A Graph-based Approach to Video Anomaly Detection from the Perspective of Superpixels

Siemon, Mia Sandra Nicole; Nasrollahi, Kamal; Moeslund, Thomas B.

Published in:
Fifteenth International Conference on Machine Vision, ICMV 2022

DOI (link to publication from Publisher):
10.1117/12.2679673

Publication date:
2023

Document Version
Accepted author manuscript, peer reviewed version

A Graph-based Approach to Video Anomaly Detection from the Perspective of Superpixels

Mia Siemon[1,2], Kamal Nasrollahi[1,2], Thomas B. Moeslund[1]
[1]Aalborg University, [2]Milestone Systems A/S

ABSTRACT

Video Anomaly Detection refers to the concept of discovering activities in a video feed that deviate from the usual visible pattern. It is a very well-studied and explored field in the domain of Computer Vision and Deep Learning, in which automated learning-based systems are capable of detecting certain kinds of anomalies at an accuracy greater than 90%. Deep Learning based Artificial Neural Network models, however, suffer from very low interpretability. In order to address and design a possible solution for this issue, this work proposes to shape the given problem by means of graphical models. Given the high flexibility of compositing easily interpretable graphs, a great variety of techniques exist to build a model representing spatial as well as temporal relationships occurring in the given video sequence. The experiments conducted on common anomaly detection benchmark datasets show that significant performance gains can be achieved through simple re-modelling of individual graph components. In contrast to other video anomaly detection approaches, the one presented in this work focuses primarily on the exploration of the possibility to shift the way we currently look at and process videos when trying to detect anomalous events.

Keywords: Computer Vision, Video Anomaly Detection, Spatio-Temporal Graphs, Graph Convolutional Networks.

1. INTRODUCTION

Approaching the problem of conducting anomaly detection in videos by means of Deep Learning (DL) is increasingly turning into an Artificial Neural Network (ANN) optimization task. Rather than introducing new solution perspectives, possibly from different domains, many solution proposals are constructed by means of many different modules which operate in parallel [1,2]. And even though these approaches are scoring state-of-the-art results on individual benchmark datasets, video anomaly detection still remains far from representing the same sophistication as other prevalent image processing tasks such as image classification and image segmentation. One of the main constraints for this is the limited amount of publicly available anomaly detection datasets, and the diversity they represent. In single-scene video anomaly detection, there exist 7 datasets [3] out of which the shortest one consists of 3.855, and the largest one out of 203.257 frames in total. While some of them are frequently used by researchers for training and evaluation, others started becoming less relevant for real-world application cases because of the either limited variety of anomalies they show, or the fact that some of them are posed. Another problem with current video anomaly detection approaches is that they do not generalize well in terms of performance across the above-mentioned datasets. StreetScene [3] in particular, poses a challenge for so-called object-centric anomaly detection suggestions which heavily rely on the preprocessing of the training frames by means of an object detector. Additionally, anomalies are heavily context-dependent, which needs to be given particular importance if the goal is to expand single-scene anomaly detection processes into multi-scene ones.

The main motivation for this work is to design and implement a video anomaly detection system which is characterized by the following: **Considerable performance improvement gains** when comparing the proposed grid-based graph creation approach to a purely object-centric baseline; **High model interpretability** thanks to the graph creation process which is designed from bottom up to encode spatio-temporal context extracted from video sequences. Additionally, an ablation study of possible edge modelling approaches is conducted.

2. RELATED WORK

The purpose of this section is to present an overview of existing attempts to bridge the gap between video anomaly detection and the concept of graphical models. After giving a short introduction about the most recent trends in the field of video anomaly detection, graphs in general will be covered along with an outline of existing methods for incorporating graph-based problem-solving techniques in the domain of video anomaly detection.

