Confidence-Aware Anomaly Detection in Human Actions

Tsung-Hsuan Wu(✉), Chun-Lung Yang, Li-Ling Chiu, Ting-Wei Wang, Gueter Josmy Faure, and Shang-Hong Lai

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan
th.wu@mx.nthu.edu.tw

Abstract. Anomaly detection in human actions from video has been a challenging problem in computer vision and video analysis. The human poses estimated from videos have often been used to represent the features of human actions. However, extracting keypoints from the video frames are doomed to errors for crowded scenes and the falsely detected keypoints could mislead the anomaly detection task. In this paper, we propose a novel GCN autoencoder model to reconstruct, predict and group the poses trajectories, and a new anomaly score determined by the predicted pose error weighted by the corresponding confidence score associated with each keypoint. Experimental results demonstrate that the proposed method can achieve state-of-the-art performance for anomaly detection from human action videos.

Keywords: Video anomaly detection · Human pose · GCN · Confident scores

1 Introduction

Anomaly detection in human activities from videos is a challenging research problem that attracts great attention from academia and industry in recent years. Due to the lack of abnormal data, anomaly detection is usually treated as an unsupervised task. Since almost all the training data is assumed to be normal, models are trained to learn the characteristics from the normal data, such as grouping, reconstructing the data, or predicting the future frames in a video. Since this skill is only learnt from normal data, it should work badly on abnormal data so that anomalies can be detected.

Due to the rarity of the abnormal data, anomaly detection is usually treated as an unsupervised task. With all (or most of) the training data are normal, the model is asked for learning a "skill" such as grouping [1–3], reconstructing the data [2–7], or predicting the futures [6–8] of the videos. Since this skill is only

S.-H. Lai—We thank to National Center for High-performance Computing (NCHC) for providing computational and storage resources.

C. Wallraven et al. (Eds.): ACPR 2021, LNCS 13188, pp. 240–254, 2022.
https://doi.org/10.1007/978-3-031-02375-0_18

learnt from normal data, it might work badly on the anomalies. As a result, the normality can be classified.

Dealing with events in videos, most recent methods focus on the appearance of the time-consecutive images [2,5,6], while the proposed method is focused on the human poses. Human poses are often used in analysis of human behaviors. [9] used pose features to recognize human actions. [10] estimated 3D-poses from 2D-poses. The pose data has the advantage that only the human features are used in the models. However, while extracting the keypoints from the images, the pose estimation may involve large errors for crowded scenes, which makes the follow-up works very difficult. To alleviate this problem, we add the confidence scores into our model so the result will be less sensitive to the errors in keypoint detection.

To classify the normal and the abnormal human actions in the videos, we reconstruct, predict and group the poses trajectories. Considered poses as graphs, our proposed model is based on the graph convolutional networks (GCN), which is recently applied to many deep learning methods on pose data. To group the poses trajectories, we use a multi-centers loss to gather the latent features to their corresponding centers. Moreover, to score the abnormality of the poses, we propose a new method, applying the confident scores, to evaluate the error between the output and the original keypoints.

The main contributions of this work are listed as follows:

- We propose a *temporal poses autoencoder* (TP-AE) framework (Fig. 1) to reconstruct and predict the poses trajectories.
- We propose the multi-centers loss function to cluster the latent features of the poses trajectories.
- We propose a novel anomaly score involving the confidence scores of pose estimation, to reduce the influence brought from the falsely detected keypoints.

2 Related Work

Anomaly Detection on Image Data. Anomaly detection on image data is usually considered as an one-class classification problem. Lukas *et al.* in [1] used a CNN model to gather the features of the normal images to a center and then detected the outliers by the distance between the features and the center. Schlegl *et al.* in [4] used a GAN model so that the generator learnt to construct, from the random latent vectors, the images similar to the normal class, while the abnormal images would not able to be constructed.

