# A Neuro-Symbolic Agent for Video Anomaly Detection

Alessia Donata Camarda[1,*], Giovambattista Ianni[1] and Angelo Laface[1]

[1]University of Calabria, Italy

**Abstract**

The field of video anomaly detection has evolved towards more flexible approaches that increasingly leverage the semantic understanding capabilities of Vision-Language Models (VLMs). These models are particularly appealing as they enable reasoning about the high-level context of anomalous events, going beyond simple pattern deviations. However, despite their strong interpretive capabilities, VLMs are often computationally expensive and may be unsuitable for deployment in high-risk domains. In this paper, we propose a neuro-symbolic agent that combines a YOLO-based perception model for object detection with a symbolic reasoning component capable of identifying anomalies formalized through explicit rules. The reasoning module also serves as a filtering mechanism, selecting ambiguous situations that require deeper contextual analysis according to user-defined criteria. Such cases are then delegated to VLMs, reducing unnecessary reliance on these costly models. Finally, we discuss the limitations of the proposed approach and illustrate its applicability through a representative real-world scenario.

**Keywords**

Neurosymbolic AI, Video Anomaly detection, Answer Set Programming, Explainable AI

## 1. Introduction

Over the last decade, Video Anomaly Detection (VAD) [1] has evolved considerably, moving from traditional deep learning architectures towards more recent multimodal approaches, including Multimodal Large Language Models (MLLMs) [2] and Vision Language Models (VLM) [3]. This evolution reflects the growing need to capture not only deviations from patterns, but also higher-level contextual information. However, while VLMs are powerful at capturing this kind of information, they are often computationally expensive and may be unsuitable for continuous or resource-constrained deployment. Additionally, as VLMs are essentially neural networks, they operate as black boxes that are not fully understood and explained. While the lack of explainability may be acceptable in low-impact applications, such systems become unsuitable in high-risk scenarios where their outcomes can directly affect individuals. For this reason, we propose a neurosymbolic agent that detects objects in the scene by combining a YOLO model [4], a symbolic reasoning module based on Answer Set Programming (ASP), a well-known framework for declarative knowledge representation and logical reasoning [5], and a VLM. The symbolic component plays a dual role: it helps to directly identify previously formalized anomalies through explicit rules, while also acting as a filtering mechanism for situations that may require deeper semantic analysis. When symbolic reasoning alone is insufficient for a conclusive decision, the agent forwards candidate situations, identified through user-defined symbolic conditions, to a VLM for higher-level semantic interpretation, since more complex cases require a richer understanding of actions, context or intent. This architecture allows a greater control over what is considered anomalous and introduces improved interpretability through explicit rule-based modeling. The main contributions of this paper are the following: (i) a neuro-symbolic architecture for video anomaly detection integrating neural perception, ASP reasoning, and VLM-based semantic interpretation; (ii) an interpretable framework for explicitly modeling anomalous situations through logical rules. A proof of concept of the agent implementation can be found at the following link: https://github.com/DeMaCS-UNICAL/NS_VAD-CARLA2026.

*Joint Proceedings of the Workshops and Doctoral Consortium of the 42nd International Conference on Logic Programming (ICLP 2026), co-located with the Federated Logic Conference (FLoC 2026), July 18–19 and 24–25, 2026, Lisbon, Portugal.*

*Corresponding author.

alessiadonata.camarda@unical.it (A. D. Camarda); giovambattista.ianni@unical.it (G. Ianni)

© 2026 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings ceur-ws.org ISSN 1613-0073

published 2026-11-03

CEUR-WS.org/Vol-4311/CARLA_short_01.pdf

## 2. Related work

To the best of our knowledge, existing work combining YOLO and Answer Set Programming have focused on general complex event recognition, rather than on video anomaly detection. A first example is the work by Khan et al. [6], which proposes a hybrid architecture to recognize parking-related events in a surveillance scenario. A YOLO module is used to detect candidate objects from video frames, while event reasoning is performed through Event Calculus encoded under Answer Set semantics. A later extension of this line of research applies a similar framework to soccer videos. In this case, YOLO is used to extract visual information and simple events, which are then processed through Event Calculus and ASP rules to recognize more complex events such as free kicks and corner kicks [7]. While, these papers focus on complex event recognition, our work investigates how a symbolic layer can be used not only to represent events but also to explicitly model anomalous conditions.