2.1 Video Anomaly Detection

Detecting anomalies in video (surveillance) footage has turned into one of the most relevant topics in Computer Vision (CV) over recent years, in addition to image classification and image segmentation [4,5]. Its main disadvantage compared to other tasks, however, is that anomalies are nearly impossible to define upfront. Therefore, unsupervised learning techniques have been mostly applied in combination with reconstruction-based DL methods [5,1,6,7] to train the underlying ANN how it should define what is normal. ANNs falling into this category are also commonly known as Auto-Encoders (AEs) [6] and Generative Adversarial Networks (GANs) [7]. Whilst these reconstruction approaches primarily target the entire frame, a shift towards object-centric anomaly detection approaches is seen in [1,8,2], as well as one towards self-supervised learning [1,9]. The most recent state-of-the-art DL-based anomaly detection methods manage to achieve detection accuracies of up to 92.9% [9] on most popular anomaly detection benchmark datasets [10]. There exists, however, no ANN capable of generalizing across different benchmark dataset with similar accuracies. A recently published dataset depicting a street surveillance setup, called Street Scene [3], in particular, challenges most anomaly detection systems [2] in contrast to others, like CUHK Avenue [10] and ShanghaiTech [11].

2.2 Graphs

Context-aware applications have been receiving more and more attention throughout the years thanks to their increasingly-improving capabilities of performing their tasks, such as conducting "intelligent" dialogue, and recommending products, for example. This context, which can be also referred to as knowledge, is most commonly modelled in terms of graphs. Whilst Knowledge Graphs [12] were designed to represent factual knowledge, Scene Graphs [13], on the other hand, were introduced shortly after to reduce the underlying complexity by only describing visual knowledge that can be derived from images. They have been primarily used to solve more semantically challenging tasks such as image captioning [14], semantic image retrieval [15], and visual questioning answering [16]. They do find, however, also application in action recognition tasks [17,18] when encoded in spatio-temporal graph structures.

2.3 Graphs in Video Anomaly Detection

To date, targeting the problem of video anomaly detection using graph-based structures still remains uncommon. The most prevalent practice of computationally describing the human pose is to use graph-like structures. In such a way, human joints can be easily modelled by means of nodes, and bones in terms of edges. Past works have already shown that is possible to encode temporal information of human actions translated to graphs [17,18]. Therefore, Markovitz et al. [19] went one step further and implemented a deep spatio-temporal graph AE allowing them to cluster graphical human pose estimations, and further detect abnormal outliers yielding a frame-level accuracy of 0.752 during tests on the ShanghaiTech dataset. Nesen and Bhargava [20] on the other hand, assign anomaly scores to objects in a frame by means of a two-step process: At first, all objects are detected in the given image by means of an off-the-shelf object detector [21]. Secondly, semantic correlation scores for all object-pairs in the frame are derived from ConceptNet [22]. Last but not least, Pourreza et al. [23] approached the problem by means of a purely graph-based manner. After detecting all objects across the frames, the authors of [23] model spatial relationships between objects within a single frame by expressing possible intersections as Intersection over Unions. Temporal relationships on the other hand, are designed between objects across two consecutive frames using the cosine similarity function. Populating the adjacency matrix for the graph with this information allows Pourreza et al. to train a Graph Convolutional Network (GCN) to perform the anomaly detection task. More details about this particular approach will follow throughout the rest of this paper.

3. GRAPH-BASED ANOMALY DETECTION

This section will serve as detailed description of the proposed methodology. At first, an insight into the manual graph creation process and different modelling schemes to be analyzed will be given. The remaining part of the section will focus on the automated part of the pipeline which considers the actual learning and inference in a DL-based environment.

3.1 Grid-inspired Graph Creation

As opposed to the graph construction approach suggested by Pourreza et al. [23] the method of this work does not model individual objects as nodes and connecting edges as spatial and temporal relations between them. Instead, it imposes a uniform grid structure onto the image treating each cell in the grid as a node in the graph. Each cell/node can be further viewed as a superpixel representative for a particular set of pixels falling into this cell. Such a perspective attributes a more rigid structure to the graph which is visualized in Figure 1(a-b). As it can be seen, all frames in the video sequence comprise

of the same number of nodes. Spatial relations between those nodes are drawn by means of bi-directional edges that connect the current node to all its neighbors of degree 1. Temporal information is further encoded in unidirectional edges between two nodes sharing the same grid position across two frames that are coherent with the direction of the chronological order of video frames. Details with respect to the values which are assigned to nodes and edges, will follow in the next paragraph.

