Self-Supervised Feature Extraction for Video Surveillance Anomaly Detection

Davi D. de Paula, Denis H. P. Salvadeo, Lucas B. Silva, Uemerson P. Junior
Institute of Geosciences and Exact Sciences, São Paulo State University
Rio Claro, Sao Paulo, Brazil
Email: {davi.duarte, denis.salvadeo, lucas.brito-silva, uemerson.junior}@unesp.br

***Abstract*—The recent studies on Video Surveillance Anomaly Detection focus only on the training methodology, utilizing pre-extracted feature vectors from videos. They give little attention to methodologies for feature extraction, which could enhance the final anomaly detection quality. Thus, this work presents a self-supervised methodology named Self-Supervised Object-Centric (SSOC) for extracting features from the relationship between objects in videos. To achieve this, a pretext task is employed to predict the future position and appearance of a reference object based on a set of past frames. The Deep Learning-based model used in the pretext task is then fine-tuned on Weak Supervised datasets for the downstream task, using the Multiple Instance Learning training strategy, with the goal of detecting anomalies in the videos. In the best case scenario, the results demonstrate an increase of 3.1% in AUC on the UCF Crime dataset and an increase of 2.8% in AUC on the CamNuvem dataset.**

I. Introduction

The research area of video surveillance anomaly detection is a promising field that aims to detect events that deviate from the normal pattern captured by a specific camera. These events are referred to as anomalies or abnormalities, and their exact appearance can vary across different application domains. Examples of anomalies include car crashes, fights, robberies, assaults, and more.

Multiple Instance Learning (MIL) [1] is an approach used to detect such anomalies that uses both normal and abnormal videos into the training process. The MIL approach relies on weakly labeled videos, where each video is labeled only to indicate if it contains an anomaly without specifying the exact location and duration. In their work, Ref. [1] also introduced the weakly labeled dataset UCF-Crime, which comprises hundreds of real videos encompassing various anomaly categories.

In the past few years, several studies have utilized the MIL approach to propose more robust models, cost functions, and learning strategies. However, all of them have not focused on the feature extraction step. Instead, they have employed pre-trained feature extractors such as I3D [2], Slow-Fast [3], C3D [4], etc. The feature extraction step is a crucial process because it determines the video representation used in the anomaly detection process and directly influences the quality of the final results. Using weights from models trained on other datasets may lead to suboptimal results. Hence, this work proposes a self-supervised approach called Self-Supervised Object-Centric (SSOC) video anomaly detection to pre-train a feature extraction model using a video dataset.

It is important to note that this work focuses only on the feature extraction step and does not propose a MIL approach. Instead, it employs existing video surveillance anomaly detection approaches to evaluate the video features extracted by SSOC on a weakly labeled video anomaly detection dataset. Figure 1 illustrates the location of this work's proposal within the video surveillance anomaly detection pipeline.

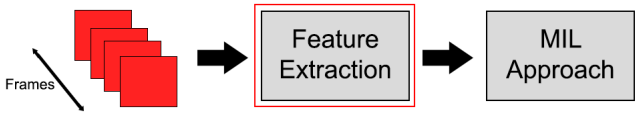


Fig. 1. The SSOC approach proposed in this paper focuses on the feature extraction step (represented by the red rectangle) in the video surveillance anomaly detection pipeline. To obtain the results, existing MIL methods are employed.

In this way, the main objective of this work is to develop a feature extraction methodology designed to leverage videos that involve interactions between objects and/or humans, aiming to capture the relationships between then in a scene. To achieve this objective, the Self-Supervised Learning approach provides an ideal framework. It involves (a) a pretext task, where a Deep Learning Network (DNN)-based model known as a pretext model is trained in an unsupervised manner, and (b) a downstream task, where the trained pretext model is fine-tuned for the desired task, such as video surveillance anomaly detection.

