NeoSGG: A Scene Graph Generation Framework for Video-Surveillance Tasks

Pierre Lefebvre
Léonard de Vinci Pôle Universitaire, Research Center
Courbevoie, France
pierre.lefebvre@devinci.fr

Steven Le Moal
De Vinci Higher Education, ESILV
Courbevoie, France
steven.le_moal@edu.devinci.fr

Ahmed Azough
Léonard de Vinci Pôle Universitaire, Research Center
Courbevoie, France
ahmed.azough@devinci.fr

Nicolas Travers
Léonard de Vinci Pôle Universitaire, Research Center
Courbevoie, France
nicolas.travers@devinci.fr

ABSTRACT

Video surveillance has developed considerably in the recent years. Analyzing the data generated by such systems has become a major challenge. To address this issue, we propose a framework for the creation of rich Labeled Property Graphs from video surveillance streams. It is based on 1) a Deep Learning pipeline architecture for video data extraction, 2) a graph database module to efficiently structure and store detections, and 3) a querying module to interact with generated graphs, enhancing the automatic analysis of scenes. Its modular architecture enables the feature extraction steps from the videos to be easily maintained, modified or interchanged. Our demonstration scenario shows the process of generating scene graphs from videos of several benchmark datasets. The audience will assist to an end-to-end execution of the pipeline showing the generation process and visualize generated graphs. They will have the opportunity to formulate queries using an interface illustrating several use case scenarios involving person re-identification and abandoned objects matching with their former owners.

1 INTRODUCTION

Video surveillance systems have developed considerably in recent years. Automatic analysis of the data produced by such systems has thus become a major challenge. Among the many potential risks is that posed by abandoned luggage. Most approaches can be broken down into two stages: the detection of static objects (motion estimation & background subtraction), and the recognition of abandoned objects [11]. While these techniques are effective in addressing challenges such as real-time alert triggering, they do not allow to extract complex situations. In fact, with previous approaches, it is often complex to identify luggage owners, even more with generalized types of object, or for luggage exchange or scenes including violence. They are therefore limited in terms of expressiveness and generalisation.

In this paper, we present NeoSGG, a framework for automatic detection and retrieval of complex scenes. Our use case here is the detection of abandoned objects. Our approach relies on Object Re-identification (Re-ID) and Scene Graph Generation (SGG) to generalize problems of complex events detection. Re-ID is a computer vision task involving identifying and tracking persons or objects from cameras [13]. On the other hand, SGG aims at transmuting a rich, contextual understanding of visual scenes into a structured representation [7]. By mapping the video scene's objects into nodes and their spatio-semantic and temporal relationships into edges in a Temporal Graphs, we capture a detailed overview of scene content and the interactions among its components. We propose 1) a Deep Learning pipeline to extract the video content, 2) a graph data model to structure and store the extracted content, and 3) a querying module for complex pattern retrieval and adaptive scene analysis which we apply here to the detection of abandoned luggage.

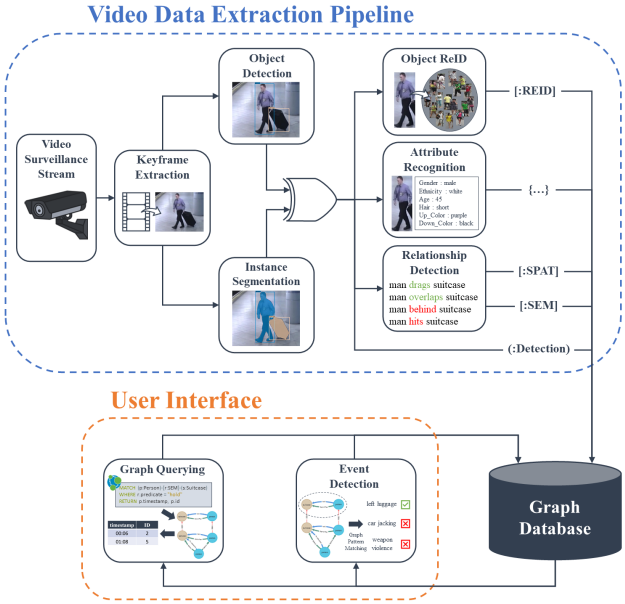

Figure 1: The NeoSGG Framework

The key idea of our approach relies on the transformation of video streams into graphs, allowing to bring deeper semantics which can be exploited to enhance person Re-ID. Existing approaches capture high-level semantics to describe a video content in an exclusive way, mainly focusing on models' performances and their inference speed [5]. In contrast, our approach choose to transform the video into an expressive and detailed graph with rich low level features and mid level semantics that enable for reasoning and retrieving specific events meeting the query of a user. Moreover, the modularity of our framework facilitates the evolution of the various feature extractors, allowing them to be more easily substituted or replaced, thus increasing its maintainability, adaptability and scalability.