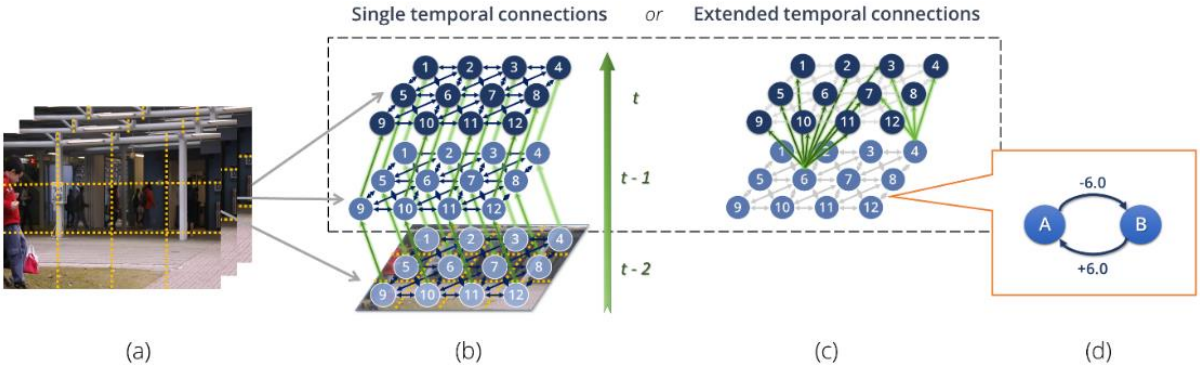

Figure 1. Illustration of the graph creation process with (a) showing how a frame is divided into uniform grid cells, (b) visualizing a spatio-temporal graph constructed with three consecutive frames, (c) depicting a possible extension of temporal connections, and (d) giving an example for calculating the edge weights between two nodes, A and B, of values 10 and 4, respectively.

3.1.1 Modelling Approaches

As has already been described in the previous paragraph, when growing the graph, grid cells are treated as nodes. For this reason, the terms *node* and *cell* are used interchangeably. Spatio-temporal information, on the other hand, is encoded in the graph edges. The aim is to analyze the impact different node and edge modelling techniques can have on the overall performance of the underlying anomaly detection system. Experiments focus primarily on different edge creation techniques in order to explore the importance of enhanced spatial and/or temporal connections between nodes.

Nodes Overall, the goal is to find the most expressive way of putting bounding boxes into relation with respective grid cells they cover. To do so, the foundation for all following edge modelling proposals will be set by firstly determining what the node values should represent. In this work, the choice was to assign the cumulative sum of Intersection over Unions (IoU) of all bounding boxes which overlap with a grid cell to its graph node, whilst all remaining nodes whose cells are not covered by any bounding box hold the value 0.

Edges When it comes to spatial connections between nodes that represent a single frame, two approaches of defining the corresponding edge values have been designed. In the first one, this value is equivalent to the numeric difference of the values held by the two nodes the edge is connecting. The sign of the value of the edge is given by its direction, and can also be interpreted as a cost required to get from one node to the other. Figure 1(d) illustrates this with an example. As it can also be seen in Figure 1(b), all spatial edges are bidirectional and connect a single node to all of its closest neighbors of degree 1. This approach will be referred to as Method 1.

In the second proposal to model spatial information between cells forming a single frame, instead of considering the numeric difference between two node values, the idea is to encode absolute information about whether a cell has overlapping bounding boxes or not. This is achieved by assigning the value 1 to edges connecting two nodes that hold values larger than 0. Otherwise, it is set to -1, in order to underline the presence/absence of objects in the grid.

In the case of temporal connections, the most crucial difference compared to spatial ones is that they are unidirectional. This choice was made to underline the flow of time and its direction. By means of these last two modelling approaches the importance of the amount of temporal connections between two consecutive frames is to be assessed. To do so, only single directional edges are established between two nodes sharing the same position across two frames at first. Along with the first modelling approach for spatial connections, this method is referred to as Method 2. Afterwards, the number of outgoing temporal edges from a single node are increased in a way that allows connections to all neighboring cells of degree 1 to the node located at the same grid position. This is shown in Figure 1(c), which, for visual simplicity reasons, only uses two nodes, 6 and 4, to illustrate this modelling approach which is referred to as Method 3 along with the second modelling proposal for spatial edges. From a high-level perspective, this approach is very similar to a weighted convolutional layer with a filter of size (3×3). The edge values are determined by the numeric difference between the values held by the connected nodes.