Appearance-based Anomaly Detection on Video Data. To detect anomaly events in videos, most of the current works are based on the appearance of the frames. To extract the temporal information, the motion information such as optical flows are often used. Liu *et al.* in [5] used a CNN U-net model to predict the future image from a sequence of frames. To improve the prediction, the intensity difference, gradient difference, optical flows and the GAN structure

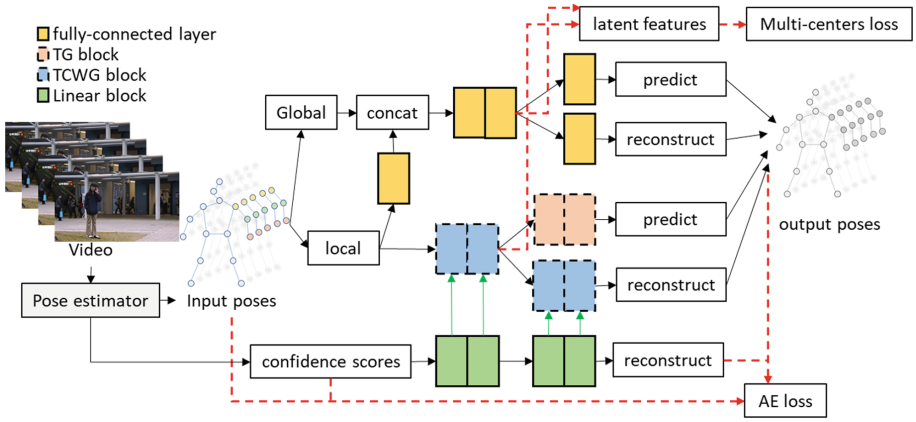


Fig. 1. Architecture of TP-AE model. After the pose estimator evaluates the keypoints and their confidence scores, the poses are separated into global and local information. The yellow pipeline is consisting of fully-connected layers and is to reconstruct and predict the global information. The blue and red parts are the GCN-based layers which reconstruct and predict the local information. The green pipeline, based on linear functions, produces weights to assist the TCWG blocks. The total loss function is composed of the errors of the reconstruction/prediction and the loss of multi-centers clustering.

are also used. Nguyen *et al.* in [6] predicted the optical flow by a single frame. Chang *et al.* in [2] predicted the difference between frames and reconstructed, clustered them in the same time.

Pose-based Anomaly Detection on Video Data. As the appearance-based methods, pose-based anomaly detection also focused on reconstruction or prediction. Morais *et al.* in [7] first separated the poses into global and local information. Then the RNN-based model reconstructed the input poses and predicted the future poses. Based on 1D-CNN, Rodrigues *et al.* in [8] proposed a multi-timescale model to make future and past predictions at different timescales. Markovitz *et al.* in [3] used GCN-based autoencoder to reconstruct the poses trajectories, while the point of this work was the grouping of the latent features by the Dirichlet process mixture model (DPMM). Different from the above error-based anomaly scores, the normality score of each sample was computed by its log probability using the fitted model.

Referring to [7] and [8], our model also reconstructs the inputs and predicts the future pose trajectories, while different from them, we use GCN-based networks and apply confidence scores of the keypoints to the networks and to the computation of anomaly scores. Moreover, we add a loss function to group the pose trajectories unsupervisedly.

3 Proposed Model

In this section, we present the details of our method. As [7] and [8], we estimate the human poses trajectories by applying poses detector and human tracker to the dataset. In addition, we preserve the confidence scores of the keypoints for later usage. To detect anomalies, we first train a model to reconstruct and predict poses on the training videos which contain only normal human behaviors. Then we apply the model to the testing videos to detect anomaly behaviors by the performance of the reconstruction and prediction. Instead of RNN or CNN, our proposed model, temporal poses autoencoder (TP-AE), is based on GCN and applies confidence scores to alleviate the problems of missing nodes. Moreover, to further improve our model, we use the multi-centers SVDD loss function to group the latent vectors. Finally, applying the confidence scores again, we propose a new score to detect anomalies.

3.1 Confidence-weighted Graph Convolution Network

To reduce the influence of the keypoints with low confidence scores, we use the confidence-weighted graph convolution network (CWGCN) [11]. Before we introduce CWGCN, let's recall the function of the classical GCN. For a graph $G = (V, E)$, where V is the set of N vertices and E are the edges, including all the self-loops. Then E induces an adjacency matrix $A \in \{0,1\}^{N\times N}$. Let $X \in \mathbb{R}^{F\times N}$ be the input of the GCN, where F is the dimension of the feature vector of each vertex. Then the output of the GCN is

$$\overline{X} = WX\widetilde{A}\ , \tag{1}$$

where $W \in \mathbb{R}^{F'\times F}$ is a learnable matrix, F' is the number of features of the next layer, and $\widetilde{A}$ denotes the column normalization of A, that is, for each i, j,

$$\widetilde{A}_{i,j} = \frac{A_{i,j}}{\sum_{k=1}^{N} A_{k,j}}\ . \tag{2}$$

In (1), we can observe that the output feature vector of the j-th vertex is $\overline{X}_{\_,j} = W(X\widetilde{A})_{\_,j}$, where

$$(X\widetilde{A})_{\_,j} = \frac{1}{\sum_{k=1}^{N} A_{k,j}} \sum_{i=1}^{N} X_{\_,i}A_{i,j} \tag{3}$$

is the average of $\{X_{\_,i}|(i,j) \in E\}$; in the other words, all the jointed vertices have the same influence on $\overline{X}_{\_,j}$.