2 THE NEOSGG FRAMEWORK

We propose the NeoSGG framework for transforming video surveillance streams into rich and structured temporal graphs. It enables the analysis of video content at spatiotemporal and semantic levels. It consists of a pipeline for extracting visual content from video surveillance streams and a graph database for structuring and storing data, and enabling pattern retrieval.

2.1 Pipeline Architecture

Our pipeline applies different steps of feature extraction from the videos and maps them into a graph using our Graph Data Model (Section 2.2). Figure 1 shows the layout of our pipeline: i) keyframe extraction, ii) object detection and instance segmentation, iii) attribute recognition, iv) spatio-semantic relationship detection and v) object re-identification. The pipeline ends with a comprehensive interface which enables the end user to query the graph database researching for complex scenes with a high expressiveness.

Keyframe Extraction is the process of summarizing a video stream by selecting the most representative images in terms of visual content and motion. For computational efficiency, it reduces the amount of data while preserving relevant scenes' visual content.

Object Detection consists in detecting all instances of persons and objects in an image. Detected objects of all classes are named *Detections* and their positions are saved as bounding boxes or contour polygons according to the chosen method.

Attribute Recognition refers to the task of identifying and categorizing specific attributes from detections such as age, gender, clothing color or other semantic features. There exists many sub-categories such as Pedestrian Attributes Recognition (PAR) or Facial Attributes Estimation (FAE).

Relationships Detection consists in understanding relationships between detections in a keyframe. It covers either general interaction between objects, as for Visual relationship Detection (VRD), or Human-Object Interaction (HOI). Relationships may be categorized as either spatial, focusing on the relative positions of detections, or semantic, highlighting their interactions.

Re-Identification is the computer vision task of connecting all instances of a detection over time across multiple keyframes. It relies on extracting their visual features and matching them using a distance metric and a re-ranking algorithm.

2.2 Graph Data Model and Querying

Graph Data Model. Conventional SGG approaches lead to poor expressiveness from a graph database point of view [7]. To tackle this issue, we propose to use a Graph Data Model to map and structure features extracted from a video. Our model belongs to the Labeled Property Graph [1] (LPG) category with temporal data. It is made up of spatial layers representing a captured scene at a discrete time $t \in T$ (Def. 2.1).

Definition 2.1 (LPG Temporal Graph). Let $\mathcal{G}^T$ be a temporal graph which definition is given below: $\mathcal{G}^T = (\mathcal{L}, \mathcal{R}_{reid})$ with $\mathcal{L}$ being the set of spatio-temporal layers, $\mathcal{R}_{reid}$ the set of re-identification edges, and T the time series of keyframes.

The special feature of our model is that it includes re-identification edges (Def. 2.3) between spatial layers (Def. 2.2) in order to link instances of the same detection and facilitate querying.

Definition 2.2 (Spatial-Semantic Layers). The spatial-semantic layer set $\mathcal{L}$ is defined as: $\mathcal{L} = \{\mathcal{L}_t \mid t \in [1, \ |\mathcal{L}|]\}$. $\mathcal{L}_t$ refers to a spatial-semantic layer of the temporal graph $\mathcal{G}^T$ and t to its observation time. Spatial-semantic layers $\mathcal{L}$, composed of detection nodes $\mathcal{N}$ and semantic or spatial relationships $\mathcal{S}$, are defined as: $\mathcal{L} = (\mathcal{N}, \mathcal{S})$. $\mathcal{N}(\mathcal{L}_t)$ and $\mathcal{S}(\mathcal{L}_t)$ denote respectively sets of nodes and spatial-semantic relationships within a given layer $\mathcal{L}_t$ of the graph.

Nodes $n \in \mathcal{N}$ belongs to a unique layer:

$$\forall\, n \in \mathcal{N}(\mathcal{L}_t), \forall\, \mathcal{L}_{t'} \mid \mathcal{L}_{t'} \neq \mathcal{L}_t, n \notin \mathcal{N}(\mathcal{L}_{t'})$$

Relationships $s_{n \to n'}$ link nodes from the same layer:

$$\mathcal{S}(\mathcal{L}_t) = \{s_{n \to n'} = (n, n', a, c) \mid n \in \mathcal{N}(\mathcal{L}_t) \wedge n' \in \mathcal{N}(\mathcal{L}_t) \wedge n \neq n'\}$$

with a a list of attributes of the spatial-semantic relationships and c its respective confidence in $[0, 1]$.

