A^k. Each layer in the GTN would perform a channel-based attention over the C channels of the input adjacency matrix. Therefore, the output of (4) would be an $N \times N \times C$ matrix. In the output layer, the representation H would be a concatenation of every C channel through a layer of graph convolution layer as follows:

$$H = concat\left(D_c^{-1}\hat{A}_c^L XW\right) \quad (5)$$

Given a channel c, with A^L is the combination over L layers of channel-based attention and Dis the normalized degree matrix of A^L. The subscript c denotes the concatenated index over the C channels.

Equation 5 is a standard GCN layer [10] that allows the projecting of the input feature X into the feature representation, which is an h-dimensional space, and learning the target output. In our label propagation, H would be combined with a classifier layer for filtering noise from the output labels of the action classifier C_θ. The noise filtering algorithm by GTN will differ mainly from the original label filtering in Algorithm 1 by using two more thresholds *lowT* and *highT* to identify labels with high confidence and propagation of their labels by GTN. The *lowT* is the low threshold and is used to identify the normal label, label 0. The *highT* is the low threshold and is used to identify the anomaly label, label 1. The detailed algorithm is described as in Algorithm 2.

The main differences between Algorithm 1 and Algorithm 2 are steps 2b, 2c, and 2d. We start by selecting the *lowT* and *highT* as thetwo ends of the region [0, 1], which reflects the labels with high confidence. The noisy labelsγ_i are then propagated to all the samples of a given full-length video through the learning of GTN. The outputs of GTN, O_i, are combined with the prediction output C_θ of to generate the labels to train the C_θ itself. Starting from step 2e, Algorithm 2 follows Algorithm 1 for training a new filter round with C_θ.

While the usage of a feature-based graph and temporal graph is similar to the study in [9], our works emphasize the effect of label propagation to filter out the noisy signal. In addition to that, we focus on learning a joint representation of the two graphs through the GTN architecture rather than a simple pooling operator as proposed in [9].

4. Experiment Results and Discussions

4.1. Testing Dataset

In this section, we evaluate our proposed model on three benchmark datasets for video anomaly detection including USCD Ped 1, USCD Ped 2, and Avenue dataset[5].

USCD Ped 1/Ped2: In the work of Lu et. al [6], both datasets are split into training videos, which contain only normal actions, and testing videos which contain abnormal actions. The background scene is pedestrians walking on the pavement. Meanwhile, abnormal activities include a wide range of actions such as driving a car, biking, or running actions. Each video in both the training and testing set contains 200 frames. In the USCD Ped 1 set, there are a total of 30 training videos with normal actions and 26 testing videos. In the USCD Ped 2 set, there are a total of 16 training videos and 12 testing videos. The frames are black and white. The image size in USCD Ped 1 is 158x238 and in USCD Ped 2 is 240x360. The frames with abnormal actions are labeled and appeared only in the test set.

Avenue dataset [5]: The Avenue dataset is recorded in the hallway. The normal actions contain people walking individually or in a group. The abnormal actions can be divided into several groups such as strange actions, wrong movement direction, or people with abnormal objects. The data is split into a training video set and a testing video set. There are 16 training videos and 21 testing videos. Meanwhile, each frame in the testing video set contains a mask that specified the region of anomaly action. The length of each video in the training set and testing set varies. The training video set only contains normal actions. The input frames are given with RGB channels and 360x640 image sizes.

As our works focus on a weak-label approach, we modify the standard splitting of the train/test dataset on the three datasets with a new training/testing video set. More specifically, for each of the three datasets USCD Ped 1, USCD Ped 2, and Avenue, we randomly select half of the training videos and 5 testing videos to create a weekly training dataset. The remaining videos for the original training and testing set are grouped into testing videos. To make the evaluating pipeline more stable, each new split is fixed by a given seed and we carry out our evaluation on three different seeds.

For training the abnormal video classifier from the weak label training set, each full video in the training and testing set is split into window size of W=8 frames with a step length of two frames. In the training set, each segmented video is associated with a weak label, which correspondent to the weak label of the original video. In the testing set, the label of segmented videos is decided by whether the segmented video contains abnormal actions or not. A long video from the original weak label abnormaltest set would correspond to several small, sub-segment videos with label normal and anomaly new testing samples.