The code for this work is available publicly at https://github.com/daviduarte/ssoc. Furthermore, an open-source platform was developed in this study to standardize feature extraction and comparison among different MIL-based video surveillance anomaly detection methods. The source-code is also available publicly at https://github.com/daviduarte/camnuvem-research-platform. Ensuring a fair comparison has been a significant challenge, given the variations in data preparation methodologies employed by different studies [5]. Thus, the main contributions of this work are as follows:

i A feature extraction methodology based on Self-Supervised Learning for extracting features related to object interactions in videos.

ii The development of an open-source research platform that automatically extracts features from videos and compares the results among several video surveillance anomaly detection methods.

Section II will discuss the background and important works. Section III will provide details about the SSOC. Section IV will discuss the results achieved. Section V will present the discussion and conclusions.

II. Related Works

A. Convolution

Traditional convolutional blocks acquire representations from images by aggregating features using a local receptive field. Features from distant areas of image are aggregated together in the deeper layers of the networks. In videos, 3D convolutions are employed, extending the 2D receptive field to the temporal axis. Many recent works on image and video understanding, including those focused on Video Surveillance Anomaly Detection [1], [6], rely on 2D or 3D convolutions [2], [4], [7], [8]. However, most works utilize only a small temporal window with a few frames to generate features. This limitation can degrade the performance of video understanding tasks, particularly those that need more contextual and long-range information, such as video surveillance anomaly detection and robbery detection [5].

B. Transforms

Transformer models based on multi-head attention [9] have been extensively used in text understanding tasks. Subsequently, this model was successfully adapted to images in the ViT model [10], where the input is represented by 16x16 patches tokens. In the context of videos, each token represents 2D segments from frames or 3D tubelets (spatio-temporal patches) [11], [12]. In recent years, a significant number of works have employed Transformers and their variations in video understanding tasks [11]–[15]. However, a major challenge is the quadratic complexity in terms of memory and computation, which arises from the number of tokens in the input, making this approach resource-intensive. To address this problem, some works have focused on selecting the most significant tokens [16] and also combining convolutions with ViT [17].

C. Recovering Objects

To accurately predict certain types of actions, having information about other objects in the scene is beneficial. For instance, when observing someone making coffee, recognizing the coffee filter and thermos can aid in correctly classifying the action. If only the moving person is detected, there is a risk of misinterpreting the action as something else, such as making tea or cooking eggs [18]. Consequently, a branch of research focuses on explicitly detecting objects in videos using object detectors or object trackers to understand their relationships. Within this branch, some works employ graph reasoning techniques to reason about objects [19]–[24]. In these approaches, objects in the scene are detected and represented as nodes in a complete graph, and various strategies are employed to establish connections and extract relevant information.

The work of Ref. [25] examined the influence of pose estimation and spatial characteristics obtained through the tubelet of individuals in a temporal window. Similar to our work, Ref. [26] employed Self-Supervised Learning to predict features from pre-detected objects. However, in their pretext task, they propose using object class or object action labels, extracted by a pre-trained action classifier. In contrast, our work aims to predict object coordinates and appearance at a future point, and it does not involve explicit object detection during testing. Instead, it utilizes entire frames to conserve computational resources.

D. Self-Supervised Learning

The work described in Ref. [27] utilizes self-supervised learning to generate representative features for videos captured from an egocentric perspective. To perform the pretext task, they employed the Free Semantic Label approach [28], which involves simulating a 3D environment to predict the relative positions of objects from the observer. The pretext model was subsequently fine-tuned on downstream tasks, such as Natural Language Queries [29] and room prediction.

III. Methodology

The objective of the SSOC approach is to train a DNN-based model that can understanding the relationships between objects and/or humans over time using an unsupervised approach. The goal is to facilitate transfer learning to other video understanding tasks, such as video surveillance anomaly detection. The self-supervised approach is well-suited for achieving this objective because it leverages the inherent structure of the dataset to automatically generate labels that optimize a pretext DNN-based model. This enables the model to understand the dataset's intrinsic structure, thereby allowing fine-tuning in a downstream task.