Definition 2.3 (Re-identification Edges). The set of Re-ID edges $\mathcal{R}_{reid}$ links nodes n and n' between spatial layers $\mathcal{L}_t$ and $\mathcal{L}_{t'}$:

$$\mathcal{R}_{reid} = \{r_{n \to n'} = (n, n', s) \mid n \in \mathcal{N}(\mathcal{L}_t) \wedge n' \in \mathcal{N}(\mathcal{L}_{t'}) \wedge \mathcal{L}_t, \mathcal{L}_{t'} \in \mathcal{G}^T \wedge t < t'\}$$

with $r_{n \to n'}$ a re-identification edge in $\mathcal{R}_{reid}$ between nodes n and n' and s the re-identification score in $[0, 1]$.

Nodes $n \in \mathcal{N}$ are re-identified at most once:

$$\forall\, n \in \mathcal{N}(\mathcal{L}_t), t > 0, \nexists (r_{n \to n'}, r_{n \to n''}) \in \mathcal{R}^2_{reid} \mid n' \in \mathcal{N}(\mathcal{L}_{t'}) \wedge n'' \in \mathcal{N}(\mathcal{L}_{t''}) \wedge t < t' \wedge t < t''$$

with n' and n'' belong to following layers.

Figure 2 gives the Graph Data Model [2] corresponding to previous definitions. Each node $n \in \mathcal{G}^T$ is of type `(:Detection)` where time $t \in T$ is the property "*timestamp*". Since our pipeline identifies different categories for detections, a hierarchy of types given by a thesaurus helps to target specific identifications.

Spatial and semantic relationships (Def. 2.2) are respectively typed as `[:SPAT]` and `[:SEM]` with corresponding extracted properties (within a temporal layer $\mathcal{L}_t$). Finally, Re-ID relationships (Def. 2.3) are added between nodes from distinct layers while checking the constraint of unicity.

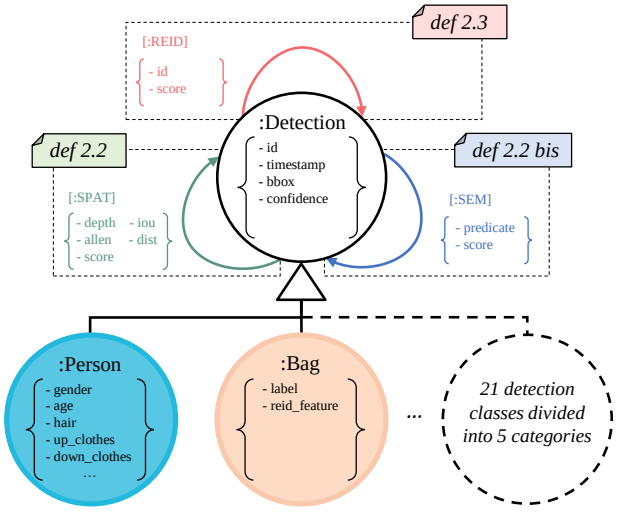

Figure 2: Temporal Graph Data Model

Graph Querying. To extract complex events from the graph $\mathcal{G}^T$ we need paths on intra (SPAT, SEM) and inter layers (re-ID). The former is a common expression pattern between nodes while Re-ID requires a chain to be composed between layers:

Definition 2.4 (Re-identification Chain). A re-identification chain is a sequence of distinct nodes linked by re-identification edges:

$$\mathcal{W}_{n_0 \to n_k} = (n_0, n_1, ..., n_{k-1}, n_k),\ k \in \mathbb{N}^* \\ |\ \forall\ i \in \mathbb{N}^*,\ i \leq k,\ \nexists\ r_{n_{i-1} \to n_i} \in \mathcal{R}_{reid}$$

3 IMPLEMENTATION

Our framework relies on different modules for features extraction, a Re-ID module, a mapping to the graph database and pattern extractions. We focus here on the pipeline implementation.

