SURVANT: an innovative semantics-based Surveillance Video Archives Investigation Assistant

Giuseppe Vella[1*], Anastasios Dimou[2] [0000-0003-2763-4217], David Gutierrez-Perez[4], Daniele Toti[5,6] [0000-0002-9668-6961], Tommaso Nicoletti[3], Ernesto La Mattina[1], Francesco Grassi[4], Andrea Ciapetti[6], Michael McElligott[4], Nauman Shahid[4], Petros Daras[2] [0000-0003-3814-6710]

[1] Engineering Ingegneria Informatica S.p.a.
Piazzale dell'Agricoltura 24, 00144 Roma, Italy
{ernesto.lamattina, giuseppe.vella}@eng.it,
[2] Centre for Research and Technology Hellas,
6th Km Charilaou-Thermi, 57001 Thermi, Greece
{dimou, daras}@iti.gr
[3] Demetrix Srl.
Via Ugo La Malfa, 28a/30 – 90146 Palermo, Italy
tommaso.nicoletti@demetrix.it
[4] United Technologies Research Centre Ireland, Ltd
Penrose Wharf Business Centre, T23 XN53, Cork City, Ireland
{grassifr, mcellim, perezd, shahidn}@rtx.com
[5] Catholic University of the Sacred Heart
Faculty of Mathematical, Physical and Natural Sciences
Via Musei 41, 25121 Brescia, Italy
daniele.toti@unicatt.it
[6] Innovation Engineering S.r.l.
via Napoleone Colajanni 4, 00191 Rome, Italy
{a.ciapetti, d.toti}@innen.it

Abstract. SURVANT is an innovative video archive investigation system that aims to drastically reduce the time required to examine large amounts of video content. It can collect the videos relevant to a specific case from heterogeneous repositories in a seamless manner. SURVANT employs Deep Learning technologies to extract inter/intra-camera video analytics, including object recognition, inter/intra-camera tracking, and activity detection. The identified entities are semantically indexed enabling search and retrieval of visual characteristics. Semantic reasoning and inference mechanisms based on visual concepts and spatio-temporal metadata allows users to identify hidden correlations and discard outliers. SURVANT offers the user a unified GIS-based search interface to unearth the required information using natural language query expressions and a plethora of filtering options. An intuitive interface with a relaxed learning curve assists the user to create specific queries and receive accurate results using advanced visual analytics tools. GDPR compliant management of personal data collected from surveillance videos is integrated in the system design.

* Corresponding author: Giuseppe Vella, Engineering Ingegneria Informatica S.p.a. Piazzale dell'Agricoltura 24, 00144 Roma, Italy, giuseppe.vella@eng.it

Keywords: Deep Learning, Inter/intra camera analytics, spatio-temporal semantic reasoning, Trajectory mining, Complex Query formulator.

1 Introduction

The ever-increasing use of video surveillance in multiple business sectors has created new challenges related to the exploitation of surveillance video archives. Performing post-event investigations in archives for large scale camera networks that may comprise multiple sites, complex camera topologies and diverse technologies requires significant human effort. Video analytics are destined to assist in this regard, but in spite of the progress achieved, advanced analytics are still at a relatively nascent stage owing to a number of challenges: (a) large scale processing requirements, (b) support diverse video content with differences in quality, format and extrinsic parameters, (c) inter-camera analytics, (d) effective data visualization and (e) a user-intuitive interface. The need for querying huge amounts of videos from multiple sources and extracting knowledge from them is creating a new demand for tools that can assist investigators to face the challenges in their line of work, streamlining the work of expert law enforcement officers and investigators by automating burdensome processes. Hence an improved situational awareness and search assistance tools to further diminish the possibility of missing evidence due to the huge workload is needed.

The SURVANT system assists investigators to search efficiently and effectively in video archives to contribute towards fighting crime and illicit activities, improving the sense and essence of security for the citizens. Enhancing safety and security, reduction of loss/theft, vandalism prevention, harassment prevention and regulatory compliance are among the main driving applications of surveillance systems. SURVANT provides solutions beyond the industry state-of-the-art to face the challenges identified in the markets targeted.

