A Neuro-Symbolic Approach with Reinforcement Learning for Explainable Anomaly Detection in Pedestrian Video Sequence

Jaeil Park and **Sung-Bae Cho**
Department of Computer Science, Yonsei University, Seoul 03722, South Korea
{wodlf603, sbcho}@yonsei.ac.kr

Abstract

Video anomaly detection in pedestrian streets requires to explain the anomaly because of its danger, such as a car moving on a pedestrian road, and to interact with supervisors with question answering. To explain the anomaly, the methods based on neural networks such as SHAP have been investigated, but they have a limitation that only takes account of the properties of the abnormal objects and is not interactive with the supervisor. This paper proposes a video anomaly detection method supporting question answering with a reinforcement learning-based neuro-symbolic approach. After converting a question into executable programs, it is operated on a scene graph with the video anomaly detection result to provide an answer for the question. After that, it executes reinforcement learning through a comparison between the result of the model and the ground-truth feedback from the supervisor. A question-answering experiment on UCSD dataset confirms that the proposed method answers the questions about anomalies, confirming 99% accuracy and demonstrating the causal inference through case analysis.

1 Introduction

Due to the extensive usage of surveillance cameras and the limitations of manpower, there is a growing demand for an automated video surveillance system [Fleck and Straßer, 2010]. One of the primary challenges that autonomous video surveillance systems face with is the automatic detection of anomalies, defined as unusual, uncommon, or irregular events occurring in complex and crowded environments [Cong et al., 2011; Xu et al., 2017].

In addition to performing detection using black box models, the anomaly detection model should provide explanations regarding the causes, outcomes, and necessary precautions for identifying visual scenarios that encompass real complex situations in a logical manner [Amarasinghe *et al.*, 2018]. Previous research on explanation has been predominantly focused on the properties of abnormal objects, thereby neglecting the comprehensive associations between objects that are linked to the risk of such abnormalities [Szymanowicz *et al.*, 2022; Szymanowicz *et al.*, 2023; Wu *et al.*, 2021]. This limited scope inhibits the ability to interact with diverse explanations and formulate strategies for risk mitigation. Consequently, the implementation of anomaly detection through question-answering grounded in visual data presents a critical challenge in assessing the explainable video anomaly detection systems.

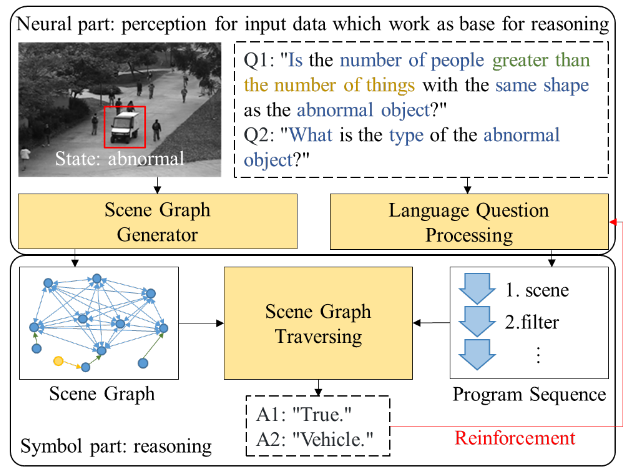


Figure 1. Solving visual question answering tasks in a pedestrian video environment with a combination of reasoning on scene graph traversing and cognition using reinforcement learning.

In the context of question-answering, integrating neural networks for recognition and reasoning between recognized symbols has been effectively employed across various tasks, such as phishing URL detection [Park *et al.,* 2021]. The effectiveness of this strategy is evidenced by the enhanced accuracy in synthetic photographic scenarios, characterized by their simplified data collection and object relationships [Yi *et al.,* 2018]. However, while the inference-based question-answering model has demonstrated validity within a restricted domain, its inference in complex real-world datasets is regarded as a significant problem in cognitive neural networks and question-answering [Amizadeh *et al.*, 2020]. For instance, a broader definition of the perception and question scope is required in pedestrian video environments, where the diversity of object types, relationships, and potential situations is increased. Furthermore, effective integration with previously established recognition methods should be considered.

In this paper, the range of possible questions and answers within pedestrian video surveillance environments is represented by predefined programs, each composed of question queries expressed in a domain-specific language. A program must balance usefulness and mapping performance. Before establishing the program, the properties, and relationships between the objects in the pedestrian image, forming the foundation of the program, are outlined for all images. This paper defines object properties and relations as shown in Table 1. Questions are translated into a sequence of the program commands $\{p_1, \ldots, p_s\}$ for simpler execution. The programs are classified into six categories, as illustrated in Table 2, and the format for each input and output is explicitly defined.