However, in case some input vertices have lower confidence scores, those unreliable features shouldn't have the same influence as others. Therefore, we take the confidence score as a weight to indicate the influence of each vertex. Let $\{c_i\}_{i=1}^{N}$ be the confidence scores and $C = diag\{c_i\}$. Then the output of the CWGCN is

$$\overline{X} = WX\widetilde{CA}\ , \tag{4}$$

where $\widetilde{CA}$ denotes the column normalization of CA, that is,

$$\widetilde{CA}_{i,j} = \frac{c_i A_{i,j}}{\sum_{k=1}^{N} c_k A_{k,j}} \ . \tag{5}$$

In this case, $\overline{X}_{\_,j}$ is determined by

$$(X\widetilde{CA})_{\_,j} = \frac{1}{\sum_{k=1}^{N} c_k A_{k,j}} \sum_{i=1}^{N} X_{\_,i} c_i A_{i,j} \ , \tag{6}$$

which is the weighted average of $\{X_{\_,i} | (i,j) \in E\}$.

3.2 Temporal Confident Weighted Graph Convolution

To consider a single pose $P = \{P_i | 1 \leq i \leq N\}$ as a graph, it is natural to regard the keypoints as vertices and the skeletons as edges, including all the self-loops. However, when things comes to a sequence of poses, various strategies might be taken. Our definition is as follow.

First we define some notations. Let $S \subset N^2$ denote the set of indices of the skeletons of a pose, that is, $(i,j) \in S$ if the i-th and the j-th keypoints are jointed. For a poses trajectory $\{P_t\}_{t=1}^{T} = \{P_{t,i} | 1 \leq t \leq T, 1 \leq i \leq N\}$, we define the temporal poses graph by $TG = (\{P_{t,i}\}, TS)$, where $P_{t,i}$ is the i-th keypoint in the time t and

$$TS = \{(t_1, i, t_2, j) | t_1 \leq T, t_2 \leq T, (i,j) \in S\} \ . \tag{7}$$

In other words, a keypoint $P_{t_1,i}$ is jointed to every $P_{t_2,j}$ with $(i,j) \in S$, for any t_1 and t_2.

Next, let P^1 and P^2 be two poses trajectories with time-lengths T_1 and T_2, respectively. For further usage, we are going to define the "temporal edges" between P^1 and P^2 by an analogous definition of TS above, that is, the keypoints $P^1_{t_1,i} \in P^1$ and $P^2_{t_2,j} \in P^2$ are jointed if $(i,j) \in S$. More specifically, the temporal edges are defined by

$$TE = \{(t_1, i, t_2, j) | t_1 \leq T_1, t_2 \leq T_2, (i,j) \in S\} \ . \tag{8}$$

Note that the adjacency matrices of TE can be simply derived by A^S, the adjacency matrix of S. First we rearrange the index of the keypoints by

$$P_{t,i} \sim P^{rearrange}_{(t-1)*N+i} \ . \tag{9}$$

Then if $A^S_{i,j} = 1$, that is, $(i,j) \in S$, we have $P^{rearrange}_{(t_1-1)*N+i}$ and $P^{rearrange}_{(t_2-1)*N+j}$ are jointed for any t_1 and t_2. Therefore, A^{TE}, the adjacency matrix of TE, is a $(T_1 * N)$ by $(T_2 * N)$ matrix generated by repeating A^S $(T_1 \times T_2)$ times. For

example, if $N = 3$, $A^S = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 1 \end{pmatrix}$, $T_1 = 2$ and $T_2 = 3$, then

$$A^{TE} = \begin{pmatrix} 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 0 & 1 & 1 & 0 & 1 & 1 \end{pmatrix} . \tag{10}$$

Following the above result, if we apply A^{TE} to GCN or CWGCN, the time-lengths of the input trajectory and the output trajectory can be different. This benefits the GCN-based models when doing poses prediction or using an autoencoder architecture. However, it also leads to a problem that the outputs $\overline{X}_{-,(t_1-1)*N+i}$ and $\overline{X}_{-,(t_2-1)*N+i}$ are the same for any t_1 and t_2 since $A^{TE}_{i,(t_1-1)*N+i} = A^{TE}_{i,(t_2-1)*N+i}$, which means the output poses are the same regardless of time.