SURVANT supports video archive investigations via the following elements:

- A modular and scalable system based on microservices and dockerized modules.
- SotA video analysis techniques based on Deep Learning to analyze footages, maintaining an optimal balance between speed and accuracy.
- An inference framework, based on automated reasoning mechanisms, able to combine low-level information and semantic annotations to discover high-level security events and/or investigative hypotheses.
- A large-scale efficient video and image indexing for high-dimensional features and semantically-mapped content (event type and attributes).
- A GIS-based user interface allowing the user to perform complex queries regarding objects and events taking into account time, location and their interconnection.
- Semantics-based query expansion techniques to improve the precision of the search results.
- A privacy-by-design framework for automated video analysis and analytics, compliant with the European legal framework for privacy and data protection. The remainder of the paper is organized as follows. The SURVANT architecture is presented in Section 2. Section 3 describes the use-cases used during the system development and

experimental. Results are presented in Section 4. Final conclusions complete the paper in Section 5.

2 SURVANT in a nutshell

The architecture of SURVANT is a microservice architecture, devised to be modular by nature so that each module is low in coupling and high in cohesion. Each module does one job and communicates with the others to orchestrate all the operations in the best possible way. The *Client layer* contains all the parts of the front-end composed by the dashboard and the main views for operators (investigators and chefs), a control panel for system maintenance and an access point for "superusers".

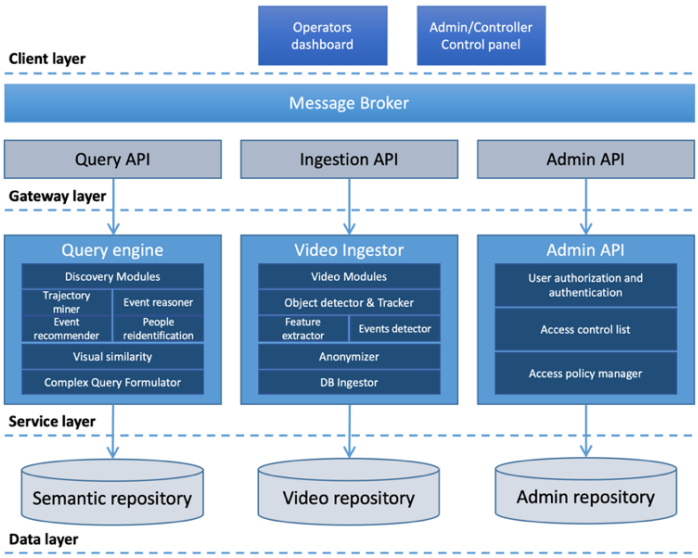

Fig. 1. SURVANT Architecture

The UI interacts with the modules within the *Service layer* through the gateway layer that contains a *Message Broker* that dispatches requests to the proper module asynchronously by means of the respective API and validate them against the User Authentication Authority Server (UAAS) and the Microservices *Access Control List* (MACL). UAAS manages the authorization and the authentication of the users on the portal, the MACL manages the authorization of the gateway in relationship with each registered microservice. The *Access Policy Manager* handles the access to all the resources applying a set of pre-defined polices. The *Service Layer* is devoted to computations. It contains the business logic of the core modules like the video modules for the ingestion of footages and the extraction of the events, objects and video features according to which the videos will be anonymized. Moreover, this layer contains the *Query Engine* modules that are dedicated to the querying of all the objects extracted and to enrich and combine them through a geographical, temporal and semantic analysis. The *Data Layer*

is the persistence layer of the SURVANT platform, it hosts multifarious storage systems according to each module's needs: a *Video Repository*, an *Admin repository* and an a *Semantic Repository* for storing RDF triples.

2.1 The SURVANT platform

The SURVANT platform allows the interaction between users and investigation's data and details, such as video footages upload, cameras involved management, notes creation, areas definition, usable semantic relations building, temporal and geo-localized results. The GUI is made up with a web portal that supports multiple users in multiple languages in a collaborative environment with a front-end user interface that allows a user to navigate, create and share information on investigations. The platform is composed of several components that includes the GUI, the different API for the ingestion, the querying and the administration and the gateway to dispatch requests from the different services (see **Fig. 1**).