Moreover, we present a neuro-symbolic approach integrating reinforcement learning for scene graph construction and constant curvature manifold (CCM)-based anomaly detection to resolve the problem (see Figure 1). During image anomaly detection, the image is transposed to a latent space founded on a non-Euclidean framework, enabling the detection of anomalies within video content. Images are transformed into a graph structure, and an explicit inference process for the

Name	Description
Object property	
Shape	Person, bicycle, car, skateboard, wheelchair, cart, truck, others
Size	Small, large
Position	x-y location
Velocity	Computed by comparing x-y coordinates within frames
Object relation	
Relative position	Left, right, in front, behind, over, under
Relative size	Larger than, smaller than
Relative velocity	Faster than, slower than
Numbers	More than, less than
Equal	Same shape, same size, same position, same abnormal

Table 1: Definition of object property and relation on a pedestrian video environment.

Name	**Description**				
Function	**Input**	**Output**	**Function**	**Input**	**Output**
Basic program					
scene	-	Object list	count	Object list	Integer
unique	Object list	Object	exist	Object list	Boolean
relate	Object list	Object list	get_frame	Integer	scene
Filter program					
filter_size	List, size	List	filter_object	List, abnormal	List
filter_shape	List, shape	List	filter_scene	List, Int	List
filter_position	Position	List	filter_frame	List, Int	List
filter_velocity	List, integer	List			
Query program					
query_size	Object	Size	query_velocity	Object	Integer
query_shape	Object	Shape	query_type	Object	List
query_position	Object	Position			
Logic program					
AND	List, List	Object list	OR	List, List	Object list
Sameness program					
same_size	Object	Object list	same_position	Object	Object list
same_shape	Object	Object list	same_velocity	Object	Object list
Compare function					
equal_integer	Int, Int	Boolean	equal_shape	Shape, Shape	Boolean
equal_size	Size, Size	Boolean	less_then	Int, Int	Boolean
equal_color	Col, Col	Boolean	greater_then	Int, Int	Boolean

Table 2: Definition of domain-specific language set for visual question and answering.

Approach	Visual perception	QA processing	Environment
End-to-end neural network	Convolutional neural network	Long short-term memory [Antol *et al.*, 2015]	Synthetic visual scenes (CLEVR)
		Modular network with encoder-decoder [Hu *et al.*, 2017]	General objects (MS-COCO)
Neuro-symbolic	Mask R-CNN (Object tables)	Domain-specific language [Yi *et al.*, 2018]	Synthetic visual scenes (CLEVR)
	Mask R-CNN (Feature vectors)	Quasi-symbolic program execution [Mao *et al.*, 2019]	
	Faster R-CNN	Differentiable first-order logic [Amizadeh *et al.,* 2020]	General objects with scene graph (GQA)

Table 3: Previous research for combining deep learning and inference algorithms for visual question and answering.

detected anomaly generates an output using an algorithm designed to traverse the transformed graph. A neuro-symbolic system then takes on the task of scene-graph reasoning, integrating the anomaly detection results from pedestrian video data with programs translated from question sets. Scene graph reasoning, predicated on object properties and relationships, generates complex inferences suitable for question-answering in realistic settings. Furthermore, our proposed model is designed to learn with reinforcement feedback, using both predicted and original answers. This allows model for tuning toward more accurate answer prediction. We have verified the model's ability to handle complex queries.

We illustrate the superior performance of our method compared to extant visual question-answering techniques through graph reasoning. Our proposed method is evaluated on five distinct types of questions, finding that it outperforms the convnetional methods in terms of efficacy. Drawing from our experimental results, we posit that the integration of a neuro-symbolic system for scene-graph reasoning with a deep learning-based question-answering mechanism furnishes a level of inference that is highly suited to real-world environments.

2 Related Works

Video Anomaly Detection (VAD). Numerous studies have explored anomaly detection, typically supervised or unsupervised methods. Despite the challenges in data collection, supervised anomaly detection models have been studied due to their superior performance. Shin and Cho, for instance, developed a data augmentation method using a generative adversarial network (GAN) [Shin and Cho, 2018]. Conversely, unsupervised anomaly detectors overcome some limitations inherent in supervised models. Zhao *et al.* proposed a model that identifies unusual events in videos via dynamic sparse coding [Zhao *et al.,* 2011], while Liu *et al.* devised a future frame prediction model for anomaly detection [Liu *et al.*, 2018]. In the latter model, predicted frames are compared with actual future frames, with large differences indicating an anomaly. Further advancements in the field include end-to-end architecture for one-class classification [Sabokrou *et al.,* 2018], and a modified GAN method that learns an encoder simultaneously during training to develop an anomaly detection method [Zenati *et al.,* 2018]. They constructed an adversarially learned one-class classifier (ALOCC) composed of an encoder, decoder, and discriminator. However, defining data representation as a simple distribution can result in unseen data easily following that distribution, potentially causing novel data to be incorrectly classified as normal. To address this issue, we propose a one-class anomaly detection model based on a constant curvature manifold, a type of non-Euclidean space.