This work aims to evaluate both the modelling approach of the graph itself, as well as the influence that the grid cell dimensionality may have on the anomaly detection performance. Depending on the image dimensions of the given dataset, experiments are performed with cell sizes 20×20, 40×40 and 80×80.

3.1.2 Object Detection

In order to detect all objects in the video and extract their respective feature representations, a pretrained object detector was deployed. The training was performed on the MS-COCO [24] dataset which represents a total of 80 different classes. YOLOv3 [25] is the third version of the object detector *You Only Look Once (YOLO)* [26] which was introduced in 2016 as a detector which is capable of generating very general object representations whilst producing less false positives during real-time inference when compared with other previous state-of-the-art networks. YOLOv3 uses Darknet-53 to extract object features to make predictions at three different scales. This allows the network to attribute equal importance to objects which are very far away from the camera, those that are very close to it, and those that are in between. Instead of focusing on the output of a single layer in the network, feature maps from all three YOLO layers are obtained, compressed, concatenated, and extended with binary-encoded class and confidence score. Compression from 3D to 1D takes place by taking the mean of all layers across the depth of the feature map, respectively, so that

$$\boldsymbol{f_b} = [\, f_1, f_2, f_3, \dots, f_{D-2}, f_{D-1}, f_D \,] \;, \tag{1}$$

$$f_d = \sum_{h=0}^{H} \sum_{w=0}^{W} l_{h,w}^{d} \;, \tag{2}$$

where $\boldsymbol{f_b}$ denotes the compressed one-dimensional feature vector for bounding box b with D elements. f_d is the vector value at depth d, equivalent to the sum of all elements $l_{h,w}$ in layer l^d with height h and width w such that $h = w$.

In YOLOv3, features extracted at the first YOLO layer, i.e., layer 81, are of shape $(19 \times 19 \times 1.024)$ whilst those at layers 93 and 105 are $(38 \times 38 \times 512)$ and $(76 \times 76 \times 256)$, respectively. After compression, concatenation and expansion, together, they form a unique object feature vector of length $(8 + 1 + 1.024 + 512 + 256 = 1.801)$. The first eight digits in the final feature vector encode the detected class of the object in binary format. One additional digit is required to keep the confidence score the network assigned to the given prediction. All remaining digits are then populated with compressed and concatenated features extracted from the network. By supplementing the compressed and concatenated feature vector with explicit information about confidence score of the predicted class and the class itself, the goal is to increase the difference between classes sharing very akin features.

3.2 Learning: Deep Graph Infomax

Deep Graph Infomax (DGI) [27] belongs to the category of GCNs. Compared to Graph Neural Networks (GNN) [28], GCNs perform the learning task based on a static adjacency matrix of the underlying graph which is not updated throughout the training process. DGI was introduced as a mean to conduct different node classification tasks in graph-structured data in an unsupervised training setting. Its architectural design allows it to generate so-called node embeddings which are equivalent to individual node signals that contain global information about all remaining nodes in the graph. After learning these signals, they can be grouped into clusters representative for the different classes in the underlying classification problem. Initially, node signals are equivalent to object feature representations which have been described in the previous paragraph on *Object Detection* and feature extraction. Given the graph that was created according to Subsection 3.1, its adjacency matrix is of dimensions $(N \times N)$, where N is equal to number of cells per frame, times the total amount of frames in the dataset. The object feature representations require therefore a shape of $(N \times 1.801)$, so that every feature vector maps to its respective grid cell. Each of these vectors is equivalent to the cumulative sum of extracted object features dependent on the relative intersection of the corresponding bounding box and the given cell:

$$\boldsymbol{f_c} = \sum_{b=1}^{B} \frac{IoU_{b,c}}{{c_w}^2} \cdot \boldsymbol{f_b} \;, \tag{3}$$

where $\boldsymbol{f_c}$ equals to the object feature representation at node c, B is the number of bounding boxes which overlap with cell c, c_w denotes the width of a grid cell (equal to its height), and $\boldsymbol{f_b}$ is taken from Equation 1.