The **Administration APIs** give the possibility to an administrator to set up for each authenticated user the authorization rights managed through the **User Authentication and Authorization** module to all the resources including the repositories, the cameras, the investigations, the objects and the events detected in the footages. The objects that can be detected are the following: Person, Handbag, Backpack, Suitcase, Car, Truck, Bus, Bicycle, Motorbike, Cell Phone. For what concerns the possible events that can be detected by the Visual Analytics Components and by the Event reasoner are of three categories: Low-level events (e.g. walking, standing), Group events (e.g. Fighting, chasing), Middle-level events (e.g. Entering a vehicle, e.g. Holding or picking an object). Every resource managed through the GUI has three possible sensitivity levels, i.e. a level of visibility that combined with the three access levels allow the users a granular access to them. The **GUI** allows user to seamlessly interact with underlying services through several macro-phases, that we could summarize as follow: i) Upload footages, ii) create an investigation, iii) collect notes on an investigation, iv) querying on an investigation. Each macro-phase is regulated by a workflow managed by the SURVANT platform that will ensure that every step of the phase is concluded successfully. These workflows imply the interaction between the Operators Dashboard of the Client layer, the Gateway, through the Message broker, listens to the invoked services in the *Service Layer*: the *Query Engine* and the *Video Ingestor* modules.

2.2 Video analysis for object tracking and event detecting

The video analysis module of SURVANT is responsible for extracting video analytics that enable content-based retrieval functionalities. The aim of this module is to correctly detect the classes that are useful for users, including Law Enforcement Agencies. Furthermore, it aims to track the objects of interest in order to check which route they followed before/after an event happened. Aim of tracking is to keep the same track identity for every object involved in a security event. Regarding the object detection part, the PVANet [1] architecture has been selected as a baseline. This detection framework follows the common pipeline of CNN (Convolutional Neural Network) feature

extraction followed by region proposal and region of interest (RoI) classification. SURVANT improves the baseline by redesigning the feature extraction and region proposal to improve efficiency and performance. Please find more details and results in [2]. Additionally, real-world surveillance data have been collected and annotated. Fine-tuning the object detection network with this data has dramatically improved performance. A "tracking-by-detection" paradigm is used for object tracking, where detections from consecutive frames are connected temporally. As input, the detection results are produced by an object detector. Given a new frame, the tracker associates the already tracked targets and the newly detected objects.

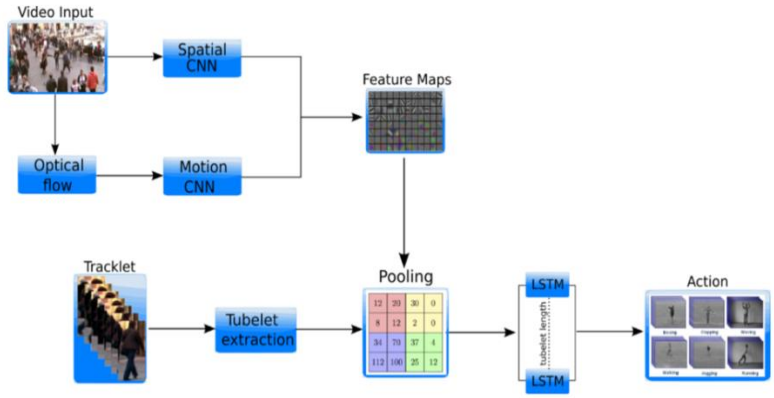

Fig. 2. Pipeline of the proposed scheme.

Object association is performed in multiple steps. The frame to frame association is performed using a smart combination of simple methods (intersection over union and HSV histogram similarity) and LSTM networks modelling the evolution of targets using their appearance, volume, position, velocity and interaction with nearby objects [3]. Tracking results can be improved with tracklet post processing.

Regarding event recognition, a tubelet-based approach was developed. The term tubelet refers to a short sequence of bounding boxes marking a person or an object of interest. These tubelets can be considered as the structural elements of the various actions. The adopted methodology is illustrated in **Fig. 2**. A two-streams approach is employed for feature extraction. The spatial stream takes as input video RGB frames and captures the appearance of the person as well as cues from the scene, while the second stream, the motion CNN, operates on optical flow input and captures the movement of the person. The event detection module uses an LSTM network to classify each of the extracted tubelets to a specific event.