Scene Graph Generation. Numerous generative methods such as conditional random field (CRF), CNN, RNN, LSTM, and graph neural networks have been developed for scene graphs. CRF-based models like SG-CRF effectively model statistical correlation in visual relationships [Cong *et al.*, 2018]. With the advent of neural models for scene graph generation, CNN- and RNN-based models have been explored. BAR-CNN, a CNN-based model, incorporates an attention mechanism but may still suffer from limited receptive neuron regions [Kolesnikov *et al*., 2019]. The RNN-based Zoom-Net model successfully recognizes complex visual relationships through deep message propagation and interaction between local object features and global predicate features without a linguistic dictionary [Yin *et al.*, 2018]. Despite the success of these models, GCN has proven to be highly effective in graph reasoning tasks, leading to numerous researchers exploring scene graph generation methods based on the graph [Goller and Kuchler, 1996; Gori *et al.*, 2005]. Graph R-CNN, for example, trims the original scene graph to generate sparse candidate graph structures [Yang *et al.*, 2018]. In this paper, we adopt Graph-RCNN, considering its efficiency and effectiveness in generating scene graphs within complex scenarios.

Question-Answering. Table 3 outlines the methods combining deep learning and inference algorithms for visual question-answering, categorized by approach, method, and environment. Initial attempts to implement image recognition and processing, as well as mapping with neural networks, defined visual question-answering tasks within a synthetic environment [Antol *et al.*, 2015]. Several methods using modular neural networks demonstrated the necessity of distinguishing between recognition and natural language processing tasks [Hu *et al.*, 2017]. In the neuro-symbolic approach, which combines inference algorithms with deep learning, symbol grounding and inference methods of objects were examined [Yi *et al.*, 2018]. The research aiming to develop domain-specific languages and symbolic processes for query and relationship representation [Amizadeh *et al.*, 2020] demonstrated high-performance question-answering compared to human respondents in synthetic environments [Mao *et al.*, 2019]. Based on previous studies, this paper redefines objects and query range to extend the neuro-symbolic approach to more complex pedestrian video surveillance environments and enhances the practicality by incorporating an anomaly detection module with neural networks.

3 Methodology

Figure 2 illustrates the proposed method in this paper. An autoencoder employing a constant curvature manifold detects anomalies, and a scene graph is formulated by integrating anomaly detection outcomes with object detection results from pedestrian video data. The input questions and associated programs are mapped onto a supervised long short-term memory (LSTM) encoder-decoder framework. The set of programs, extracted from the input questions, executes a filtration process with scene graph traversal. This methodology produces the outcomes by applying a specific program to a group of nodes within a scene graph. After that, the model is trained to generate suitable responses via reinforcement learning.

3.1 Anomaly Detection with Autoencoder

Variational autoencoders (VAEs) or generative adversarial networks (GANs) may not be well-suited for learning complex data representations. We aim to address this issue using a constant curvature manifold in the latent space. As a result, even when a novel anomaly appears, it is readily simulated by the normal variance. This phenomenon can be easily observed in videos with minor changes, where the background remains fixed while only the object changes. The representation that our model learns is based on a constant curvature manifold, which belongs to a class of non-Euclidean spaces.

The d-dimensional CCM $\mathcal{T}$ is a Riemannian manifold characterized by a constant curvature $\kappa \in \mathbb{R}$. It can be defined as follows:

$$\boldsymbol{\mathcal{T}} = \{\boldsymbol{x} \in \mathbb{R}^{\boldsymbol{d+1}} | < \boldsymbol{x}, \boldsymbol{x} >= \boldsymbol{\kappa^{-1}}\} \tag{1}$$

where $<\cdot,\cdot>$ denotes a scalar product. In the CCM, it is defined from the pseudo-Euclidean scalar product:

$$< \boldsymbol{x}, \boldsymbol{y} > = \boldsymbol{x^T} \begin{pmatrix} \boldsymbol{I_{d\times d}} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{-1} \end{pmatrix} \boldsymbol{y} \tag{2}$$

where $I_{d\times d}$ is the identity matrix with size of d and T means transpose operator.

The three components are trained to define the data representation as the constant curvature manifold. An encoder g is trained to project the input data into latent space while the features of data are maintained. A discriminator D learns to distinguish features $g(x')$ of normal data from other extracted features. The encoder is forced to project x and x' to the same point which follows a CCM.Compared to the previous works [Cruz-Esquivel and Guzman-Zavaleta, 2022; Wang *et al.*, 2022; Chang *et al.*, 2020], our discriminator has compressed features as input, resulting in the small size of the model. In this process, the encoder is forced to project x and x' to the same point which follows a CCM as shown in Figure 3, and the discriminator is trained to classify $g(x)$, $g(x')$, and z. Therefore, to explicitly verify whether the trained latent