Overall, the training procedure of the DGI is unsupervised. In the domain of Anomaly Detection, this implies that the network is only trained on data which is considered to be representative for normal activities. Any abnormal behavior which is then fed into the network upon testing should therefore strongly deviate from the learned distribution. Even though

DGI does not require to be confronted with real abnormal data when trained, it does, however, rely on *corrupted* data which needs to be supplied along with the normal training data. This corrupted data essentially serves as artificially induced anomalies which are created by means of row-wise shuffling of the object feature representations [27].

3.2.1 Inference: Anomalous Cell Detection

Inference of anomalies is conducted by identifying the most anomalous cell in a frame. The operation of the DGI network involves firstly, the creation of test node embeddings utilizing the pretrained model weights, and secondly, their comparison with the global summary vector which was extracted at the end of training. In order to create a graph with meaningful temporal information, at least three consecutive frames are required for inference.

3.2.2 Training

The training of the DGI network described earlier in this subsection is performed in 3 steps:

1) **Object Detection**: All objects in the training set are detected, their bounding box, class, confidence score and features extracted, out of which the last three are merged into $\boldsymbol{f_b}$
2) **Graph Creation**: The training graph is grown from scratch, modelled as a bidirectional graph with nodes representing grid cells across all frames. Spatio-temporal information is encoded in the edges connecting nodes within a single frame, and those across two consecutive frames.
3) **Learning**: The DGI network is supplied with the adjacency matrix of the training graph as well as with the node-wise object feature representations in order to learn the underlying node signals. It outputs a summary vector encapsulating global information of normal activities occurring in the training video

In practice, the resulting size of an adjacency matrix described in subsection 3.2 is very difficult to process computationally, given large amounts of training images. Therefore, after concatenating all training frames into one sequence, the image data is processed in batches. The size of these batches further depends on the chosen grid cell sizes as shown in Table 1.

Three most common anomaly detection datasets have been chosen for training and evaluation of this grid-based anomaly detection approach: CUHK Avenue, ShanghaiTech and StreetScene. It is important to note that only a subset of videos was chosen from ShanghaiTech for the purpose of this experiment. This is due to the large camera movement which is present in videos 01 and 04 which therefore were left out.

Image Size	**CUHK Avenue**	(640×360)		**ShanghaiTech**	$(1.280 \times 720)^*$		**StreetScene**	(1.280×720)	
Cell Size		20	40		40	80		40	80
Nodes/frame		576	144		576	144		576	144
Batch Size		7	30		7	30		7	30

Table 1. Overview of graph creation and training parameters. The image size denoted by the asterisk (*) does not represent the original dimensions given in [9]. It was adapted to enable a computationally feasible cell size choice. The DGI network uses ADAM as optimizer module at learning rate 10^{-4} and weight decay 0.0.

3.2.3 Evaluation

Similar to the training procedure explained in the preceding paragraph, all test frames are concatenated in chronological order, at first. Afterwards, a test graph is grown, its node embeddings extracted and then correlated with the summary vector obtained after training, by means of a bilinear transformation. The resulting logits are then compared against the ground truth values. All images are processed in batches according to the specifications given in Table 1. The overall performance of the network is assessed by means of the frame-level ROC AUC metric.

3.2.4 Technical Details

The implementation of this work subsists in two major modules: Object and Anomaly Detection. The former was executed by means of a partially modified implementation of YOLOv3 in C[1] as described in 3.1.2. The latter on the other hand, is purely based on Python 3 and PyTorch. All graph computations and analyses were performed utilizing the

[1] https://pjreddie.com/darknet/yolo/

networkx[2] package. In terms of hardware, all experiments were executed on an NVIDIA GPU, model GeForce RTX 3080 Ti, with 12GB of memory, running CUDA 11.5.

4. EXPERIMENTAL RESULTS

In order to have a baseline to compare the obtained results against, two versions of Ano-Graph, as given by Pourreza et al. in [23], are implemented: one with Faster RCNN (original version of Ano-Graph), and the other with YOLOv3 (modified version of Ano-Graph) as object detector. The primary motivation for choosing this work as the baseline is two-fold: Firstly, the choice of a graphical model to perform video anomaly detection, and secondly because the effect of defining spatial information in terms of a grid-dependent perception of locality is to be analyzed. Results recorded in Table 2 show, that it was unfortunately not possible to reproduce the results as they were reported in [23]. A shift of focus from modelling the anomaly detection problem as an analysis of inter-object to cell-object relations however, shows the superiority of our approach versus the reproduced baselines.