Section III-A will provide a description of the pretext task, which focuses on learning the object relationships in the scene. Section III-B will discuss how the pretext model, which has been optimized in the pretext task, will be fine-tuned using video surveillance anomaly detection datasets.

A. Pretext Task

In the pretext task, a reference object r is selected within a small video segment, and its position and appearance feature vector F in a future frame are estimated based on a set of past frames, as indicated by Equation 1, where p represents the set of past frames and $\mathbf{g}$ represents a DNN-based model.

$$\mathbf{g}(p) = F \tag{1}$$

By estimating the position and appearance of an object in the future, it is expected that the model will learn how the objects influence each other and their intrinsic relationships. The term "object" here is generic and several object classes could be used, like "person", "car", "motorcycle", etc.

This approach is depicted in Figure 2. A temporal window (or scene) with $T + T'$ frames is considered, where the first T frames are used as the set of past frames p to predict the feature vector F for the reference object r in the $(T + T')$-th future frame. The reference object r is chosen as the object in the first frame of the temporal window with the highest confidence score. The method uses the Object Tracking module (gray square at the bottom of Figure 2) to provides the re-identification and tracking of object r in any of the T frames. Therefore, given the object r in the first frame and using the Object Tracking module, it is possible to access its position and appearance in the $(T + T')$-th frame (the last frame in the temporal window). This will be the label in the training process. The appearance of an object is represented by a vector $F_a \in \mathbb{R}^f$, where f is the vector size, extracted from an intermediate layer of some object detector (implemented as part of the Object Tracking module), and it is named the appearance feature vector. The position of an object is represented by a vector $X \in \mathbb{R}^4$, named the position feature vector, which describes the bounding box of the object $(x_1, y_1), (x_2, y_2)$, where the first 2D coordinate represents the top-right bounding box corner and the second 2D coordinate represents the bottom-left bounding box corner. Concatenating the appearance feature vector and the position feature vector results in the object feature vector $F \in \mathbb{R}^{f+4}$. The feature vector F of the reference object r is shown in the black rectangle in the output of the Object Tracking in Figure 2.

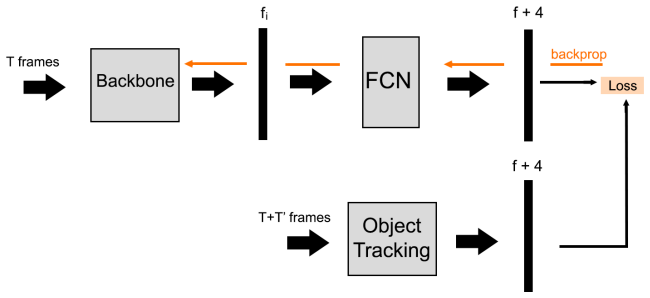


Fig. 2. The pretext task of the SSOC approach. This tasks aims to teach a DNN-based model how the objects interact in the scene.

The pretext model is composed of a deep neural network called backbone model, followed by a fully connected network (FCN) represented on top of Figure 2. The backbone receives the first T frames from the scene as input and outputs a feature vector of size f_i, corresponding to an intermediate layer that serves as a representation for the first T frames of the scene. This feature vector is then passed to the FCN, which aims to estimate the coordinates and appearance F of size $f+4$ for the reference object r in the future $(T+T')$. The real coordinates and appearance F of the reference object r in the $(T+T')$-th frame are estimated by the Object Tracker, which computes it through the re-identification of object r in all the $T + T'$ frames.