To correct this problem, we design the *temporal confidence weighted graph convolution network* (TCWGCN) as follows. Let $X \in \mathbb{R}^{F \times T_1 N}$ be the input of the TCWGCN and $C \in \mathbb{R}^{T_1 N}$ be the confidence scores, where F is the number of the input features, T_1 is the input time-length and N is the number of nodes. Let F' be the number of the output features, T_2 be the output time-length. The output of the TCWGCN at the time point t is

$$\overline{X^t} = W^t X \widetilde{CA^{TE}} , \tag{11}$$

where each $W^t \in \mathbb{R}^{F' \times F}$ is a learnable matrix for the time t, $1 \leq t \leq T_2$. In addition, we also define the *temporal graph convolution network* (TGCN) to be the TCWGCN with all scores are replaced by 1.0.

3.3 Network Architecture

As [7], our model first separates each pose into global and local parts. Let B be the bounding box of the pose $P = \{P_i\}_{i=1}^N$. The global information of P is a $4-$dimensional vector $P^G = (w, h, x, y)$ formed by the width w, the height h and the center (x, y) of B. Then the local information P^L is obtained by regularizing P by P^G, that is,

$$P_i^L = (\frac{P_{i,x} - x}{w} + 0.5, \frac{P_{i,y} - y}{h} + 0.5), \tag{12}$$

where $(P_{i,x}, P_{i,y}) = P_i$.

As shown in Fig. 1, the TP-AE model contains 3 pipelines: the global pipeline, the local pipeline and the confidence score pipeline. The global and the local pipelines aim to reconstruct and predict the global and the local information, so each contains one encoder and two decoders. On the other hand, the confidence

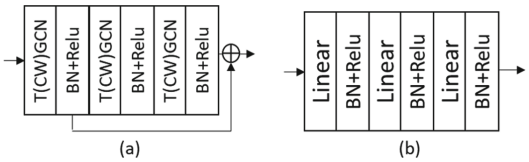


Fig. 2. (a) TCWG and TG block (b) Linear block

score pipeline contains only one encoder and one decoder since the prediction of the confidence scores is unreasonable.

The global encoder and decoders are composed of fully-connected layers, while the local encoder and decoders are composed of TCWG blocks or TG blocks. As shown in Fig. 2 (a), each TCWG block (or TG block) contains three TCWGCNs (or TGCNs). As [10], we use residual blocks, and each TCWGCN is followed by a batch normalization and a ReLU activation. In the encoder, the first TCWGCN in each block may remain or half the time-lengths of the inputs, while in the decoders the first TCWGCN may remain or double the time-lengths.

The confidence score pipeline is almost the same as the local pipeline, except that the TCWGCN's are replaced by linear functions, and there is no residual block (Fig. 2 (b)). As a result, the midterm outputs, as simulated confidence scores, can be passed to all the TCWGCN's in the local pipeline.

3.4 Loss Function

In our model, the total loss function consists of AE-Loss and the multi-centers loss.

The AE-loss is the combination of the loss of pose estimation errors and the loss of the confidence scores. The loss of pose estimation errors is determined by the error between the estimated and the real poses. As [8], the error of a pose is the weighted mean square error

$$e(P) = \sum_{i=1}^{N} \frac{c_i}{\sum_{k=1}^{N} c_K} \left(P_i - \hat{P}_i\right)^2 , \tag{13}$$

where c_i is the i-th confidence score, P_i and $\hat{P}_i$ are the real and the estimated i-th keypoints of P, respectively. Then the loss of pose estimation errors is given by

$$L_p = \frac{1}{|\mathbb{REC}|} \sum_{b \in \mathbb{REC}} e(\hat{P}^b) + \frac{1}{|\mathbb{PRED}|} \sum_{b \in \mathbb{PRED}} e(\hat{P}^b) , \tag{14}$$

where $\mathbb{REC}$ and $\mathbb{PRED}$ are the sets of indices of the reconstructed and the predicted poses.