## 3. Video Anomaly Detection Agent

The agent $A$ we propose aims to detect possible anomalies and reduce the need for complex and expensive deep learning modules. $A$ behaves according to an inner loop divided into three main stages as it is illustrated in Figure 1:

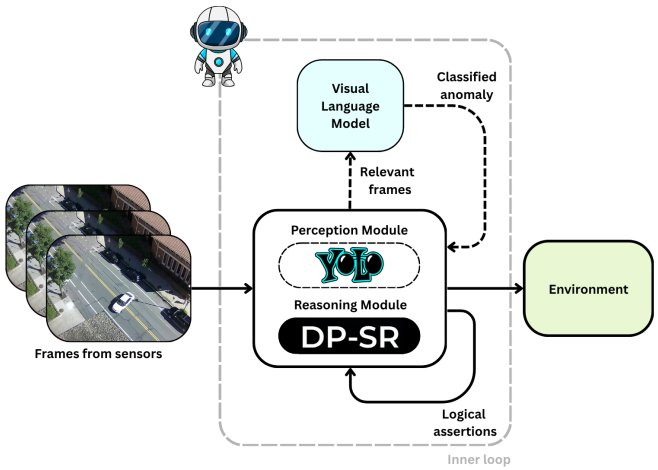



**Figure 1:** Overview of the proposed neuro-symbolic pipeline for video anomaly detection.

1. Thanks to its sensors and cameras, the agent receives a video of the external environment, which is then broken down into individual frames. This data is consequently sent to the perception module, which detects relevant objects in the scene and extracts symbolic information.
2. The information obtained in the previous step is fed into the reasoning module, which enables the system to assess whether any anomalies are present.
3. If the agent identifies a candidate anomaly requiring further investigation, it forwards the relevant frames to an external VLM to gain a better contextual understanding of the scene; the VLM then determines whether the proposed anomaly is present or not. Depending on the anomalies confirmed and their level of risk, the agent takes action in the external environment.

In the following section, we will describe a possible use case and explain how the agent functions. To do this, we will use the Street Scene dataset [8], which contains more than 200,000 high-resolution video frames captured by a static camera overlooking a two-way urban road.

## 4. Urban Traffic and Pedestrian Anomaly Detection

We selected the domain of Urban Traffic as the testbed for our approach. Our aim in deploying our agent within the urban-traffic domain is to reduce the number of road accidents. Specifically, agent A is designed to identify any anomalous, and possibly dangerous, behaviour among road users. These can be alerted in real-time so they can stop or limit their behavior while the anomaly is occurring. For example, if a pedestrian crosses the road outside the pedestrian crossing, $A$ can emit a sound to alert them of their unusual behaviour, and the user can then correct it.

**Perception Module** Once the current frame has been obtained from the cameras, it is fed into a YOLO network for processing and object detection. The model can also be provided with a set of manually defined semantic regions associated with the scene. Each region is represented as a polygon with a semantic label, such as `pavement`, `cycle lane` or `motorway lane`. For each detection, the module returns a tracking identifier assigned by YOLO's tracking component, the predicted class, and the corresponding bounding box. This identifier is then used to link detections of the same physical object across consecutive frames, thereby enabling the same object to be tracked over time. Each tracked detection is then converted into symbolic facts and represented through a predicate of the form:

```
object(ObjectID, ClassName, X, Y)
```

where `ObjectID` is the identifier assigned by YOLO's tracking component, `ClassName` denotes the predicted object class, `X` and `Y` denote the cartesian coordinates of the center of the detected bounding box. The coordinate system follows the standard image convention, with the origin located at the top-left corner, the $x$-axis increasing from left to right, and the $y$-axis increasing from top to bottom. If semantic regions are provided, the module also produces facts describing the regions and the spatial relation between detected objects and regions:

```
area(AreaID, AreaType), in_area(AreaID, ObjectID)
```

The predicate `area/2` represents the semantic regions given as input, where `AreaType` specifies the meaning associated with the region. The predicate `in_area(AreaID, ObjectID)` states that the object identified by `ObjectID` is located within the region `AreaID` in the current frame. This latter predicate is then populated at each reasoning step on the basis of the detections produced for the current frame.

**Reasoning Module** Agent $A$ reasons about the external environment using data processed by the perception module. This data is used as input for an ASP system, in this case DP-sr [9, 10], together with a program provided by the user when the agent is initialized, to determine whether anomalies are present. In fact, when $A$ is started a set of logical rules must be provided as input to encode what the user considers anomalous and on which the agent must focus on its attention. Along with these rules, the user can instruct the agent to focus on specific areas within the scene. Specifically, within this module, two categories of predicates are introduced. These predicates are not intended as raw input facts, but are derived by the reasoning module according to user-defined ASP rules:

- `anomaly`: denotes anomalies directly recognized by the reasoning module. This predicate can be expressed either at frame level as `anomaly(ID, AnomalyType, T)`, or over a temporal interval as `anomaly(ID, AnomalyType, T1, T2)`.
- `candidate_anomaly`: denotes potentially anomalous situations requiring further semantic interpretation through the VLM. Similarly, this predicate can be represented either at frame level as `candidate_anomaly(ID, AnomalyType, Description, T)`, or over a temporal interval as `candidate_anomaly(ID, AnomalyType, Description, T1, T2)`.