In the training process, the output of the FCN and Object Tracker is compared and the error is backpropagated through

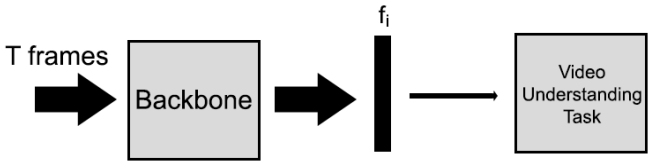


Fig. 3. Downstream task for the SSOC approach.

the FCN and backbone networks. The weights of the Object Tracker are kept frozen during this process. The quality of the training depends on the quality of the implementation of the Object Tracker and the object detector implemented in it to perform re-identification in the frames. The idea is to force the backbone model to predict the main object of the first frame of the scene (the one with the highest score in the Object Tracking's object detector) in the future. Note that, as a design consideration, the backbone model does not explicitly detect the objects in the scene to keep the computational requirements low. In this work, the Object Tracker module is implemented using the YoloV5 as object detector. More information about its implementation can be found in the official GitHub repository of this work.

B. Downstream Task

In the downstream training phase, only the backbone network from the pretext model is used. The idea here is that the backbone weights have been modified in the pretext task to understand the object relations in the scene. The FCN is excluded following the transfer learning methodology, where the last layers are removed to utilize only the features generated from the backbone. In this task, the resulting model, shown in Figure 3, is called the downstream model, and the training process will fine-tune this model.

Although this work focuses on fine-tuning in the video surveillance anomaly detection task (detailed in Section III-C), this methodology can be applied to any video dataset that presents relations between different types of objects. Therefore, Figure 3 contains a generic block labeled as "Video Understanding Task" (last gray box) to represent the downstream task.

In the training process, the error can be backpropagated through the backbone network, or new trainable layers can be inserted while keeping the backbone weights frozen. This choice depends on the experimental setup. For this work, all the details of the experimental setup are described in Section III-C.

C. Experimental Setup

As explained earlier, the video understanding task used to measure the proposed approach is video surveillance anomaly detection using Multiple Instance Learning (MIL) [1]. Among the many approaches available, the Robust Temporal Feature Magnitude learning (RTFM) [6], Weakly Supervised Anomaly Localization (WSAL) [30], and Real-World Anomaly Detection Surveillance (RADS) [1] were chosen. RTFM and WSAL

are modern techniques that are top-ranked on the UCF-Crime dataset, while RADS is a traditional technique commonly used in the evaluation of MIL works [6], [30]. Therefore, to clarify, the RTFM, WSAL, and RADS methods were used in the Video Understanding Task block shown in Figure 3.

The datasets used to generate the results presented in Section IV are the UCF-Crime dataset [1] (Section IV-A1) and the CamNuvem dataset [5] (Section IV-A2), sampled at 30 fps (original sampling rate). The parameters T and T' are set as $T = 16$ and $T' = 16$, so the objective is to teach the backbone model to predict the feature vector F of the reference object r 16 frames into the future, given the 16 past frames. These values were set empirically. We chose a small window initially to demonstrate the effectiveness of the SSOC method, with the intention of evaluating the potential impact of increasing the window size in future works.

In the pretext task, the training phase used the training partition, and the test phase used the test partition of both the CamNuvem and UCF-Crime datasets. In the downstream task, both the training and test phases also utilized the training and test partitions of the CamNuvem and UCF-Crime datasets. There was no mixing of samples from these datasets. At no point were test samples used to adjust weights in the training phase, whether in the pretext task or the downstream task. The feature vector f_i (the output of the Backbone in Figure 2) has a size of 2048, while the feature vector $f + 4$ (the output of the FCN) has a size of $1280 + 4 = 1284$. The loss function used was Mean Squared Error (MSE).

Regarding the transfer learning methodology, this work chooses to keep the weights of the backbone model frozen in the downstream task (Figure 3) and add new trainable layers on top of the downstream model. These layers are part of the RTFM, WSAL, and RADS architectures.

In the training of the downstream task, we followed the experimental methodology of RTFM, WSAL, and RADS, which means that all the final feature vectors of size $f + 4$ generated from the first T frames of the temporal windows were averaged to result in 32 segments. RTFM was trained for 1500 epochs with a learning rate of 0.001, using normalized input data. WSAL was trained for 500 epochs with a learning rate of 0.0001 for I3D and 0.001 for SSOC+I3D, as the latter did not converge with the former learning rate value. The input data was standardized for WSAL. RADS was trained for 75 epochs with a learning rate of 0.001, using normalized data. The utilized loss function was MSE.