Cell size / **Method**		**20**	**40**		**40**	**80**		**40**	**80**
(1)	**CUHK Avenue**	0,5898	**0,6512**	**Shanghai Tech**	0,5565	0,5634	**Street Scene**	0,5355	**0,5406**
(2)		0,6186	0,5971		0,5272	0,5566		0,5099	0,5104
(3)		0,6034	0,6392		0,5485	**0,5716**		0,5188	0,5274
Ano-Graph, orig.		*0,5496*			*0,5652*			*0,5337*	
Ano-Graph, mod.		*0,5241*			*0,5375*			*0,5092*	

Table 2. Frame-level AUC scores for presented graph-based anomaly detection proposals: (1) Single temporal links, (2) Extended temporal links, (3) Absolute spatial links and extended temporal ones. The original version of Ano-Graph uses Faster RCNN whilst the modified version employs YOLOv3. Numbers in bold font represent the highest performance score achieved per dataset.

A slight drop in frame-level AUC scores can be always observed between the original and the modified Ano-Graph implementations. The main reason for this is most likely to be that the performance of YOLOv3 becomes inferior to Faster RCNN when detecting objects which are very far away from the camera [29]. This becomes very apparent in the StreetScene dataset which is recording activities taking place on a street from a tall building. This has an impact on the original Ano-Graph implementation being more accurate than the modified version, as well as on the low performance of the grid-based approach proposed in this work. The strongest performance of the grid-based approach is very visible when trained and evaluated on the CUHK Avenue dataset compared to the modified version of Ano-Graph: The performance growth equals to around 13%. In contrast to initial expectations, best results are coherently observed for the greater cell size. Even though the smaller value adds a finer grid granularity, the lower AUC scores indicate that this particular graph model is too sensitive for connections between nodes that could become too cluttered. A comparison of all three edge modelling proposal leads to the conclusion that the extended temporal receptive field only leads to more accurate predictions when combined with absolute spatial connections. One possible explanation for this can be that the GCN becomes more responsive to temporal connections given the difference in variances between spatial and temporal links. Example predictions made by the best performing methods (Table 2) per dataset are shown in Figure 2.

Figure 2. Example anomaly detections on CUHK Avenue (left: backpack worn by girl walking in an unusual direction), ShanghaiTech (middle: two students walking closely, one on a bike), and StreetScene (right: ladder on car roof). Anomaly scores are given in red font in the upper left corner. On a scale of 0 to 1 the convention is that 0 denotes an anomalous and 1 a normal cell, respectively.

[2] https://networkx.org/

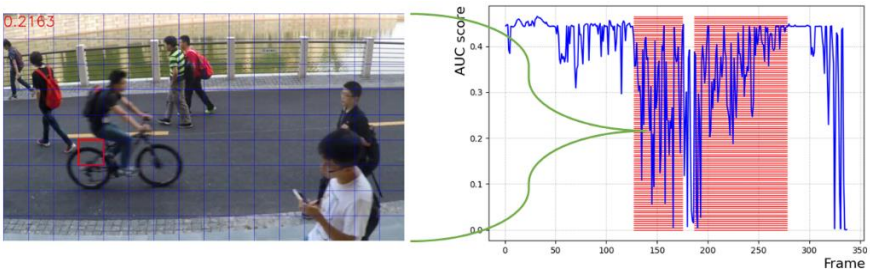

Figure 3. Anomaly scores across frames of test video 11 from ShanghaiTech (right) computed by the best model/method given in Table 2, and a corresponding (correct) detection of an abnormal behavior in the scene, i.e., a person riding a bike (left).