Keyframe Extraction Module. The implemented method for keyframe extraction [10] is based on ORB algorithm [9]. The module takes a video stream and computes its keypoints and descriptors. Keypoints are then associated with their two nearest neighbors from previous keyframe using a KNN algorithm. An image is returned as a new keyframe if the number of marked keypoints (lowest distances with following images) falls below a threshold. This method enables adaptive extraction, taking into account visual changes in terms of object updates within images.

Object Detection Module. We use two different methods to perform either object detection or instance segmentation.

First, `YOLOv5`[1] is a popular one-stage detector allowing a fast inference for object detection. Its architecture computes feature maps fusion at different scales to detect various size objects belonging to 80 classes and their coordinates. Five different model sizes are available, offering a good compromise between inference speed and detection performance.

Second, the `PointRend` [6] algorithm is built on top of a two-stage segmentation models. It performs pixel-based segmentation using small MLP networks to refine object borders. For feature extraction, either `Resnet50` and `Resnet101` backbones can be chosen from for inference speed or detection sharpness.

Attribute Recognition Module. For this module, we mainly focused on Pedestrian Attribute Recognition (PAR). We trained a multi-head CNN with a `Resnet50` backbone and 9 classification branches. We trained our model on `Market-1501 attributes` dataset [8], consisting of 32,000 hand annotated images of 1,500 different identities.

Relationships Detection Module. This module extracts spatial and semantic relationships between detections. Our spatial relationships relies on the use of `Adabins` [3], a SOTA transformer-based model dividing depth into bins whose centers are adaptively estimated for every keyframe. The model was trained on the `KITTI Depth` [12] dataset consisting of 94,000 outdoor RGB images and depth maps. We combined it with the `RCC8` standard applied to bounding boxes. Objects depth maps are employed to estimate their relative depths denoted as: *Same Plan*, *Behind* and *In Front Of*. Objects relative positions are determined by using their bounding box coordinates according to the connectivity classes (*Disconnected, Externally Connected, Equal, Partially Overlapping, Tangential Proper Part, Tangential Proper Part Inverse, Non-Tangential Proper Part*).

The semantic relationship detection task relies on `PViC` [14], a SOTA transformer-based detector. It uses box pairs positional embedding to promote attention on spatially close object and visual feature reintroduction using cross-attention to provide fine-grain contextual information in order to improve HOI classification. The model was trained on `HICO-DET` [4], a benchmark dataset for HOI detection and classification including 47,000 images with 117 predicates for 80 different object classes.

Re-Identification Module. This last module consists of three main parts: a backbone model for feature extraction, a distance metric to evaluate feature similarities and an ID reallocation algorithm based on similarity scores. For feature extraction, we selected the SOTA `AGW` baseline [13] which consists of a `Resnet50` architecture with an additional GeM layer and nonlocal attention blocks. We performed the training on `Market-1501` dataset using the configuration provided by the authors. Feature vectors of detections between two consecutive keyframes are then compared using a similarity matrix of which the coefficients are obtained using the cosine similarity.

Finally, IDs are reassigned using a *Hungarian Assignment* algorithm performed on the similarity matrix to maximize the sum of similarity scores for the final combination.

Graph Database. Each detection from the pipeline is mapped into a graph database using the Graph Data Model presented in Section 2.2. For this, we rely on the `Neo4j`[2] graph database.

The mapping of instances has been designed to be direct for detected categories and their properties (object Detection & attribute recognition), relationships between nodes (spatial, semantics and Re-ID). Consequently, each keyframe produces a new temporal layer $\mathcal{L} \in \mathcal{G}^T$ connected to others in the past.

The produced graph can be queried to extract events. Re-ID chain patterns (Def. 2.4) are translated as Cypher queries:

```
MATCH (X) -[:REID*]-> (Y)
```

For intra-layer paths, it corresponds to a pattern query mixing types and properties from both nodes and queries within a given layer:

```
MATCH (X{a}) -[:SEM|SPAT{b}]-> (Y{c})
```

4 DEMONSTRATION PLANS

Our demonstration scenario consists of three main interactive parts focusing on the execution of our pipeline and the process of database querying using our interface. The aim is to demonstrate the effectiveness of the `NeoSGG` framework for detecting abandoned objects such as backpacks, handbags and suitcases.

Pipeline Showcase. We begin with the demonstration of the pipeline with different steps for extracting structured scene graphs from video surveillance streams as introduced in Section 2.1. The audience will assist to each step of the full execution of our `NeoSGG` framework on several benchmark videos from `PETS 2006`[3] and `AVSS 2007`[4] benchmarks. The focus is placed on abandoned luggage detection. However, the genericity of our framework allows us to handle more complex situations, such as scenes of violence.