The I3D model [2] was used as the backbone model because it has been widely used in the evaluation of many video surveillance anomaly detection works [1], [6], [30]. We extracted an intermediate layer to be the features of the set of T frames (Figures 2 and 3). The I3D weights in the pretext task were pre-trained on Kinects [2].

In the results Section (Section IV), our proposed SSOC method applied in the I3D backbone (SSOC+I3D) will be compared with the I3D network pre-trained on the Kinect dataset. Note that the models are the same, the difference is that our SSOC+I3D followed the novel pre-training procedure described in Section 2.

IV. Results

This Section presents the results of the SSOC method described in Sections III-A and III-B, using the experimental setup described in Section III-C. Section IV-A1 describes the results for the UCF-Crime dataset, and Section IV-A2 describes the results for the CamNuvem dataset.

A. General Evaluation

Tables I and II present the AUC values for the UCF-Crime dataset and CamNuvem dataset, respectively. The tables are divided into two parts: (i) All test set, which represents the AUC values calculated over all samples in the test partition, and (ii) Only abnormal videos, which represents the AUC values calculated over only the abnormal samples in the test partition. As done in Ref. [5], using method (ii) allows for a better measurement of how well a method performs at localizing the anomaly in the video. The first column represents the approach used to acquire the feature vectors from the videos, namely I3D and SSOC+I3D. The second, third, and fourth columns present the values using the RTFM, WSAL, and RADS methods, respectively. All values were calculated after 5 runs. In each cell, the highest value is depicted, followed by the mean and standard deviation in parentheses.

1) UCF-Crime dataset: Note that in Table I, considering the All test set partition, the RTFM method achieved an AUC of 0.747 using the I3D feature vector, while it achieved an AUC of 0.734 using the SSOC+I3D feature vector. This represents a worsening of 0.013 in AUC, which is a 1.3% absolute worsening. The WSAL method achieved an AUC of 0.723 using the I3D feature vector, while it achieved an AUC of 0.751 using the SSOC+I3D feature vector. This represents an improvement of 0.028 in AUC, which is a 2.8% absolute improvement. The RADS method achieved an AUC of 0.700 using the I3D feature vector, while it achieved an AUC of 0.728 using the SSOC+I3D feature vector. This represents an improvement of 0.028 in AUC, which is a 2.8% absolute improvement. Figure 4a shows the ROC curves for RTFM, WSAL, and RADS using all the samples in the UCF-Crime dataset.

In the Only abnormal videos partition of Table I, the RTFM method achieved an AUC of 0.540 using the I3D feature vector, while it achieved an AUC of 0.528 using the SSOC+I3D feature vector. This represents a worsening of 0.012 in AUC, which is a 1.2% absolute worsening. The WSAL method achieved an AUC of 0.630 using the I3D feature vector, while it achieved an AUC of 0.633 using the SSOC+I3D feature vector. This represents an improvement of 0.003 in AUC, which is a 0.3% absolute improvement. The RADS method achieved an AUC of 0.479 using the I3D feature vector, while it achieved an AUC of 0.549 using the SSOC+I3D feature vector. This represents an improvement of 0.070 in AUC, which is a 7% absolute improvement. Figure 4b

shows the ROC curves for RTFM, WSAL, and RADS using only the abnormal samples in the UCF-Crime dataset.

	RTFM	WSAL	RADS
All test set			
I3D	**0.747** **(0.745 ± 0.002)**	0.723 (0.700 ± 0.016)	0.700 (0.698 ± 0.001)
SSOC+I3D (ours)	0.734 (0.723 ± 0.011)	**0.751** **(0.739 ± 0.008)**	**0.728** **(0.721 ± 0.006)**
Only abnormal videos			
I3D	**0.540** **(0.531 ± 0.007)**	0.630 (0.616 ± 0.012)	0.479 (0.477 ± 0.001)
SSOC+I3D (ours)	0.528 (0.511 ± 0.012)	**0.633** **(0.622 ± 0.008)**	**0.549** **(0.539 ± 0.006)**

TABLE I
GREATER VALUES OF AUC FROM 5 RUNS USING UCF-CRIME DATASET.