Here, `AnomalyType` identifies the anomaly category, while `Description` provides a natural-language description later supplied to the VLM as contextual information. In particular, `candidate_anomaly` is

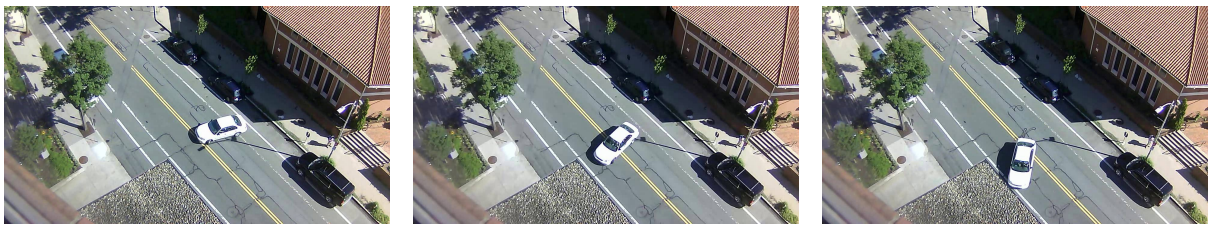
used for situations that cannot be conclusively classified through symbolic reasoning alone and therefore require higher-level semantic interpretation by the VLM. In the predicates above, `ID` is a symbolic identifier referring to the entity, object, or event instance involved in the anomaly. In most cases, it corresponds to the tracked object identifier appearing in `object/4`. However, some anomalies may concern the entire frame rather than a specific tracked object. In such cases, the frame identifier itself may be used as the reference entity. The temporal intervals in the `anomaly` and `candidate_anomaly` predicates are constructed by the reasoning module as it processes the incoming stream of frames. `T1` is set to the first timestamp at which the condition defining the anomaly holds, while `T2` is set once the condition ceases to be true, closing the interval. This allows the agent to express persistence directly, resulting in a relevant anomaly when a condition holds continuously for a minimum duration.

Considering the dataset at our disposal, pedestrians crossing outside designated pedestrian areas are considered anomalous. This anomaly can be modeled using the following ASP rules extended with constructs specific to the DP-sr system.

```
#temp in_crosswalk(ObjectID) :- in_area(AreaID, ObjectID), area(AreaID, crosswalk).
#temp in_traffic_lane(ObjectID) :- in_area(AreaID, ObjectID),
                                   area(AreaID, traffic_lane).
#temp pedestrian(ID) :- object(ID, person, _, _).
#temp p_outside_crosswalk(ID) :- pedestrian(ID), in_traffic_lane(ID),
                                 not in_crosswalk(ID).
anomaly(ID, jaywalking, @now) :- p_outside_crosswalk(ID).
```

**Visual Language Model** In the domain of urban traffic anomalies can be prohibited U-turns, pedestrian crossing outside allowed stripes, etc. Anomalies cannot always be captured through a precise symbolic encoding alone. Consider, for instance, a U-turn maneuver: while the trajectory reversal itself can be detected geometrically, establishing whether it occurred in a zone where U-turns are prohibited, or distinguishing it from a legal turn or an obstacle-avoidance maneuver, requires contextual cues that are not available to the symbolic layer alone. In such cases, symbolic reasoning can be used as a first step to derive a subset of events that may correspond to the target anomaly, based only on the available input predicates. This subset can then be passed to the VLM module, which carries out a higher-level contextual interpretation of the scene to determine whether the candidate events actually correspond to the anomaly of interest. Specifically, for each `candidate_anomaly(ID, AnomalyType, Description, T1, T2)` derived by the reasoning module, the agent extracts the frames spanning the interval $[T1, T2]$ in which the object identified by `ID` is present in the scene (respectively, for frame-level anomalies only the considered frame at time $T$ is extracted). These frames are then supplied to the VLM together with a prompt built from the `AnomalyType` and `Description` terms associated with the candidate anomaly. The VLM is then queried to provide an answer whether the described anomaly is confirmed or not, together with a short justification.