2.3 The Complex Query Formulator

In a typical investigation, based on a large video collection, the investigator has to search for evidence of criminal activities that may have happened by multiple persons, in multiple locations and time points. To support the search for evidence, SURVANT

provides a set of tools for processing and analyzing the raw content, as well as visualizing and gathering evidences.

The videos are analyzed in an offline process, after which the investigator is able to search through the analyzed results using several types of queries. Such types can be text, an image crop, an event, a location, a time point, etc. Most times, however, a more complex type of search is needed for speeding up the investigation time, especially in the first critical hours of the investigation. The Complex Query Formulator (CQF) allows the formation of queries of higher complexity by combining simple query types, identifying the necessary services that need to be queried and interrogating these services to retrieve the relevant information. The main investigation sequences to which the CQF participates are: "Search by Image", "Search by Sentence" and "Geographical Analysis" (see a visual detail in **Fig. 3**).

Search by Image: The user can obtain investigation results by searching for similarities among the videos related to an investigation. He can select a particular frame of the footage and search for similar people or objects or he can upload an image that he got from other sources and search the object in the video repository.

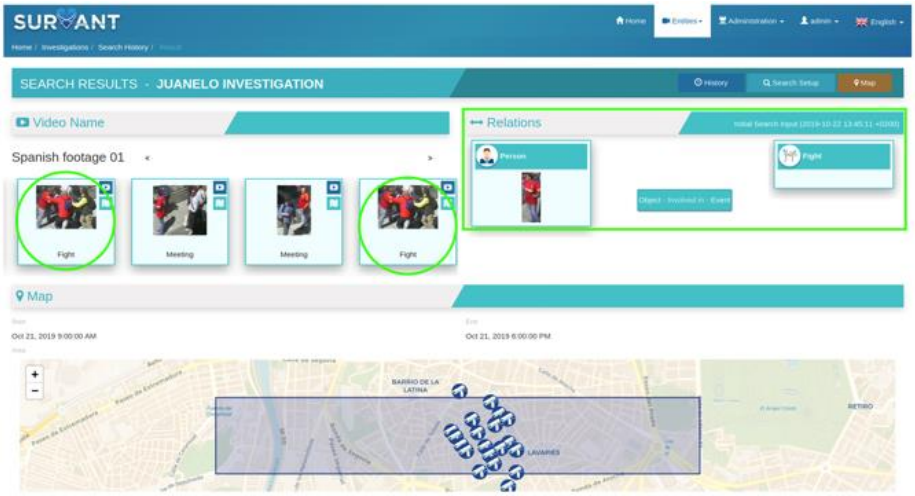

Fig. 3. A detailed view of the setup Complex query

Search by Sentence: The investigator can perform an investigation using SURVANT's complex query formulator feature and navigate footages through the bundle of resulting high level events. The user can create through a drag and drop mechanism a list of relations sentence-like composed of objects, events, geographical and temporal coordinates.

Geographical Analysis: Starting from the previous mentioned types of search, objects, suspects, victims or other persons of interest are detected. The investigator can then request the trajectories of the annotated individual(s) and may request to repeat the analysis based on new or refined queries.

2.4 The indexer: visual similarity and people re-identification

When human investigators are tasked with analyzing large bodies of surveillance camera footage in an effort to identify possible sightings of a suspect, it proves to be highly demanding in terms of time and human resources. Locating possible sightings of a suspect based on similarity to a target image was a key target for SURVANT to automate. The Indexer module takes the bounding boxes and object types reported by the Object Detector for each video frame and constructs a feature vector for the area of frame within the bounding box co-ordinates. This vector is akin to a fingerprint and similar images should have a similar fingerprint. The Indexer stores the feature vector along with metadata (such as frame number, bounding box, time, etc.) in the Indexer Repository. Given a target image, the Indexer constructs a feature vector and compares to the features stored in the Indexer Repository – ranking the results in order of similarity.