5. CONCLUSION AND FUTURE WORK

This paper presents an approach to design and grow a bidirectional spatio-temporal graph based on superpixels which constitute patch-wise summaries of local neighborhoods in the frame. In contrast to the chosen baseline model, the contribution consists of a graph modelling proposal which encodes the location of object bounding boxes and interactions thereof in a rigid scene structure. The fundamental design of the graph-based learning model leads to an increased interpretability of the modelling process. Contrary to other ANN and DL models which are mapping images to low-level representations in latent space, this graphical model broadens the opportunity to enhance human understanding of this CV domain. On the other side, several drawbacks of graph-based image processing remain unsolved. These include exponentially increasing computational complexity, and a highly sensitive modelling process. In this work, especially the latter seems very apparent when looking at the highest frame-level AUC score achieved. 65% is relatively far from latest state-of-the-art achieved in the domain of video anomaly detection (92.9% [9]). Nevertheless, the analysis performed on different graph modelling techniques in this work clearly shows the potential and opportunity for more adequate graph compositions. Future work opportunities may thus involve the design of a probabilistic graphical model approach such as Markovian and/or Bayesian modelling techniques. This would erase the need for deploying a DL network (GCN) which heavily decreases the interpretability of the problem-solving approach. If the grid-based perspective of images is to be kept, one additional step to extend this work would be to refine the prediction outcome from cell- to object-based.

REFERENCES

[1] M.-I. Georgescu et al., “Anomaly Detection in Video via Self-Supervised and Multi-Task Learning”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 12737–12747, IEEE, Nashville, TN, USA (2021) [doi:10.1109/CVPR46437.2021.01255]

[2] M. I. Georgescu et al., “A Background-Agnostic Framework with Adversarial Training for Abnormal Event Detection in Video”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–1 (2021) [doi:10.1109/TPAMI.2021.3074805]

[3] B. Ramachandra and M. J. Jones, “Street Scene: A new dataset and evaluation protocol for video anomaly detection”, IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 2558–2567, IEEE, Snowmass Village, CO, USA (2020) [doi:10.1109/WACV45572.2020.9093457]

[4] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly detection: A survey”, ACM Comput. Surv. 41(3), 1–58 (2009) [doi:10.1145/1541880.1541882].

[5] B. Ramachandra, M. J. Jones, and R. R. Vatsavai, “A Survey of Single-Scene Video Anomaly Detection”, IEEE Transactions on Pattern Analysis and Machine Intelligence (2020)

[6] M. S. Minhas and J. Zelek, “Semi-supervised Anomaly Detection using AutoEncoders”, 1, Journal of Computational Vision and Imaging Systems 5(1), 3–3 (2019)

[7] I. J. Goodfellow et al., “Generative Adversarial Networks”, arXiv:1406.2661, arXiv (2014) [doi:10.48550/arXiv.1406.2661]

[8] R. T. Ionescu et al., “Object-Centric Auto-Encoders and Dummy Anomalies for Abnormal Event Detection in Video”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7834–7843, IEEE, Long Beach, CA, USA (2019) [doi:10.1109/CVPR.2019.00803]

[9] N.-C. Ristea et al., “Self-Supervised Predictive Convolutional Attentive Block for Anomaly Detection”, arXiv:2111.09099, arXiv (2022) [doi:10.48550/arXiv.2111.09099]

[10] C. Lu, J. Shi, and J. Jia, “Abnormal Event Detection at 150 FPS in MATLAB”, IEEE Intl. Conference on Computer Vision, pp. 2720–2727, IEEE, Sydney, Australia (2013) [doi:10.1109/ICCV.2013.338]

[11] W. Liu et al., “Future Frame Prediction for Anomaly Detection - A New Baseline”, IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 6536–6545, IEEE, Salt Lake City, UT (2018) [doi:10.1109/CVPR.2018.00684]

[12] S. Ji et al., “A Survey on Knowledge Graphs: Representation, Acquisition, and Applications”, IEEE Transactions on Neural Networks and Learning Systems, 1–21 (2021) [doi:10.1109/TNNLS.2021.3070843]

[13] X. Chang et al., “A Comprehensive Survey of Scene Graphs: Generation and Application”, IEEE Transactions on Pattern Analysis and Machine Intelligence, 1–1 (2021) [doi:10.1109/TPAMI.2021.3137605]

[14] X. Yang et al., “Auto-Encoding Scene Graphs for Image Captioning”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10677–10686, IEEE, Long Beach, CA, USA (2019) [doi:10.1109/CVPR.2019.01094]

[15] J. Johnson et al., “Image retrieval using scene graphs”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3668–3678, IEEE, Boston, MA, USA (2015) [doi:10.1109/CVPR.2015.7298990]