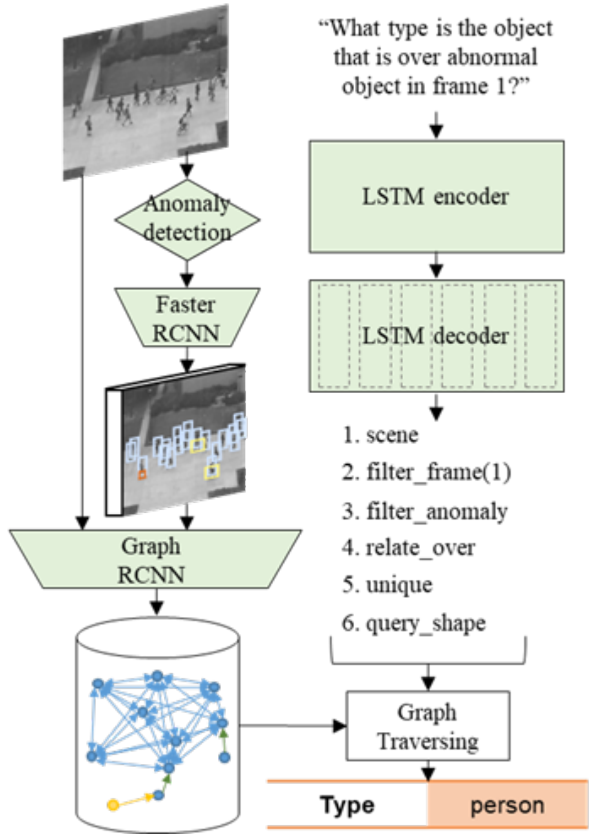


Figure 2. An illustration of the proposed method. Anomalies have been detected with autoencoder with CCM, which is added to the scene graph generated from graph-RCNN. Questions are translated into executable programs with LSTM, and neuro-symbolic integration is applied with scene graph traversal.

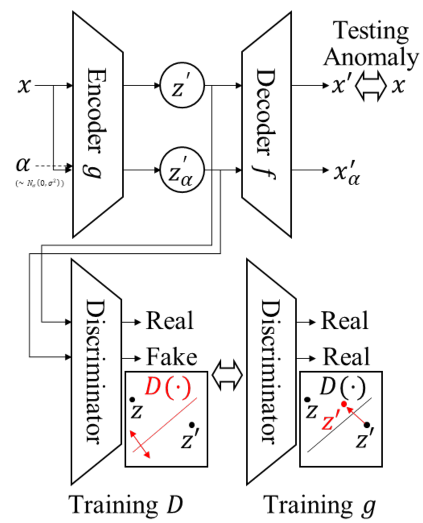


Figure 3. Structure of the anomaly detector with CCM.

space forms CCM, we add a membership function $\mu(\cdot)$ as follows:

Algorithm 1: Anomaly Detection Training Process

Data: hyperparameters
Result: discriminator D, autoencoder AE, encoder g, and decoder f
for $i = 1,\dots,M$ **do**
  for $j = 1,\dots,N$ **do**
    Sample x and x' form X and $X + N_\sigma$
    Sample z from CCM
    $\mathcal{L}_D \leftarrow \mathcal{L}_D - \frac{\partial \mathcal{L}_D}{\theta_D}(x, x', z)$
    $\mathcal{L}_{AE} \leftarrow \mathcal{L}_{AE} - \frac{\partial \mathcal{L}_{AE}}{\theta_D}(x)$
    $\mathcal{L}_g \leftarrow \mathcal{L}_g - \frac{\partial \mathcal{L}_g}{\theta_g}(x, x')$
    $\mathcal{L}_f \leftarrow \mathcal{L}_f - \frac{\partial \mathcal{L}_f}{\theta_f}(x, x')$
    $\mathcal{L}_D \leftarrow \mathcal{L}_D - \frac{\partial \mathcal{L}_D}{\theta_D}(x, x', z)$
  end
end
return f, g and D

$$\boldsymbol{\mu(z) = exp(\frac{-(<z,z>-\kappa^{-1})^2}{2\sigma^2})} \quad (3)$$

where σ is the hyperparameter to control the scale of CCM. The final forms of the objective function $\mathcal{L}_g$ and $\mathcal{L}_D$ for the encoder and the discriminator are as follows:

$$\boldsymbol{\mathcal{L}_g} = \begin{matrix} \mathbb{E}_{x\sim X}\left[\boldsymbol{l}\left(\boldsymbol{x}, \boldsymbol{f}\left(\boldsymbol{g}(\boldsymbol{x})\right)\right)\right] + \\ \mathbb{E}_{x'\sim X+N_\sigma}\left[\boldsymbol{log}\left(\boldsymbol{1} - \left(\boldsymbol{D}\left(\boldsymbol{g}(\boldsymbol{x'})\right) + \boldsymbol{\alpha\mu}(\boldsymbol{x'})\right)\right)\right] \end{matrix} \quad (4)$$