4.2. Preprocessing and hyper-parameters

Preprocessing: We follow the same procedure for preprocessing each input frame which is to resize the frames into 224x224. With the USCD Ped 1/Ped 2, as input images are gray images, we convert them into 3-channel images by multiple the gray images into three channels Red, Green, and Blue. With the Avenue dataset, input images are given with three-channel RGB, therefore we keep the same images and only use resize preprocessing. We further scale the value to a float in [0, 1] and normalize to zeros mean and standard deviation 1 by:

$$mean = [0.485, 0.456, 0.406]$$

$$std = [0.229, 0.224, 0.225]$$

This is the normal standard procedure for working with input images.

For learning video features and training the anomaly video classification, the base classifier TAM model [18] is used. It is trained on dataset Something-Something V2 (SSV2), Kinetics-400 (Kinetics), and Moments-in-time (MiT). The model is used to generate the video features and training with noisy labels. The number of output features is $d=2048$. With each filter round, we trained the TAM on two epochs with the new filtered labels. From our observation, it is not profitable for training longer as the models are easy to overfit to noise signal rather than the signal of anomaly. We used the Adam optimizer

with learning rate 1e-4 and Binary Cross Entropy for learning the anomaly actions (Label 1) against the normal actions (Label 0).

From the baseline feature extractions and video classifications, we implement a GTN architecture with 2 layers for the filtering step in Algorithm 2. The number of hidden features is set at h=256. With each filtering step, we use the *lowT* and *highT* threshold at the percentile of 20th and 95th, respectively. We lower the percentage of the high confidence anomaly label as it is expected that the number of anomaly actions would be lower than the normal ones. More specifically, in each filter round, 25% of labels are considered as high confidence outputs and used to propagate the labels to the remaining nodes in the graph. For the label propagation apart, we trained the GTN with 10 epochs and then used the output probabilities to assign the label for the remaining 75% of datasets. The optimizer is also Adam with a learning rate of 1e-4 and Binary Cross Entropy loss.

The updated probability from step 2e in Algorithm 2 is used for learning with noisy labels. A threshold value T is used to split the output probability from the baseline predictions into noisy Label 0/1. In our experiment, T is set to 80 percentiles. For comparison purposes, we test our frameworks on 4 baselines including the base self-supervised training in Algorithm 1 and the GCN filter for weak-supervised anomaly video classification as follows:

Baseline: The baseline illustrates Algorithm 1. In this approach, we do not use any noisy label propagation. The output probability P_i is calculated directly from the output of the TAM model. All other training parameters are set the same as the GTN approach.

GCN: We adapt the approach in [9] with GCN for label propagation in the filtering step 2d in Algorithm 2. The settings of other hyper-parameters such as *lowT*, *highT*, training epochs, and optimizers are kept the same as GTN. The layers GCN is set at 2 layers with. We also keep the same dimension of *h=256* for hidden embedding in GCN label propagation, which is the same as GTN.

Logistic/MLP: we employ two simple learning models in the label propagation process, which are Logistic Regression and Multi-layer Perceptron (MLP). The two models learn directly the noisy labels without building the graph.

Implementation: All of our models, TAM and GCN, GTN are set up with Python 3.7, Pytorch, and torch-geometry on a workstation with NVIDIA GPU 3090.

4.3. Results and Discussion

We report the AUC (Area-Under-The Curve) on three datasets with the mean and standard deviation of 3 different seeds.

From Table 1, we can see that our proposed GTN outperforms the four baseline methods, which includes Baseline, Logistic, MLP, and GCN on the USCD Ped 1 and Avenue dataset. Both the USCD Ped 1 and Avenue datasets come with a larger training and testing set. They also include more complicated anomaly actions in the extracted frames. By letting the label propagation through the feature-based graph and the temporal graph, both GTN and GCN can improve the AUC score from the baseline. Moreover, as the GTN can exploit the joint representation from the two graphs, it can learn to detect the anomaly better than the GCN counterpart. In the case of USCD Ped 2, both the GTN and GCN have comparable performance and have a slightly improve from the Baseline performance. The Logistic and MLP approaches are not suitable for propagating noisy labels. In comparison with similar works in [5] or [27], we note that our works use a different train/test split. Therefore, the AUC results of the detection frame do not directly comparable. In addition to that, the propagation noisy labels approach uses the week-supervised approach which can learn significantly faster than the image-reconstruction approach like in [27].