An example of a candidate anomaly detected by the reasoner is shown in Figure 2, which presents a sample of the frames extracted for the event and forwarded to the VLM.



(a) (b) (c)

**Figure 2:** Sample of the frames provided to the VLM and later classified as anomalous.

## 5. Conclusion and Future work

Although modern deep learning models offer accurate and efficient object detection, they lack explicit reasoning over contextual constraints and can be difficult to interpret when analyzing complex events. Vision language approaches alone may also be unsuitable for long-term deployment, particularly in continuous surveillance settings, due to computational and explainability contraints. To address these issues, we proposed an agent that integrates neural perception with symbolic reasoning. Object detections from YOLO are converted into symbolic facts and processed using ASP rules, enabling the explicit definition and transparent evaluation of anomalies and candidate anomalies. For ambiguous cases, a VLM is consulted to complement symbolic reasoning. Despite these advantages, the agents presents several limitations on which we are currently working on.

- The agent is sensitive to perspective distortions, so spatial reasoning based solely on image coordinates may not accurately reflect real-world relationships.
- The pretrained YOLO26x model [11] chosen for the tests is trained on the COCO [12] dataset. The latter lacks classes relevant to our target domain (i.e. cyclists) and is not optimized for challenging surveillance conditions, such as occlusions, motion blur, and non-standard perspectives.
- The absence of relevant frames may lead to ambiguous interpretations, affecting both the symbolic reasoning process and the VLM evaluation.

As this work constitutes a proof of concept, a quantitative evaluation on the Street Scene dataset is left for future work. Specifically, we plan to report a quantitative analysis over the number of VLM calls avoided through symbolic filtering, together with a comparison against YOLO-only and VLM-only baselines.

## Acknowledgments

This work was partially supported by the AI4Nature and SINTESI projects, funded under the PNRIC 2021-2027 program (DM n.307/2025).

## Declaration on Generative AI

Besides the usage of generative AI as described in the scientific contribution of the paper for the identification and evaluation of candidate anomalies in video frames, as described in the paper's scientific contribution, the authors used Claude Haiku 4.5 to do grammar and spelling check and rewriting of specific sentences. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

## References

[1] S. Gao, P. Yang, H. Guo, Y. Liu, Y. Chen, S. Li, H. Zhu, J. Xu, X. Zhang, L. Huang, The evolution of video anomaly detection: A unified framework from DNN to MLLM, CoRR abs/2507.21649 (2025).

[2] Y. Jin, J. Li, T. Gu, Y. Liu, B. Zhao, J. Lai, Z. Gan, Y. Wang, C. Wang, X. Tan, L. Ma, Efficient multimodal large language models: a survey, Vis. Intell. 3 (2025).

[3] A. Radford, J. W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, G. Krueger, I. Sutskever, Learning transferable visual models from natural language supervision, in: ICML, Proceedings of Machine Learning Research, PMLR, 2021, pp. 8748–8763.

[4] J. Redmon, S. K. Divvala, R. B. Girshick, A. Farhadi, You only look once: Unified, real-time object detection, in: CVPR, IEEE Computer Society, 2016, pp. 779–788.

[5] V. Lifschitz, Answer Set Programming, Springer, 2019.

[6] A. Khan, L. Serafini, L. Bozzato, B. Lazzerini, Event detection from video using answer set programing, in: CILC, CEUR Workshop Proceedings, CEUR-WS.org, 2019, pp. 48–58.

[7] A. Khan, L. Bozzato, L. Serafini, B. Lazzerini, Visual reasoning on complex events in soccer videos using answer set programming, in: GCAI, EPiC Series in Computing, EasyChair, 2019, pp. 42–53.
[8] B. Ramachandra, M. J. Jones, Street scene: A new dataset and evaluation protocol for video anomaly detection, in: WACV, IEEE, 2020, pp. 2558–2567.
[9] L. Laboccetta, E. Mastria, F. Calimeri, N. Leone, S. Perri, G. Terracina, Towards effective asp-based stream reasoning: Facilitate the reasoning over patterns of events, in: PPDP, ACM, 2024, pp. 18:1–18:12.
[10] DP-sr, Official Website, http://ares.mat.unical.it:20140/en/home, 2026.
[11] G. Jocher, J. Qiu, A. Chaurasia, Ultralytics yolo, 2023. URL: https://github.com/ultralytics/ultralytics.
[12] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, C. L. Zitnick, Microsoft COCO: common objects in context, in: ECCV (5), Lecture Notes in Computer Science, Springer, 2014, pp. 740–755.