$$\boldsymbol{\mathcal{L}_D} = \begin{matrix} \mathbb{E}_{x\sim X, z\sim CCM}\begin{bmatrix} \mathbf{log}(\mathbf{1} - (\frac{\boldsymbol{D}(\boldsymbol{g}(\boldsymbol{x}))+\boldsymbol{D}(\boldsymbol{z})}{\mathbf{2}} \\ +\boldsymbol{\alpha\mu}(\boldsymbol{x}))) \end{bmatrix} \\ +\mathbb{E}_{x'\sim X+N_\sigma}[\mathbf{log}(\boldsymbol{D}(\boldsymbol{g}(\boldsymbol{x'})) + \boldsymbol{\alpha\mu}(\boldsymbol{x'})] \end{matrix} \quad (5)$$

l is a binary function to measure the difference between the input data and the reconstructed data. α is a hyperparameter for balance between the outputs of the discriminator (implicit verification) and the membership function (explicit verification), and α is a hyperparameter for balance.

$$\boldsymbol{\mathcal{L}_f} = \mathbb{E}_{x\sim X, \alpha\sim N_\sigma}[\boldsymbol{l}(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{x}))), \boldsymbol{l}(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{x}+\boldsymbol{\alpha})))] \quad (6)$$

$$\boldsymbol{\mathcal{L}_{AE}} = \mathbb{E}_{x\sim X}[\boldsymbol{l}(\boldsymbol{x}, \boldsymbol{f}(\boldsymbol{g}(\boldsymbol{x})))] \quad (7)$$

The final objective function for the proposed one-class anomaly detection model is shown in equation (8). To balance each term, we use the hyperparameters β, γ , and δ. Algorithm 1 shows the whole training process.

$$\boldsymbol{\mathcal{L} = \mathcal{L}_D + \beta\mathcal{L}_g + \gamma\mathcal{L}_f + \delta\mathcal{L}_{AE}} \quad (8)$$

where M is the number of epochs and N is the number of batches.

3.2 Scene Graph Generation using Graph R-CNN

The scene graph describes the properties and relationships of objects. Given a set of object property categories $C = \{C_1, \dots, C_m\}$ and a set of object relationship categories R, a scene graph is a tuple (O, E) where $O = \{o_1, \dots, o_n\}$ is a set of objects with each o_i, an object that $o_i = \{c_{i1}, c_{i2}, \dots, c_{im}\}$ where $c_{ij} \in C_j$, and $E \subseteq O \times R \times O$ is a set of directed edges of the form (o_i, r, o_j) where $o_i, o_j \in O$ and $r \in R$. For this scene graph, object property and relation are defined as the same as those of the questions (Table 1 and Table 2).

This paper details the transformation of images from pedestrian video sequences into scene graphs via a three-step procedure based on the defined scene graph and object property. Initially, a 3D convolution operation-based autoencoder, factoring in a time axis, determines the normality of the corresponding image. Subsequently, objects within images are identified where anomalies have been detected with Faster R-CNN. Lastly, the detection outcome is produced through Graph R-CNN in conjunction with the original image.

Faster R-CNN undertakes object detection for scene graph parsing. The model proposed integrates anomaly detection results from an autoencoder with constant curvature manifold and an image to detect objects, with an accompanying representation of their normality.

Graph R-CNN, a leading method among scene graph generation algorithms, successfully elucidates the relationships between objects more effectively. It employs a relationship proposition network (RePN) that efficiently manages secondary potential relationships between image objects and a graph convolutional network (GCN). In this paper, the images of pedestrian video frames are input into corresponding algorithm models, trained with the VQA dataset, preserving valid information correlating to the predefined object properties. The resultant scene graph facilitates knowledge representation that can more distinctly express object relationships while safeguarding information on the objects.

3.3 Neuro-Symbolic QA with Reinforcement

In the proposed model, a question is translated into a sequence of programs $\{p_1, \dots, p_s\}$ via an LSTM encoder-decoder structure. Scene graph traversing is performed using the corresponding translated program and the resultant scene graph from section 3.1. Each program operates on a set of nodes in the scene graph. For instance, the "scene" program returns all objects in the current scene. Programs other than "relation" and "scene" do not require any relational information. Each of these objects in the set is processed by an "if-else" operation, and the resulting output is calculated.

Programs associated with relationships require information about the relationships between objects. This model employs a method of searching through the edges of the scene graph. It verifies whether an edge, corresponding to a connection for filtering, is connected to each node for a set of nodes that are used as input when the edge is present. Then, the program for connection proceeds by calculating a set of nodes comprised of target nodes and outputs it. This process is illustrated in Algorithm 2. Through this algorithm, logical reasoning for each program stage becomes feasible.