Fig. 4 describes the person re-identification pipeline. A feature vector is extracted from a query image of a person of interest (blue part at the top of the figure). Next, this vector is compared against catalogued vectors, which are previously computed offline (green part at the bottom of the figure). Finally, a ranked list of identities is obtained by ordering the images by similarity measure.

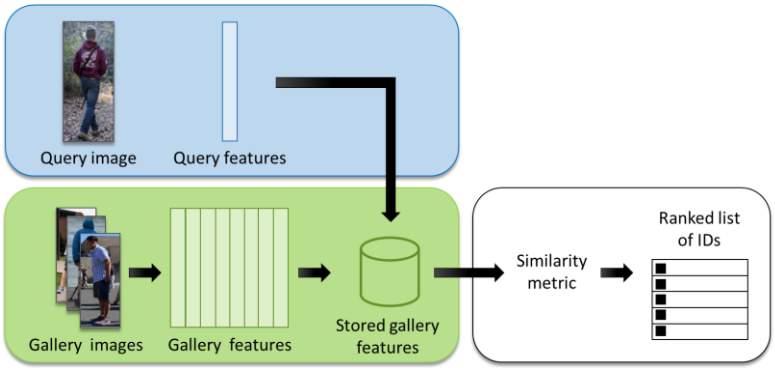

Fig. 4. Person re-identification pipeline

From an architectural point of view, the *Indexer* is composed of three submodules, namely, the *Feature Extractor*, the *Database Ingestor*, and the *Similarity Ranker*. These submodules can be identified in **Fig. 4**. The *Feature Extractor* performs the computationally intensive task of reading the output of the *Object Detector & Tracker* module and generating feature vectors for each detected object. The *Database Ingestor* stores the feature vectors into a database and provides an internal API for performing queries. The *Similarity Ranker* provides the *Indexer*'s general-purpose, high-performance REST API that is used by the *Complex Query Formulator* as well as provides extensibility capabilities for future modules. The video ingestion pipeline is comprised of a workflow which co-ordinates activity across object detection, feature vector cataloguing, reasoner analysis and anonymization (via targeted blurring) according to object

classification and sensitivity level to fine tune the anonymization required according to the privilege level of the person viewing the footage.

The Feature Extractor received the majority of the focus within the Indexer module as its efficiency and quality in constructing a feature vector drives the overall usability of the module. The feature representations are extracted from a Convolutional Neural Network considering the identities as classes and taking the output from the last layer before the softmax layer as the deep features. After initially analyzing an approach using ResNet [4], we subsequently focused our attention on MobileNets [5] due to its improved efficiency. While MobileNets offer an alternative smaller and more efficient network since its feature extraction time is lower, we also want to maximize the performance of a small network to be as accurate as possible. For this purpose, we leveraged network distillation [6] using ResNet-50 as a large model that can act as teacher and MobileNets as a suitable architecture for playing the role of the student network.We further sought to improve the performance of the pipeline in person re-identification in terms of computational cost at test time. Once the deep features have been extracted and assembled into a feature vector then they are stored in the Indexer Repository. To compute similarity between feature vectors we use Euclidian distance – it is simple and can be computed in-situ within the database allowing significant speed improvements over an approach which retrieves all potentially relevant data and computes the similarity externally.

An analysis of different techniques for person re-identification was performed prior to choosing the implemented solution. The considered algorithms can be classified as classical (hand-crafted) and deep learning-based methods. The latter ones outperform significantly the former ones, but with the disadvantage of requiring dedicated hardware (i.e. GPUs), and large amounts of data and time for training. The classical methods evaluated included Local Maximal Occurrence Representation (LOMO) and Cross-view Quadratic Discriminant Analysis (XQDA). As for deep learning-based methods, ResNet [4] and MobileNet [5] were tested, as well as several combinations of both by applying network distillation techniques [6] and swapping the teacher and student roles. For evaluation, Market-1501 [7] and DukeMTMC-reID [8] were used. To summarize the results, classical methods provided much lower accuracy than deep learning-based methods, thus they would be only applicable where there are strong hardware limitations. ResNet-50 and MobileNets showed a similar performance with some differences, while MobileNets described better the features for the Market-1501 dataset, ResNet-50 performed better for the and DukeMTMC-reID dataset.