On the other hand, we use L_2-loss to evaluate the reconstruction loss of the confidence scores L_c.

$$L_c = \frac{1}{|\mathbb{REC}|} \sum_{i=1}^{N} \frac{1}{N} \left(c_i^b - \hat{c}_i^b\right)^2 \ , \tag{15}$$

Then the AE-loss is

$$Loss_{AE} = \lambda_p L_p + \lambda_s L_c \ . \tag{16}$$

Consider a pose trajectory as an action, we propose to include unsupervised action grouping to facilitate the detection of abnormal human actions. The classifier, training by the normal actions, should consider the abnormal actions as outliers of all the groups.

Let $\{y_i\}$ denote the latent features extracted by the encoders, and y_i be the vector formed by concatenating the outputs of the local and global encoders. Inspired by the SVDD loss in [1], we define the multi-center loss function by

$$Loss_{mc} = \frac{1}{n} \sum_{i=1}^{n} \min_{1 \le d \le D} \|y_i - center_d\|_2 \ , \tag{17}$$

where $center_d$ is the center of the d-th group and D is the number of groups. From (17), it is obvious that the action x_i is assigned to the d-th group if $center_d$ is the closest center to x_i's latent feature y_i. While minimizing $Loss_{mc}$, all the latent features of the actions belonging to the d-th group would be gathered to $center_d$. As a result, the model would learn to classify the actions.

In the first 20 training epochs, only $Loss_{AE}$ is used to warm up the model. Then all the $\{center_d\}$'s are initialized by applying K-Means on the latent vectors $\{y_i\}$ and they are updated for every 5 epochs. Then the total loss function is defined by

$$Loss_{total} = Loss_{AE} + \lambda_{mc} Loss_{mc} \ . \tag{18}$$

3.5 Anomaly Detection

Similar to most of the works on anomaly detection of human action videos, we evaluate the accuracy of anomaly detection at frame level. Similar to [7] and [8], the anomaly score of a frame F is the maximum of the anomaly scores for the human poses in this frame, that is,

$$Score(F) = \max_{P_t \in F} score(P_t) \ . \tag{19}$$

In this paper, $score(P_t)$ is composed of the score of errors, $score_e$, and the score of grouping, $score_g$.

Let $P = \{P_t\}_{t=1}^{T_1+T_2}$ be a pose trajectory, where T_1 and T_2 are the time-lengths of inputs and predictions, respectively, and the pose P_t is composed of the keypoints $\{P_{t,k}\}_{k=1}^{K}$. Let $c_{t,k}$ be the confidence score of $P_{t,k}$, both given by the

pose estimator, and let $P_{t,k}^{real}$ denote the "real" location of the k-th keypoint of P_t. In general, lower the confidence score $c_{t,k}$, longer the distance $\|P_{t,k} - P_{t,k}^{real}\|$.

Therefore, we assume that there is a high possibility that $P_{t,k}^{real}$ is lied in a circle with center $P_{t,k}$ and radius $R_{t,k}$, which is inversely proportional to $c_{t,k}$. We call $R_{t,k}$ the *confidence radius* and define it by

$$R_{t,k} = \frac{\beta}{100 * c_{t,k}} \ . \tag{20}$$

Under this assumption, the error of reconstruction or prediction of $P_{t,k}$ should be ignore during anomaly detection if it is less than $R_{t,k}$.

Moreover, let f denote the TP-AE model and $f(P)$ denote the output of f with input $\{P_t\}_{t=1}^{T_1}$. Since P^{real} is around P, the output $f(P^{real})$ should lie in a neighborhood Nb of $f(P)$, while in case P has detection errors, the distance of $f(P^{real})$ and $f(P)$ could be large. Therefore, a large size of the neighborhood Nb should represent a higher "tolerance" of the reconstruction/prediction errors. Now we use $r_{t,k}$ to simulate the radius of the neighborhood of $f(P)_{t,k}$ and call it the *simulated confident radius*.

To estimate $r_{t,k}$, let $P + R$ be the pose trajectory consisting of the poses

$$P_t + R_t = \{P_{t,k} + (R_{t,k}, R_{t,k})\}_{k=1}^{K} \ , \tag{21}$$

that is, $P_t + R_t$ is one of the farthest "neighbor" of P and is standing on the border of the region in where P_t^{real} is lied. Therefore, $f(P+R)$ can be considered as the trajectory staying on the border of the above neighborhood. Then $r_{t,k}$ is defined by

$$r_{t,k} = \|f(P + R)_{t,k} - f(P)_{t,k}\|_1 \ . \tag{22}$$

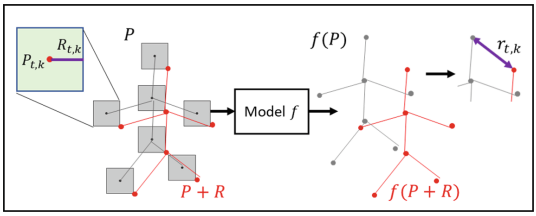


Fig. 3. Confidence radius and simulated confidence radius. After computing the confidence radius of the gray pose trajectory P, the red pose trajectory $P + R$ is one of the farthest "neighbor" of P. Then the simulated confidence radius is given by the distance between $f(P)$ and $f(P + R)$.