[16] F. Kenghagho Kenfack et al., “RobotVQA — A Scene-Graph- and Deep-Learning-based Visual Question Answering System for Robot Manipulation”, IEEE/RSJ Intl. Conference on Intelligent Robots and Systems (IROS), pp. 9667–9674 (2020) [doi:10.1109/IROS45743.2020.9341186]

[17] J. Ji et al., “Action Genome: Actions As Compositions of Spatio-Temporal Scene Graphs”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10233–10244, IEEE, Seattle, WA, USA (2020) [doi:10.1109/CVPR42600.2020.01025]

[18] D. Li et al., “Representing Videos As Discriminative Sub-Graphs for Action Recognition”, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), p. 10, IEEE (2021)

[19] A. Markovitz et al., “Graph Embedded Pose Clustering for Anomaly Detection”, IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 10536–10544, IEEE, Seattle, WA, USA (2020) [doi:10.1109/CVPR42600.2020.01055]

[20] A. Nesen and B. Bhargava, “Knowledge Graphs for Semantic-Aware Anomaly Detection in Video”, IEEE Third Intl. Conference on Artificial Intelligence and Knowledge Engineering (AIKE), pp. 65–70 (2020) [doi:10.1109/AIKE48582.2020.00018]

[21] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection”, arXiv:2004.10934, arXiv (2020) [doi:10.48550/arXiv.2004.10934]

[22] H. Liu and P. Singh, “ConceptNet — A Practical Commonsense Reasoning Tool-Kit”, BT Technology Journal 22(4), 211–226 (2004) [doi:10.1023/B:BTTJ.0000047600.45421.6d]

[23] M. Pourreza, M. Salehi, and M. Sabokrou, “Ano-Graph: Learning Normal Scene Contextual Graphs to Detect Video Anomalies”, arXiv:2103.10502 [cs] (2021)

[24] T.-Y. Lin et al., “Microsoft COCO: Common Objects in Context”, Computer Vision – ECCV, D. Fleet et al., Eds., pp. 740–755, Springer Intl. Publishing, Cham (2014) [doi:10.1007/978-3-319-10602-1_48]

[25] J. Redmon and A. Farhadi, “YOLOv3: An Incremental Improvement”, arXiv:1804.02767, arXiv (2018)

[26] J. Redmon et al., “You Only Look Once: Unified, Real-Time Object Detection”, IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779–788, IEEE, Las Vegas, NV, USA (2016) [doi:10.1109/CVPR.2016.91]

[27] P. Veličković et al., “Deep Graph Infomax”, Intl. Conference on Learning and Representations (ICLR) **7**, (2019)

[28] J. Zhou et al., “Graph neural networks: A review of methods and applications”, AI Open 1, 57–81 (2020) [doi:10.1016/j.aiopen.2021.01.001]

[29] Ying Liu, Luyao Geng, Weidong Zhang, Yanchao Gong, and Zhijie Xu, "Survey of Video Based Small Target Detection”, Journal of Image and Graphics, Vol. 9, No. 4, pp. 122-134, (2021) [doi: 10.18178/joig.9.4.122-134]

AUTHORS' BACKGROUND

Your Name	Title*	Research Field	Personal website
Mia Siemon	Industrial PhD Candidate	Computer Vision, Anomaly Detection, Graphs, Ontologies	https://dk.linkedin.com/in/mia-sa-ni-sie
Kamal Nasrollahi	Full Professor at Aalborg University, and Head of Machine Learning at Milestone Systems	Computer Vision, Image Processing, Robot Vision, Face Recognition	https://dk.linkedin.com/in/kamal-nasrollahi-676b3352
Thomas B. Moeslund	Full Professor at Aalborg University (AAU), Head of Visual Analysis and Perception Lab at AAU, Head of Center for AI for the People, AAU	Computer Vision, Image Analysis, AI, XAI, Machine Vision, Pattern Recognition, Machine Learning, Perception, HCI, and Robotics	https://vbn.aau.dk/da/persons/103282

*This form helps us to understand your paper better, the form itself will not be published.

*Title can be chosen from: master student, Phd candidate, assistant professor, lecture, senior lecture, associate professor, full professor