	RTFM	WSAL	RADS
All test set			
I3D	0.858 (0.848 ± 0.006)	0.807 (0.798 ± 0.006)	**0.791** **(0.788 ± 0.002)**
SSOC+I3D (ours)	**0.884** **(0.878 ± 0.003)**	**0.838** **(0.837 ± 0.001)**	0.776 (0.772 ± 0.002)
Only abnormal videos			
I3D	0.560 (0.522 ± 0.021)	**0.628** **(0.598 ± 0.018)**	**0.519** **(0.518 ± 0.001)**
SSOC+I3D (ours)	**0.616** **(0.601 ± 0.014)**	0.573 (0.562 ± 0.009)	0.467 (0.465 ± 0.001)

TABLE II
GREATER VALUES OF AUC FROM 5 RUNS USING CAMNUVEM DATASET.

2) CamNuvem dataset: Note that in Table II, considering the All test set partition, the RTFM method achieved an AUC of 0.858 using the I3D feature vector, while it achieved an AUC of 0.884 using the SSOC+I3D feature vector. This represents an improvement of 0.026 in AUC, which is a 2.6% absolute improvement. The WSAL method achieved an AUC of 0.807 using the I3D feature vector, while it achieved an AUC of 0.838 using the SSOC+I3D feature vector. This represents an improvement of 0.031 in AUC, which is a 3.1% absolute improvement. On the other hand, the RADS method achieved an AUC of 0.791 using the I3D feature vector, while it achieved an AUC of 0.776 using the SSOC+I3D feature vector. This represents a worsening of 0.015 in AUC, which is a 1.5% absolute worsening. Figure 5a shows the ROC curves for RTFM, WSAL, and RADS using all the samples in the test set.

In the Only abnormal videos partition of Table II, the RTFM method achieved an AUC of 0.560 using the I3D feature vector, while it achieved an AUC of 0.616 using the SSOC+I3D feature vector. This represents an improvement of 0.056 in AUC, which is a 5.6% absolute improvement. The WSAL method achieved an AUC of 0.628 using the I3D feature vector, while it achieved an AUC of 0.573 using the SSOC+I3D feature vector. This represents a worsening of 0.055 in AUC, which is a 5.5% absolute worsening. The RADS method achieved an AUC of 0.519 using the I3D feature vector, while it achieved an AUC of 0.467 using the SSOC+I3D feature vector. This represents a worsening of 0.052 in AUC, which is a 5.2% absolute worsening. Figure 5b shows the ROC curves for RTFM, WSAL, and RADS using only the abnormal samples in the test set.

V. DISCUSSION AND CONCLUSIONS

We proposed the SSOC approach to generate high-quality feature vectors from videos, aiming to extract relationships between objects and/or humans. We evaluated our approach alongside the traditional I3D feature extractor and found that, out of the 12 cases evaluated, 7 of them (58%) achieved better results. In particular, the WSAL and RTFM methods, applied to the UCF-Crime and CamNuvem datasets respectively, demonstrated superiority over traditional approaches in all evaluated metrics.

A significant limitation of this work is the label generation process, where the success of the training relies on the quality of the Object Tracking module. Therefore, in future works, the evaluation of the Object Tracking module using well-defined metrics will be explored, as well as the comparison of SSOC with alternative feature extraction approaches (beyond just I3D) to establish the validity of SSOC.

ACKNOWLEDGMENT

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES)

REFERENCES

[1] W. Sultani, C. Chen, and M. Shah, "Real-world anomaly detection in surveillance videos," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 6479–6488.

[2] J. Carreira and A. Zisserman, "Quo vadis, action recognition," *A new model and the kinetics dataset. CoRR, abs/1705.07750*, vol. 2, no. 3, p. 1, 2017.