Now we define the error of the output $\hat{P}_t$ by

$$e'(\hat{P}_t) = \sum_{k=1}^{K} \frac{\|\hat{P}_{t,k} - P_{t,k}\|_1}{R_{t,k} + r_{t,k}} \ . \tag{23}$$

In addition, to bridge the gap between the large size pose and small size pose, we normalize the pose error by the pose height $h(P_t)$ as follows:

$$\bar{e}(\hat{P}_t) = \frac{e'(\hat{P}_t)}{h(P_t)} \ . \tag{24}$$

Thus, the anomaly score of errors of a pose P_t is defined by

$$score_e(P_t) = \frac{1}{|B|} \sum_{b \in B} \bar{e}(\hat{P}_t^b) \ , \tag{25}$$

where $\{\hat{P}_t^b\}_{b \in B}$ are the reconstructions and the predictions of P_t with the index set B.

On the other hand, since the TP-AE model has learned to group the normal actions by $Loss_{mc}$ (17), an abnormal action can be detected if its latent feature is far away from all the centers. Therefore, the anomaly score of grouping of P_t is defined by

$$score_g(P_t) = \frac{1}{|B|} \sum_{b \in B} \min_{1 \le d \le D} \|y_b - center_d\|_2 \ . \tag{26}$$

Finally, the total anomaly score of pose P_t is given by

$$score(P_t) = \lambda_e score_e(P_t) + \lambda_g score_g(P_t) \ . \tag{27}$$

4 Experiments

Table 1. Frame-level AUC score of the experiments. * is evaluated by [7]'s open-source code.

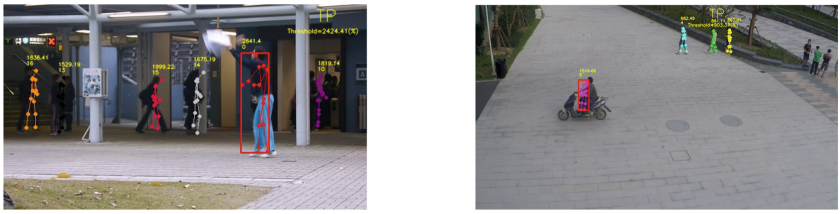
	Method	Avenue	ShanghaiTech
Appearance-based	Chang [2]	86.0	73.3
	Liu [5]	84.9	72.8
	Nguyen [6]	**86.9**	–
Pose-based	Morais [7]	81*	73.4
	Rodrigues [8]	82.9	76.0
	Markovitz [3]	–	76.1
	Ours (a)	81.0	**76.6**
	Ours (b)	85.5	69.5

4.1 Data Preparation

In this paper, we experiment with two of the most widely used datasets for anomaly detection tasks, namely the ShanghaiTech Campus [12] and CUHK Avenue datasets [13]. Each of them presents specific challenges due to their singularity. Here, we present a brief introduction to these datasets.

With 13 different scenes and 130 abnormal events spanning over more than 400 videos, the ShanghaiTech Campus dataset [12] contains more realistic scenarios than other anomaly detection datasets making it very challenging for current anomaly detection models. This dataset is known for its diversity but equally important is its complex light conditions and view angles. Also, it includes new, more challenging anomalies such as chasing and brawling.

Smaller than the ShanghaiTech Dataset, the CUHK Avenue Dataset [13] consists of 37 video clips (16 for training and 21 for testing) that were captured on the CUHK campus avenue, hence the name. The training videos only contain normal situations with a few outliers whereas the testing videos contain both normal and abnormal ones. The dataset includes anomalous behaviors such as people running or moving in an unexpected direction as well as challenging settings like slight camera shake.