Table 1. The average AUC score of each models on three datasets

Model	USCD Ped 1	UCSD Ped 2	Avenue
Baseline	0.57 ± 0.03	0.84 ± 0.09	0.69 ± 0.02
Logistic	0.58 ± 0.04	080 ± 0.06	0.69 ± 0.01
MLP	0.59 ± 0.09	0.80 ± 0.03	0.70 ± 0.01
GCN	0.59 ± 0.05	**0.85 ± 0.05**	0.70 ± 0.03
GTN	**0.63 ± 0.02**	**0.85 ± 0.08**	**0.72 ± 0.02**

Figure 3. An example frame from a segmented video is an abnormal action where Baseline, GCN missing the prediction (predict 0) and our proposed GTN can predict the true label 1. The left is in the input frame and the right figure is the mask.

Figure 4. An example frame from a segmented video with two abnormal actions where Baseline, GCN, and GTN can predict the label 1. The left is in the input frame and the right figure is the mask.

We plot two examples from the USCD Peds 2 dataset in Fig. 3 and Fig. 4. In Fig. 3, both three models can predict the anomaly actions. However, in Fig. 4, both the Baseline and the GCN approaches fail to predict the anomaly action. From the plot, it can be seen that the GTN approach performs well when there is a small outlier signal from video embedding features. In this case, the GTN model is able to classify the Fig. 4 which contains only one region of the anomaly action.

Table 2. The AUC score of using different the numberof hidden features on the GTN layers

Number of hidden features	USCD Ped 1	UCSD Ped 2
h = 64	0.62	0.80
h = 128	0.64	0.75
h = 256	0.64	0.85
h = 512	0.59	0.84
h = 1024	0.60	0.81
h = 2048	0.59	0.80

To understand more about the effect of GTN architecture on label propagation, we first alternate the number of hidden features in the GTN layers and report the AUC on the USCD Ped 1, and USCD Ped 2 datasets (Table 2). From the results, we can see that the features of the hidden layer in GTN be most stable at 256 dimensions for learning the anomaly actions on both USCD Ped 1 and USCD Ped 2 datasets. Given that the output feature of the used video classifier θ is 2048, keeping the same dimension can hurt the performance of the classifier. It can be explained that the learned features for recognition actions are not well represented in the anomaly properties in the weakly supervised setup. The GTN has to rely on the temporal and feature relation to find the best-projected dimension for filtering noise. The next best candidates for the number of dimensions are *h=64* and *h=512*.

Table 3. Ablation study on the effect of different graph representations in GTN

Model	USCD Ped 1
GCN	0.59 ± 0.05
GTN-A	0.59 ± 0.05
GTN-T	0.56 ± 0.03
GTN	**0.63 ± 0.02**

In Table 3, we run an ablation study using only one of the two graphs in the noisy label propagation phase on the USCD Peds 1. The version GTN-A denotes the approach when only the adjacency matrix in the feature embedding

domain is used in learning with GTN. The version GTN-T denotes the approach when only the temporal relationship is used in the learning with GTN. While the GTN-A can perform as well as the GCN baseline, the GTN-T results in lower performance. By combining the two graphs, our GTN approach can improve the detection results.

Table 4. Training performance comparison of the two training approaches, **Baseline** and **GTN**

Model	Train Time	Label Propagation	Increase
Baseline	95s	-	-
GTN	95s	20s	21%

For the comparison aspect of our performance to the standard baseline model TAM, we have added a training time per epoch for each of the testing models in Table 4. The table showsthat the label propagation approach base on GTNonly takes around 15-20% more computing timethan the baselines. Moreover, the computation overhead only happens in the training phase. During the testing phase, it would not affect the running time of the proposed methods.