[3] C. Feichtenhofer, H. Fan, J. Malik, and K. He, "Slowfast networks for video recognition," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2019, pp. 6202–6211.

[4] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "Learning spatiotemporal features with 3d convolutional networks," in *IEEE International Conference on Computer Vision*, 2015, pp. 4489–4497.

[5] D. D. de Paula, D. H. Salvadeo, and D. M. de Araujo, "Camnuvem: a robbery dataset for video anomaly detection," *Sensors*, vol. 22, no. 24, p. 10016, 2022.

[6] Y. Tian, G. Pang, Y. Chen, R. Singh, J. W. Verjans, and G. Carneiro, "Weakly-supervised video anomaly detection with robust temporal feature magnitude learning," in *IEEE/CVF International Conference on Computer Vision*, 2021, pp. 4975–4986.

[7] C. Feichtenhofer, "X3d: Expanding architectures for efficient video recognition," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 203–213.

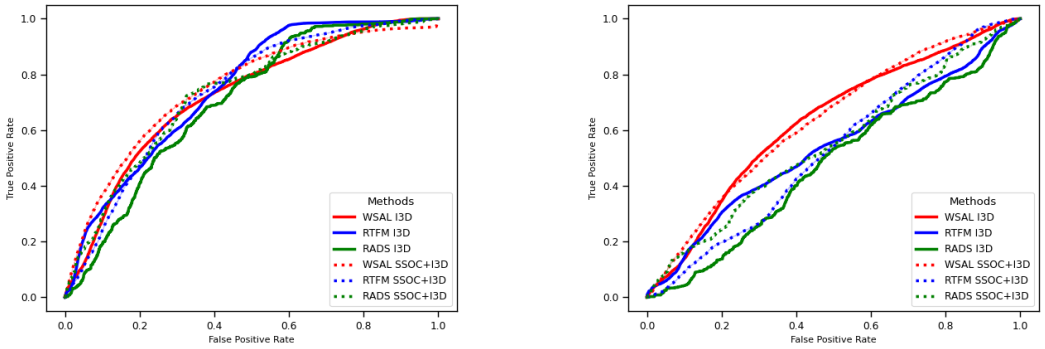
[8] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 7794–7803.

[9] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[10] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly *et al.*, "An image is worth 16x16 words: Transformers for image recognition at scale," *arXiv preprint arXiv:2010.11929*, 2020.

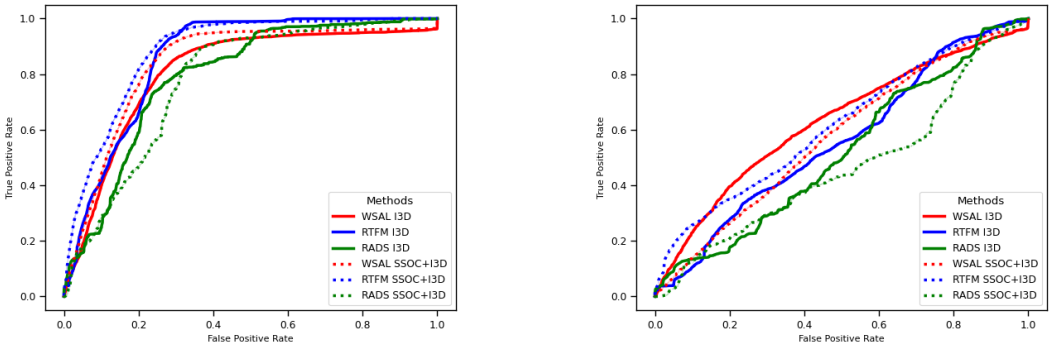
[11] A. Arnab, M. Dehghani, G. Heigold, C. Sun, M. Lučić, and C. Schmid, "Vivit: A video vision transformer," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 6836–6846.