(a) (b)

Fig. 4. Examples of the detected anomalies (in the red boxes). (a) In Avenue, the one unnaturally throwing the papers is detected since the model fails to reproduce his pose. (b) In ShanghaiTech, the one riding on the footway is detected since the model fails to adapt his speed. (Color figure online)

To estimate the poses trajectories, [7] first utilized AlphaPose [14–16] to detect poses in the video frames. Then they combined sparse optical flow with the detected skeletons to assign similarity scores between pairs of skeletons in neighboring frames. On the other hand, [8] run a human detector and a multi-target tracker to obtain human trajectories, and run a pose detector to get the poses and the confidence scores.

Different from them, we directly obtain the poses trajectories and the confidence scores by the poses trackers LightTrack [17] or AlphaPose [14–16]. The estimated poses contain 15 keypoints for LightTrack and 17 keypoints for AlphaPose. Due to the crowded and staggered people, in some trajectories part of the poses might missing. Therefore, we construct the missing poses and their confidence scores by interpolation.

4.2 Implementation Details

For Avenue dataset, we apply LightTrack, and set the time-lengths $T_1 = 8$ of the input, $T_2 = 4$ of predicted poses, $D = 5$, the number of the groups, and $\lambda_g = 5$. For ShanhhaiTech, we apply AlphaPose, and set $T_1 = T_2 = 4$, $D = 5$ and $\lambda_g = 0.1$. For both dataset, $\lambda_p = \lambda_s = 1$, $\lambda_{mc} = 0.01$, $\beta = 0.1$ and $\lambda_e = 1$. Moreover, before the total AUC of the 12 scenes in ShanhhaiTech is counting, we linearly transform the anomaly scores of frames in a scene so that the lowest anomaly score is 0 and the top 0.5% score becomes 1.

4.3 Results and Discussion

In Table 1, we compare our method with [2,3,5–8]. The scores represent the frame-level AUC.

(a) (b) (b)

Fig. 5. (a) The running man is a failed case that the pose estimator and the human tracker are failed. (b) The bike is a failed case since our model has no information about it. (c) The dancing man is hard to be considered as anomaly unless he waves arms significantly. The upper images of (b) (c) show the poses given by the pose estimator; the bottom images of (b) (c) show the output poses of our model.

Specially, we present two results for our method. Ours(a) is implemented as mentioned above, while Ours(b) does not use equation (24), that is, the poses errors are not normalized by the poses' heights when computing the anomaly scores, which is similar to the pose-based methods [8] and [8]. In fact, in the Avenue dataset, most of the abnormal people are relatively close to the monitor and have larger sizes. Therefore, (24) will decrease their anomaly scores. As a result, though Ours(a) is more reasonable, it has a lower accuracy of the anomaly detection on Avenue dataset.

[2,5,6] are methods based on the appearance and the optical flows of the frames. By contrast, [3,7,8] and our method are at a disadvantage since there is no information about the abnormal vehicles can be extracted from the human

poses. However, all the pose-based methods perform better on ShanghaiTech and the AUC score of our method is only 1.4 less than [6] on Avenue.

Comparing with previous pose-based methods, our method performs 2.6% better than [8] on Avenue (85.5 vs 82.9), and 0.8% better than [3] on ShanghaiTech (76.6 vs 76.1).

Case Discussion. Fig. 5 shows some failed cases and a hard example.

The anomaly in Fig. 5 (a) is a running man who moves too fast so that the human tracking is failed and since his appearance is a little bit blurred, the pose estimator cannot find him in some frames. As a result, this man is got rid of the pose data in our experiments, so the anomaly detection failed.

The anomaly in Fig. 5 (b) is a man walking a bike. Since he is walking like others and his speed is normal, our model considers his action normal. Therefore, the anomaly detection failed because of lack of information about the bike.

Figure 5 (c) is a hard example. The anomaly is a man dancing in place. Our model detects the anomaly only when he waves his arms. On the other hand, since the appearance is a normal human and his speed is not obvious, it is much more difficult for appearance-based anomaly detection methods, which usually rely on the appearance and the optical flows of the frames.

In conclusion, (a) and (b) shows the weakness of pose-based methods, that is, the abnormal object has to be detected by the pose estimator. On the other hand, (c) shows an advantage of pose-based methods that it helps to find anomalies which are irrelevant to the objects' appearance and speeds. Combining the two types of methods and extracting their respective strengths can be the future research direction.

4.4 Ablation Study

Table 2. AUC of the experiments with/without $loss_{mc}$ and $score_g$, and the experiments of replacing (23) by (13).