In this paper, a two-stage procedure is implemented to train LSTM, with the aim of elucidating the mapping between a question and its corresponding program. Initially, a few ground truth question-program pairs are extracted from the training set to pretrain the model under direct supervision. Subsequently, the model is paired with a deterministic program executor. Reinforcement learning is then employed to

Algorithm 2: Scene Graph Traversing Algorithm

Data: scene graph $G = \{O, E\}$ and program sequence $P = \{p_1, p_2, \ldots, p_n\}$
Result: traversing result – answer to question
for $p_i \in P$ **do**
    if p_i is “scene” **do**
        $S.push(\phi)$
    else
        if p_i is in “relation” **then**
            $O_{org} = S.pop()$
            $O_{new} = \phi$
            for $o_j \in O_{org}$ **do**
                for $e_{jk} \in E$ where $e_{jk} = \overline{o_j o_k}$ **do**
                    if e_k is relation in p_i **then**
                        $O_{new} = O_{new} + \{e_k\}$
                    end
                end
            end
            $S.push(O_{new})$
        else
            $S = p_i(S)$
        end
    end
end
return $S.pop()$

fine-tune the LSTM, utilizing a larger dataset of question-answer pairs. Notably, only the accuracy of the execution result is used as the reward signal in this reinforcement learning phase.

Employing reinforcement learning for question and answer pairs contributes to generating more precise responses to inquiries. The decision to respond to the input image and query serves as a reward signal r, wherein the value of $r - b$ is propagated for model learning by establishing a baseline b to inhibit decay. The value of b is initially set to zero and is subsequently updated whenever a reward value manifests, shown as equation (9), thereby modulating the learning of extant models.

$$\boldsymbol{b} \leftarrow (\boldsymbol{1} - \boldsymbol{\alpha_{decay}})\boldsymbol{r} + \boldsymbol{\alpha_{decay}}\boldsymbol{b} \tag{9}$$

4 Experiments

4.1 Real-World Pedestrian Video Dataset

In order to evaluate the efficacy of the proposed method, we employ the UCSD pedestrian datasets, which are collected from stationary CCTV footage. This data comprises pedestrians and various moving objects captured moving in both directions. As in Table 1, the object attribute table generated from this data includes combinations of two to four-wheeled vehicles (bicycles, cars, skateboards, wheelchairs, carts, and trucks), along with various backgrounds (wood, roads, and grass).

In this paper, question-program pairs are formulated based on the objects within an image. The program is comprised of a sequence of domain-specific languages, as specified in Table 2, and each pair originates from a predefined template. The queries have been categorized into five types, each typified by its distinct properties.

“Querying Attribute” refers to inquiries about an object's characteristics, including queries concerning the attributes of anomaly objects. “Compare Attribute” involves the comparison of attributes between two objects and contains queries that can also determine anomaly attributes. “Exist” and “count” are demarcated as queries about the existence and quantity of specific objects, respectively. Lastly, “compare Number” is classified as a query that contrasts the number of objects across various sets.

4.2 Question and Answering Performance

Table 4 compares the performance of our method with the conventional question-answering methods, segregated by program type. In scenarios where image recognition using convolutional neural networks is coupled with question processing using LSTM, and mapped using simple supervised learning, our method seldom misclassifies, exhibiting an accuracy of 0.9971. This contrasts starkly with the considerably lower accuracy of 0.6457 when the number of objects is precisely specified in the table of perceived object properties. Furthermore, in complex environments such as pedestrian video sequences, our method, in combination with inference capabilities, outperforms the encoder-decoder approaches based on modular neural networks, achieving an accuracy of 0.9991 against the latter's 0.9232.

Method	Count	Exist	Compare number	Compare attribute	Query attribute	Overall accuracy
CNN-LSTM [Antol et al., 2015]	64.57%	87.44%	53.78%	77.47%	77.47%	72.15%
Mask R-CNN	85.23%	92.93%	83.45%	90.68%	92.68%	88.99%
Module network with Encoder-decoder [Hu *et al.*, 2017]	86.77%	96.61%	86.48%	96.51%	95.27%	92.32%
Ours	99.71%	99.97%	99.96%	99.93%	99.98%	99.91%

Table 4: 10-fold cross-validation of accuracy with other methods by query type.