Simple Pattern. A second part is dedicated to presenting our interface and the generation process of Cypher queries. Figure 3 shows the visual aspect of our interface with a projection on the graph using the `Neo4j` GDBMS. It offers a good level of abstraction, enabling a novice user to formulate Cypher queries without any prior knowledge of database query language. The `Query Builder` section is made up of check-boxes and inputs to modify the various fields of Cypher queries and modulate its behavior.

[1] YOLOv5: https://zenodo.org/records/4154370

[2] Neo4j: https://neo4j.com/

[3] PETS 2006: https://ftp.cs.reading.ac.uk/pub/PETS2006/

[4] AVSS 2007: https://www.eecs.qmul.ac.uk/~andrea/avss2007_d.html

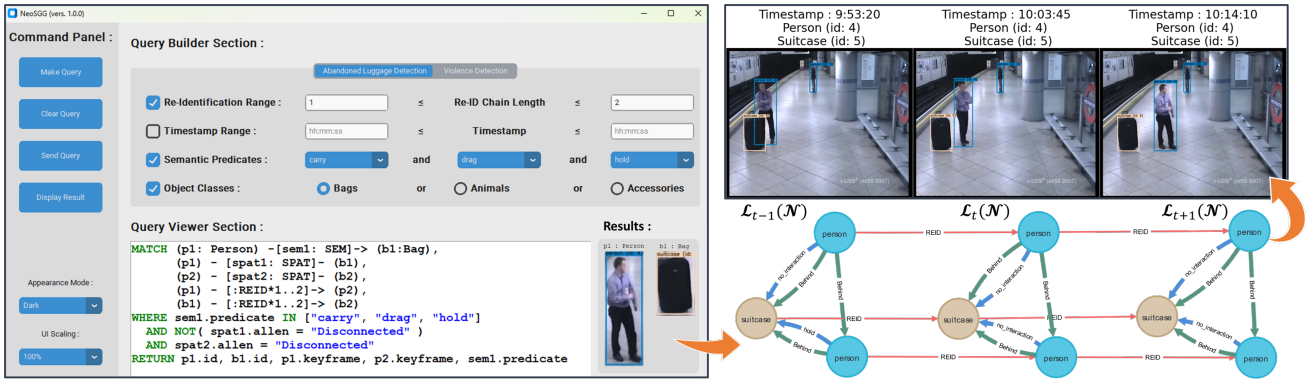

Figure 3: NeoSGG Interface with Abandoned Luggage Detection Use Case

The `Query Viewer` section allows more advanced users to modify the query directly. The query results appear as a temporal graph in the bottom right section showing abandoned objects (and their former owners) with their respective IDs, in the form of pop-up windows displaying a full view of the corresponding keyframes and their timestamps.

Below, we show an example of a Cypher query for the abandoned luggage use case:

```
MATCH (p1:Person) -[sem1:SEM]-> (b1:Bag),
      (p1) -[spat1:SPAT]-> (b1),
      (p1) -[:REID*1..2]-> (p2),
      (p2) -[spat2:SPAT]-> (b2),
      (b1) -[:REID*1..2]-> (b2)
WHERE sem1.predicate IN ["carry", "drag", "hold"]
  AND NOT(spat1.allen="Disconnected")
  AND spat2.allen="Disconnected"
RETURN p1.id, b1.id, p1.timestamp, p2.timestamp
```

It consists in detecting a pattern where a person `p1` is spatially close (*spat1* not disconnected) to a bag-type object `b1` and has a semantic relationship with the latter such as `carry`, `pull` or `hold`. We then recursively go back up the re-identification chain (*REID**) to find a keyframe in which the person and the bag are disconnected.

Complex Use Case. As previously mentioned, our framework is not limited to the lost luggage use case. It can be used for detecting specific use cases that are hardly detected with direct machine learning algorithms. Below is an example of a query designed to detect all the scenes of violence in which the assailant attacks more than 5 different victims with a weapon:

```
MATCH (p1_1:Person) -[sem1:SEM {predicate: "hit"}]-
        (p2_1:Person),
    (p1_x:Person) -[semx:SEM {predicate: "hit"}]-
        (p2_x:Person),
    (p1_1) -[:REID*]-> (p1_x:Person),
    (p1_1) -[:SEM_REL]- (w:Weapon)
WHERE NOT ( (p2_1) -[:REID*]-> (p2_x) )
WITH p1_1, count(*) AS NB
WHERE NB >= 5   RETURN p1_1, NB  ORDER Y NB DESC
```

On the same principle as for the previous query, we first detect all patterns in which a person `p1_1` (the assailant) hits another person `p2_1`. We then recursively descend the re-identification chains to check whether the re-identified assailant `p1_x` strikes different people (different `p2_x` re-ID with "*NOT*") with a weapon. We finally return the ID of the assailants with a minimum number of victims (here set to 5 or more).