Grouping	$score_e$	Avenue	ShanghaiTech
	(13)	76.7	74.0
✓	(13)	77.5	73.3
	(23)	81.1	74.2
✓	(23)	**85.5**	**76.6**

Table 2 depicts the result of the ablation study. We first examine the effect of the multi-center grouping. If the model does not group the latent features, that is, $loss_{mc}$ and $score_g$ are not used, the AUC decreases 4.4% on Avenue and 2.4% on ShanghaiTech.

[8] uses equation (13) to evaluate the anomaly scores from the errors of reconstruction or prediction, so we do the experiments that (23) is replaced by (13). In this case, Table 2 shows that the AUC decrease 8.0% on Avenue and 3.3% on ShanghaiTech.

5 Conclusion

In this work, we present a GCN autoencoder model to reconstruct and predict the pose trajectories in the videos. In this model, we also group the actions by gathering the latent features to the group centers. In addition, we develop the new anomaly score weighted by the confidence radii to detect abnormal human actions. Our experimental results show that we achieve the state-of-the-art accuracy among all the pose-based anomaly detection methods.

References

1. Ruff, L., et al.: Deep one-class classification. In: ICML, vol. 80, pp. 4390–4399. Publisher (2018)
2. Chang, Y., Tu, Z., Xie, W., Yuan, J.: Clustering driven deep autoencoder for video anomaly detection. In: ECCV (2020)
3. Markovitz, A., Sharir, G., Friedman, I., Zelnik-Manor, L., Avidan, S.: Graph embedded pose clustering for anomaly detection. In: The IEEE Conference on Computer Vision and Pattern Recognition (2020)
4. Schlegl, T., Seeböck, P., Waldstein, S.M., Schmidt-Erfurth, U., Langs, G.: Unsupervised anomaly detection with generative adversarial networks to guide marker discovery. In: Niethammer, M., Styner, M., Aylward, S., Zhu, H., Oguz, I., Yap, P.-T., Shen, D. (eds.) IPMI 2017. LNCS, vol. 10265, pp. 146–157. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-59050-9_12
5. Liu, W., Luo, W., Lian, D., Gao, S.: Future frame prediction for anomaly detection - a new baseline. In: The IEEE Conference on Computer Vision and Pattern Recognition (2018)
6. Nguyen, T.N., Meunier, J.: Anomaly detection in video sequence with appearance-motion correspondence. In: The IEEE International Conference on Computer Vision (2019)
7. Morais, R., Le, V., Tran, T., Saha, B., Mansour, M., Venkatesh, S.: Learning regularity in skeleton trajectories for anomaly detection in videos. In: The IEEE Conference on Computer Vision and Pattern Recognition (2019)
8. Rodrigues, R., Bhargava, N., Velmurugan, R., Chaudhuri, S.: Multi-timescale trajectory prediction for abnormal human activity detection. In: The IEEE Winter Conference on Applications of Computer Vision (2020)
9. Yan, S., Xiong, Y., Lin, D.: Spatial temporal graph convolutional networks for skeleton-based action recognition. In: AAAI (2018)
10. Zhao, L., Peng, X., Tian, Y., Kapadia, M., Metaxas, D.N.: Semantic graph convolutional networks for 3D human pose regression. In: IEEE Conference on Computer Vision and Pattern Recognition (2019)
11. Vashishth, S., Yadav, P., Bhandari, M., Talukdar, P.: Confidence-based graph convolutional networks for semi-supervised learning. In: Proceedings of Machine Learning Research, pp. 1792–1801 (2019)

12. Luo, W., Liu, W., Gao, S.: A revisit of sparse coding based anomaly detection in stacked RNN framework. In: ICCV (2017)
13. Lu, C., Shi, J., Jia, J.: Abnormal event detection at 150 FPS in matlab. In: ICCV (2013)
14. Fang, H.-S., Xie, S., Tai, Y.-W., Lu, C.: RMPE: regional multi-person pose estimation. In: ICCV (2017)
15. Li, J., Wang, C., Zhu, H., Mao, Y., Fang, H.-S., Lu, C.: CrowdPose: efficient crowded scenes pose estimation and a new benchmark. arXiv preprint. arXiv:1812.00324 (2018)
16. Xiu, Y., Li, J., Wang, H., Fang, Y., Lu, C.: Pose flow: efficient online pose tracking. In: BMVC (2018)
17. Ning, G., Huang, H.: LightTrack: a generic framework for online top-down human pose tracking. In: The IEEE Conference on Computer Vision and Pattern Recognition Workshops (2020)