5. Conclusions

In this paper, we present a label propagation based on GTN architecture for improving the detection of anomaly videos. Our approach builds on the combination of a deep network for an action classifier and a filtering phase. By combining the feature-based graph and temporal graph, the GTN architecture can identify the anomaly events more efficiently than other baselines. On the AUC score, our proposed approach could improve the accuracy of the weakly supervised approach significantly. Given the flexibility of the proposed method, our work can extend to the domain of unsupervised learning for detecting anomaly videos.

References

[1] B. Ramachandra, M. J. Jones, R. R. Vatsavai, A Survey of Single-scene Video Anomaly Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 44, No. 5, 2020, pp. 2293–2312, https://doi.org/10.1109/TPAMI.2020.304059.

[2] B. Mohammadi, M. Fathy, M. Sabokrou, Image/Video Deep Anomaly Detection: A Survey, arXiv Preprint (2021), https://doi.org/10.48550/arXiv.2103.01739.

[3] V. Saligrama, J. Konrad, P.-M. Jodoin, Video Anomaly Identification, IEEE Signal Processing Magazine, Vol. 27, No. 5, 2010, pp. 18–33, https://doi.org/10.1109/MSP.2010.937393.

[4] Y. Kong, J. Huang, S. Huang, Z. Wei, S. Wang, Learning Spatiotemporal Representations for Human Fall Detection in Surveillance Video, Journal of Visual Communication and Image Representation, Vol. 59, 2019, pp. 215–230, https://doi.org/10.1016/j.jvcir.2019.01.024.

[5] W. Li, V. Mahadevan, N. Vasconcelos, Anomaly Detection and Localization in Crowded Scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 36, No. 1, 2013, pp. 18–32, https://doi.org/10.1109/TPAMI.2013.111.

[6] C. Lu, J. Shi, J. Jia, Abnormal Event Detection at 150 Fps in Matlab, Proceedingsof the IEEE International Conference on Computer Vision, 2013, pp. 2720–2727, https://doi.org/10.1109/ICCV.2013.338.

[7] W. Liu, W. Luo, Z. Li, P. Zhao, S. Gao, et al., Margin Learning Embedded Prediction for Video Anomaly Detection with A Few Anomalies, International Joint Conferences on Aritificial Intelligent, 2019, pp. 3023–3030. https://doi.org/10.24963/ijcai.2019/419.

[8] W. Ren, Y. Li, H. Su, D. Kartchner, C. Mitchell, C. Zhang, Denoising Multi-source Weak Supervision for Neural Text Classification, Findings of the Association for Computational Linguistics, EMNLP 2020, Association for Computational Linguistics, Online, 2020, pp.3739-3754, https://doi.org/10.18653/v1/2020.findings- emnlp.334.

[9] J.-X. Zhong, N. Li, W. Kong, S. Liu, T. H. Li, G. Li, Graph Convolutional Label Noise Cleaner: Train a Plug- and-play Action Classifier for Anomaly Detection, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition 2019, pp. 1237–1246. https://doi.org/10.1109/CVPR.2019.00133.

[10] T. N. Kipf, M. Welling, Semi-supervised Classification with Graph Convolutional Networks, arXiv preprint arXiv:1609.02907 (2016), https://doi.org/10.48550/arXiv.1609.02907.

[11] D. Tran, L. Bourdev, R. Fergus, L. Torresani, M. Paluri, Learning Spatiotemporal Features with 3d Convolutional Networks, Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition, 2015, pp. 4489–4497. https://doi.org/10.1109/ICCV.2015.510.

[12] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, L. Van Gool, Temporal Segment Networks for Action Recognition in Videos, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 41, No. 11, 2018, pp. 2740-2755, https://doi.org/10.1109/TPAMI.2018.2868668.

[13] X. Wang, A. Gupta, Videos As Space-time Region Graphs, Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 399–417. https://doi.org/10.1007/978-3-030-01228-1 ₂5.

[14] S. Yun, M. Jeong, R. Kim, J. Kang, H. J. Kim, Graph Transformer Networks, Advances in Neural Information Processing Systems, Vol. 32, (2019), https://doi.org/10.48550/arXiv.1911.06455.

[15] L. Kratz, K. Nishino, Anomaly Detection in Extremely Crowded Scenes Using Spatio-temporal Motion Pattern Models, Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition 2009, pp. 1446–1453, https://doi.org/10.1109/CVPR.2009.5206771.