Various types of queries will be applied on extracted graphs to showcase the expressiveness of our approach.

ACKNOWLEDGMENTS

This research is financially supported by the French Ministry of Defence - Innovation and Defence Agency.

REFERENCES

[1] Renzo Angles. 2018. The Property Graph Database Model. In *Workshop on Foundations of Data Management* (Cali, Colombia) *(AMW)*, Dan Olteanu and Barbara Poblete (Eds.), Vol. 2100. CEUR-WS.org.

[2] Maciej Besta, Robert Gerstenberger, Emanuel Peter, Marc Fischer, Michał Podstawski, Claude Barthels, Gustavo Alonso, and Torsten Hoefler. 2023. Demystifying Graph Databases: Analysis and Taxonomy of Data Organization, System Designs, and Graph Queries. *ACM Comput. Surv.* 56, 2, Article 31 (sep 2023), 40 pages.

[3] Shariq Farooq Bhat, Ibraheem Alhashim, and Peter Wonka. 2021. Adabins: Depth estimation using adaptive bins. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 4009–4018.

[4] Yu-Wei Chao, Yunfan Liu, Xieyang Liu, Huayi Zeng, and Jia Deng. 2018. Learning to Detect Human-Object Interactions. In *WACV'18*.

[5] Kevin Hsieh, Ganesh Ananthanarayanan, Peter Bodik, Shivaram Venkataraman, Paramvir Bahl, Matthai Philipose, Phillip B Gibbons, and Onur Mutlu. 2018. Focus: Querying large video datasets with low latency and low cost. In *USENIX Symposium on OSDI*. 269–286.

[6] Alexander Kirillov, Yuxin Wu, Kaiming He, and Ross Girshick. 2020. PointRend: Image Segmentation As Rendering. In *CVPR'20*. 9796–9805.

[7] Hongsheng Li, Guangming Zhu, Liang Zhang, Youliang Jiang, Yixuan Dang, Haoran Hou, Peiyi Shen, Xia Zhao, Syed Afaq Ali Shah, and Mohammed Bennamoun. 2024. Scene Graph Generation: A comprehensive survey. *Neurocomputing* 566 (2024), 127052.

[8] Yutian Lin, Liang Zheng, Zhedong Zheng, Yu Wu, Zhilan Hu, Chenggang Yan, and Yi Yang. 2019. Improving Person Re-identification by Attribute and Identity Learning. *Pattern Recognition* (2019).

[9] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R. Bradski. 2011. ORB: An efficient alternative to SIFT or SURF. *ICCV'11* (2011), 2564–2571.

[10] Klaus Schoeffmann, Manfred Del Fabro, Tibor Szkaliczki, Laszlo Böszörmenyi, and Jörg Keckstein. 2015. Keyframe extraction in endoscopic video. *Multimedia Tools and Applications* 74, 24 (01 Dec 2015), 11187–11206.

[11] Sorina Smeureanu and Radu Tudor Ionescu. 2018. Real-time deep learning method for abandoned luggage detection in video. In *2018 26th European Signal Processing Conference (EUSIPCO)*. IEEE, 1775–1779.

[12] Jonas Uhrig, Nick Schneider, Lukas Schneider, Uwe Franke, Thomas Brox, and Andreas Geiger. 2017. Sparsity invariant cnns. In *2017 international conference on 3D Vision (3DV)*. IEEE, 11–20.

[13] M. Ye, J. Shen, G. Lin, T. Xiang, L. Shao, and S. H. Hoi. 2022. Deep Learning for Person Re-Identification: A Survey and Outlook. *IEEE Transactions on Pattern Analysis amp; Machine Intelligence* 44, 06 (jun 2022), 2872–2893.

[14] Frederic Z. Zhang, Yuhui Yuan, Dylan Campbell, Zhuoyao Zhong, and Stephen Gould. 2023. Exploring Predicate Visual Context in Detecting Human–Object Interactions. In *ICCV'23*. 10411–10421.