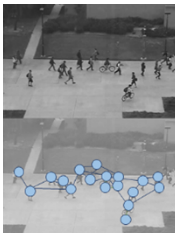
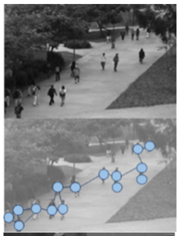
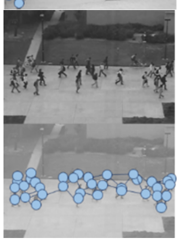
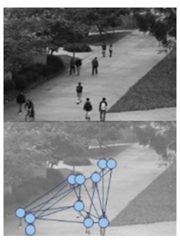
Scene	Query	Question	Program Representation	Answer (P&L)/ Answer (Graph)
	Count	What number of large normal persons are behind the small man?	scene filter size[small] unique relate[behind] filter size[large] filter anomaly[normal] filter shape[person] count	14/15
	Exist	Are there any things in front of the small normal person?	scene filter size[small] filter anomaly[normal] filter shape[person] unique relate[front] exist	False / True
	Compare number	Are there more humans on the left side of the scene than on the right?	scene filter position[left] filter shape[person] scene filter position[right] filter shape[person] greater than	True / False
	Count	What number of large normal persons are behind the small man?	scene filter size[small] unique relate[behind] filter size[large] filter anomaly[normal] filter shape[person] count	7/8

Table 5: Program representation and scene-graph for each case of correct response.

For every classification, our method exhibits the highest accuracy. Notably, our method achieves a remarkable accuracy of 0.9996 in the "compare number" classification, the most challenging category that records the lowest figure for all other algorithms. This demonstrates the potential of the neuro-symbolic approach in tackling problems that could yield varied and complex values, such as numerical comparison, and affirms the role of the scene graph in bolstering this capability. In addition, we also report higher accuracy in "query attribute" and "count" categories, which can lead to complex results and require precise determination, respectively.

Table 5 shows the instances where questions and answers fail when employing data inclusive of object properties and locations, but succeed when scene graph data is employed. In cases where relational information is required for questions, misidentification of relationships frequently occurs based on data with object property and location. However, when the proposed method is adopted for image information representation, the relational information can be more accurately handled even with more complex programs.

5 Conclusions

In this paper, we propose a neuro-symbolic visual question-answering method tailored for pedestrian anomaly video sequences, which closely resemble real-world environments. This method is facilitated by defining object properties, relationships, and question coverage and incorporating a scene graph generator as well as an anomaly detector. The proposed method demonstrates considerable accuracy of 0.9978 across five types of queries.

However, the proposed method's inference algorithm, designed to map questions and answers, is implemented as a basic filter algorithm operation. This approach needs validation in the general image field, where object relationships are more complex than in pedestrian video sequences. Particularly, as the emergence of various objects tends to complicate the scene graph, thereby increasing computational demand, a learning method that considers computational optimization will be required in the future work.

Acknowledgements

This work was supported by the Yonsei Fellow Program funded by Lee Youn Jae, and Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (No. 2020-0-01361, Artificial Intelligence Graduate School Program (Yonsei University); No.2021-0-02068, Artificial Intelligence Innovation Hub).

References

[Amarasinghe *et al.*, 2018] K. Amarasinghe, K. Keney, and M. Manic. Toward explainable deep neural network based anomaly detection. *11th Int. Conf. on Human System Interaction*, pp. 311–317. IEEE, 2018.

[Amizadeh *et al.*, 2020] S. Amizadeh, H. Palangi, A. Polozov, Y. Huang, and K. Koishida. Neuro-symbolic visual reasoning: Disentangling. *Int. Conf. on Machine Learning,* pp. 279–290. PMLR, 2020.

[Antol *et al.*, 2015] S. Antol, A. Agrawal, J. Lu, M. Mitchell, D. Batra, C. L. Zitnick, and D. Parikh. VQA: Visual question answering. *IEEE Int. Conf. on Computer Vision*, pp. 2425–2433, 2015.

[Chang *et al.,* 2020] Y. Chang, Z. Tu, W. Xie, and J. Yuan. Clustering driven deep autoencoder for video anomaly detection. *16th European Conf. on Computer Vision, Part XV 16*, pp. 329–345. Springer, 2020.

[Cong *et al*., 2011] Y. Cong, J. Yuan, and J. Liu. Sparse reconstruction cost for abnormal event detection. *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3449–3456. IEEE, 2011.

[Cong *et al*., 2018] W. Cong, W. Wang, and W.-C. Lee. Scene graph generation via conditional random fields. *arXiv preprint arXiv:1811.08075*, 2018.

[Cruz-Esquivel and Guzman-Zavaleta, 2022] E. Cruz-Esquivel and Z. J. Guzman-Zavaleta. An examination on autoencoder designs for anomaly detection in video surveillance. *IEEE Access*, 10:6208–6217, 2022.

[Fleck and Straßer, 2010] S. Fleck and W. Straßer. Privacy sensitive surveillance for assisted living–A smart camera approach. *Handbook of Ambient Intelligence and Smart Environments*, pp. 985–1014, 2010.

[Goller and Kuchler, 1996] C. Goller and A. Kuchler. Learning task-dependent distributed representations by backpropagation through structure. *Int. Conf. on Neural Networks*, vol. 1, pp. 347–352. IEEE, 1996.

[Gori *et al.*, 2005] M. Gori, G. Monfardini, and F. Scarselli. A new model for learning in graph domains. *IEEE Int. Joint Conf. on Neural Networks*, vol. 2, pp. 729–734, 2005.