[16] M. Ravanbakhsh, M. Nabi, H. Mousavi, E. Sangineto, N. Sebe, Plug-and-play Cnn for Crowd Motion Analysis: An Application in Abnormal Event Detection, Proceedings of the 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE, 2018, pp. 1689-1698. https://doi.org/10.1109/WACV.2018.00188.

[17] C. He, J. Shao, J. Sun, An Anomaly-introduced Learning Method for Abnormal Event Detection, Multimedia Tools and Applications, Vol. 77, 2018, pp. 29573–29588. https://doi.org/10.1007/s11042- 017-5255-z.

[18] W. Liu, H. Chang, B. Ma, S. Shan, X. Chen, Diversity-measurable Anomaly Detection, arXiv Preprint arXiv:2303.05047 (2023), https://doi.org/10.48550/arXiv.2303.05047.

[19] Q. Fan, C.-F. R. Chen, H. Kuehne, M. Pistoia, D. Cox, More Is less: Learning Efficient Video Representations by Big-little Network and Depthwise Temporal Aggregation, Advances in Neural Information Processing Systems, Vol. 32, (2019), https://doi.org/10.48550/arXiv.1912.00869.

[20] K. Liu, W. Liu, C. Gan, M. Tan, H. Ma, T-C3d: Temporal Convolutional 3d Network for Real-time Action Recognition, Proceedings of The AAAI Conference on Artificial Intelligence, Vol. 32, 2018, https://doi.org/10.1609/aaai.v32i1.12333.

[21] H. Zhou, D. Ren, H. Xia, M. Fan, X. Yang, H. Huang, Ast-gnn: An Attention-based Spatio-temporal Graph Neural Network for Interaction-aware Pedestrian Trajectory Prediction, Neurocomputing, Vol. 445, 2021, pp. 298–308, https://doi.org/10.1016/j.neucom.2021.03.024.

[22] W. Wang, J. Gao, X. Yang, C. Xu, Learning Coarse-to-fine Graph Neural Networks for Video-Text Retrieval, IEEE Transactions on Multimedia, Vol. 23, 2020, pp. 2386–2397.

[23] M. Gori, G. Monfardini, F. Scarselli, A New Model for Learning in Graph Domains, Proceedings of the 2005 IEEE International Joint Conference on Neural Networks, 2005., Vol. 2, IEEE, 2005, pp. 729–734, https://doi.org/10.1109/IJCNN.2005.1555942.

[24] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals,G. E. Dahl, Neural Message Passing for Quantum Chemistry, Proceedings of the International Conference on Machine Learning, PMLR, 2017, pp. 1263–1272, https://doi.org/10.48550/arXiv.1704.01212.

[25] J. Bruna, W. Zaremba, A. Szlam, Y. Lecun, Spectral Networks and Locally Connected Networks on Graphs, Proceedings of the International Conference on Learning Representations (iclr2014), CBLS, April (2014), https://doi.org/10.48550/arXiv.1312.6203.

[26] M. Defferrard, X. Bresson, P. Vandergheynst, Convolutional Neural Networks on Graphs with Fast Localized Spectral Filtering, Advances in Neural Information Processing Systems, Vol. 29, (2016), https://doi.org/10.48550/arXiv.1606.09375.

[27] P. Veli˘ckovi´c, G. Cucurull, A. Casanova, A. Romero, P. Lio, Y. Bengio, Graph Attention Networks, arXiv preprint arXiv:1710.10903 (2017).

[28] V. P. Dwivedi, X. Bresson, A Generalization of Transformer Networks to Graphs, arXiv preprint arXiv:2012.09699 (2020).

[29] Q. Li, Z. Han, X.-M. Wu, Deeper Insights into Graph Convolutional Networks for Semi-Supervised Learning, Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 32, 2018, https://doi.org/10.1609/aaai.v32i1.11604.

[30] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, K. Weinberger, Simplifying graph Convolutional Networks, International Conference on Machine Learning, PMLR, 2019, pp. 6861–6871, https://doi.org/10.48550/arXiv.1902.07153.