[12] G. Bertasius, H. Wang, and L. Torresani, "Is space-time attention all you need for video understanding?" 2021.



(a) All test set (b) Only abnormal videos

Fig. 4. Roc curve for I3D and I3D+SSOHC using the methods WSAL, RTFM, and RADS.



(a) All test set (b) Only abnormal videos

Fig. 5. Roc curve for I3D and I3D+SSOHC using the methods WSAL, RTFM and RADS.

[13] L. Xu, Y. Guan, S. Jin, W. Liu, C. Qian, P. Luo, W. Ouyang, and X. Wang, "Vipnas: Efficient video pose estimation via neural architecture search," 2021.

[14] R. Girdhar, J. Carreira, C. Doersch, and A. Zisserman, "Video action transformer network," 2019.

[15] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," 2020.

[16] M. S. Ryoo, A. Piergiovanni, A. Arnab, M. Dehghani, and A. Angelova, "Tokenlearner: What can 8 learned tokens do for images and videos?" *arXiv preprint arXiv:2106.11297*, 2021.

[17] B. Graham, A. El-Nouby, H. Touvron, P. Stock, A. Joulin, H. Jégou, and M. Douze, "Levit: a vision transformer in convnet's clothing for faster inference," in *Proceedings of the IEEE/CVF international conference on computer vision*, 2021, pp. 12 259–12 269.

[18] J. Wang, W. Zhu, P. Wang, X. Yu, L. Liu, M. Omar, and R. Hamid, "Selective structured state-spaces for long-form video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 6387–6397.

[19] D. Li, Z. Qiu, Y. Pan, T. Yao, H. Li, and T. Mei, "Representing videos as discriminative sub-graphs for action recognition," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 3310–3319.

[20] B. Wu, G. Niu, J. Yu, X. Xiao, J. Zhang, and H. Wu, "Towards knowledge-aware video captioning via transitive visual relationship detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 32, no. 10, pp. 6753–6765, 2022.

[21] X. Wang and A. Gupta, "Videos as space-time region graphs," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 399–417.

[22] M. Tomei, L. Baraldi, S. Calderara, S. Bronzin, and R. Cucchiara, "Video action detection by learning graph-based spatio-temporal interactions," *Computer Vision and Image Understanding*, vol. 206, p. 103187, 2021.

[23] R. Li, X.-J. Wu, and T. Xu, "Video is graph: Structured graph module for video action recognition," *arXiv preprint arXiv:2110.05904*, 2021.

[24] N. Li, J.-X. Zhong, X. Shu, and H. Guo, "Weakly-supervised anomaly detection in video surveillance via graph convolutional label noise cleaning," *Neurocomputing*, vol. 481, pp. 154–167, 2022.

[25] J. Rajasegaran, G. Pavlakos, A. Kanazawa, C. Feichtenhofer, and J. Malik, "On the benefits of 3d pose and tracking for human action recognition," 2023.

[26] C.-Y. Wu and P. Krahenbuhl, "Towards long-form video understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1884–1894.

[27] T. Nagarajan, S. K. Ramakrishnan, R. Desai, J. Hillis, and K. Grauman, "Egoenv: Human-centric environment representations from egocentric video," *arXiv preprint arXiv:2207.11365*, 2022.

[28] L. Jing and Y. Tian, "Self-supervised visual feature learning with deep neural networks: A survey," *IEEE transactions on pattern analysis and machine intelligence*, 2020.

[29] K. Grauman, M. Wray, A. Fragomeni, J. P. Munro, W. Price, P. Arbelaez, D. Crandall, D. Damen, G. M. Farinella, B. Ghanem *et al.*, "Around the world in 3,000 hours of egocentric video," in *Computer Vision and Pattern Recognition (CVPR)*, 2022.

[30] H. Lv, C. Zhou, Z. Cui, C. Xu, Y. Li, and J. Yang, "Localizing anomalies from weakly-labeled videos," *IEEE Transactions on Image Processing*, vol. 30, pp. 4505–4515, 2021.