[Hu *et al.*, 2017] R. Hu, J. Andreas, M. Rohrbach, T. Darrell, and K. Saenko. Learning to reason: End-to-end module networks for visual question answering. *IEEE Int. Conf. on Computer Vision*, pp. 804–813, 2017.

[Kolesnikov *et al.*, 2019] A. Kolesnikov, A. Kuznetsova, C. Lampert, and V. Ferrari. Detecting visual relationships using box attention. *IEEE/CVF Int. Conf. on Computer Vision Workshops*, pp. ???–???, 2019.

[Li *et al.*, 2018] Y. Li, W. Ouyang, B. Zhou, J. Shi, C. Zhang, and X. Wang. Factorizable net: An efficient subgraph-based framework for scene graph generation. *European Conf. on Computer Vision*, pp. 335–351, 2018.

[Liu *et al.*, 2018] W. Liu, W. Luo, D. Lian, and S. Gao. Future frame prediction for anomaly detection–A new baseline. *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 6536–6545, 2018.

[Mao *et al.*, 2019] J. Mao, C. Gan, P. Kohli, J. B. Tenenbaum, and J. Wu. The neurosymbolic concept learner: Interpreting scenes, words, and sentences from natural supervision. *arXiv preprint arXiv:1904.12584,* 2019.

[Park *et al.*, 2021] K.-W. Park, S.-J. Bu, and S.-B. Cho. Evolutionary optimization of neuro-symbolic integration for phishing URL detection. *Int. Conf. on Hybrid Artificial Intelligence Systems*, pp. 88–100. Springer, 2021.

[Sabokrou *et al*., 2018] M. Sabokrou, M. Khalooei, M. Fathy, and E. Adeli. Adversarially learned one-class classifier for novelty detection. *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3379–3388, 2018.

[Shin and Cho, 2018] W. Shin and S.-B. Cho. CCTV image sequence generation and modeling method for video anomaly detection using generative adversarial network. *Int. Conf. on Intelligent Data Engineering and Automated Learning*, pp. 457–467. Springer, 2018.

[Szymanowicz *et al.*, 2021] S. Szymanowicz, J. Charles, and R. Cipolla. X-man: Explaining multiple sources of anomalies in video. *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 3224–3232, 2021.

[Szymanowicz *et al.*, 2022] S. Szymanowicz, J. Charles, and R. Cipolla. Discrete neural representations for explainable anomaly detection. *IEEE/CVF Winter Conf. on Applications of Computer Vision,* pp. 148–156, 2022.

[Tang *et al.,* 2019] K. Tang, H. Zhang, B. Wu, W. Luo, and W. Liu. Learning to compose dynamic tree structures for visual contexts. *IEEE/CVF Conf. on Computer Vision and Pattern Recognition*, pp. 6619–6628, 2019.

[Wang et al., 2022] L. Wang, H. Tan, F. Zhou, W. Zuo, and P. Sun. Unsupervised anomaly video detection via a double-flow ConvLSTM variational autoencoder. IEEE Access, 10:44278–44289, 2022

[Wu *et al.*, 2021] C. Wu, S. Shao, C. Tunc, P. Satam, and S. Hariri. An explainable and efficient deep learning framework for video anomaly detection. *Cluster Computing*, pp. 1–23, 2021.

[Xu *et al.*, 2017] D. Xu, Y. Yan, E. Ricci, and N. Sebe. Detecting anomalous events in videos by learning deep representations of appearance and motion. *Computer Vision and Image Understanding*, 156:117–127, 2017.

[Yang *et al.*, 2018] J. Yang, J. Lu, S. Lee, D. Batra, and D. Parikh. Graph R-CNN for scene graph generation. *European Conf. on Computer Vision*, pp. 670–685, 2018.

[Yi *et al.*, 2018] K. Yi, J. Wu, C. Gan, A. Torralba, P. Kohli, and J. Tenenbaum. Neural-symbolic VQA: Disentangling reasoning from vision and language understanding. *Advances in Neural Information Processing Systems*, 31, 2018.

[Yin *et al.*, 2018] G. Yin, L. Sheng, B. Liu, N. Yu, X. Wang, J. Shao, and C. C. Loy. Zoom-net: Mining deep feature interactions for visual relationship recognition. *European Conf. on Computer Vision*, pp. 322–338, 2018.

[Zenati *et al*., 2018] H. Zenati, C. S. Foo, B. Lecouat, G. Manek, and V. R. Chandrasekhar. Efficient GAN-based anomaly detection. *arXiv preprint arXiv:1802.06222*, 2018.

[Zhao *et al.*, 2011] B. Zhao, L. Fei-Fei, and E. P Xing. Online detection of unusual events in videos via dynamic sparse coding. *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3313–3320. IEEE, 2011.