Market-1501	Rank-1 (%)	mAP (%)	GPU time (ms)
LOMO + XQDA	43.32	22.01	17.25
ResNet-50	64.46	38.95	7.82
MobileNet 1.0 independent	67.37	39.54	1.84
MobileNet 0.25 independent	59.74	34.13	1.63
MobileNet 0.25 distilled from ResNet-50	71.29	45.76	1.63
MobileNet 0.25 distilled from MobileNet 1.0	70.46	45.24	1.63

DukeMTMC-reID	Rank-1 (%)	mAP (%)	GPU time (ms)
LOMO + XQDA [53]	30.75	17.04	17.25
ResNet-50	67.1	44.59	7.82
MobileNet 1.0 independent	57.41	34.86	1.84
MobileNet 0.25 independent	49.69	28.67	1.63
MobileNet 0.25 distilled from ResNet-50	64.99	42.32	1.63
MobileNet 0.25 distilled from MobileNet 1.0	59.69	38.48	1.63

2.5 The trajectory miner

In the field of video-surveillance analysis, the problem of tracking a person of interest given a ground truth image (query), in a large crowded spatiotemporal region is known as "Trajectory Mining" [9]. Several algorithmic challenges have been arising due to absence of information about 1) the length of the true trajectory, 2) the identities of various persons (except for the query), and 3) starting/end point or the position of query in the true trajectory. In particular, the scenario of interest involves m persons moving in a spatial zone and being detected by cameras. Every detection is represented as a tuple (f_i, t_i, s_i), where f_i is the i^{th} frame/detection, t_i is the timestamp and $s_i = (x_i, y_i)$ are the spatial coordinates of the detected person or the camera. This problem via a graph signal processing approach has been tackled, modeling the tuples (f_i, t_i, s_i) as node signals and edge weights as a fusion of visual, spatial and temporal information. The information fusion algorithm can be summarized as follows: **Step 1: Visual Graph**. For every frame f_i a "deep visual signature" $\xi \in \mathbb{R}^p$ from the ResNet50 [4] architecture has been extracted, with dimensionality $p = 2048$. A k_{nn} similarity graph G_ξ [10], where k_{nn} is the number of neighbors, between the features has been obtained using the UMAP (Universal Manifold Approximation and Projection) graph construction algorithm [11]. **Step 2: Spatiotemporal Graph.** k_{nn}-nearest spatial neighbour graph G_s is constructed between the coordinates $s_i = (x_i, y_i)$ using Haversine distance. Denoting $d_{i,j}$ the Haversine distance between coordinates (s_i, s_j), temporal information is fused in the spatial graph using:

$$W_{st}(i,j) = \frac{1}{1+\exp\left(-10\left(\frac{|t_i - t_j|}{d_{i,j}} - 0.4\right)\right)} W_s(i,j). \quad (1)$$

Assuming an average walking speed of 1.4 m/s [12], the above spatiotemporal weighting scheme allows SURVANT system to decrease the spatial similarity $W_s(i,j)$ between coordinates (s_i, s_j), whose distance cannot be physically traveled in time $|t_i - t_j|$. **Step 3: Fusion of visual and spatial graphs.** To merge the three sources of information into one graph $G = (\mathcal{V}, \mathcal{E}, W)$ the module computes the Hadamard product of the two adjacency matrices $W = W_\xi \circ W_{st}$. The matrix W is still a valid similarity matrix and its entries represent the spatiotemporal similarity in the fused feature space between the tuples (f_i, f_j). **Step 4: Conversion to directed fused graph**. Since a trajectory can only move forward in time, a direction is assigned to each edge such

that $e = (v_i, v_j) \notin \mathcal{E}$ if $t_i - t_j > 0$. It is easy to show that the obtained graph G is a directed acyclic graph (DAG). The described algorithm filters out several edges which might lead to false paths, especially those related to poor quality of visual features. In a crowded environment the actual image of a person might be occluded by other people, shadows, objects etc. Thus, the refinement via spatial graph, which is accurate due to the exact knowledge of the spatial coordinates, helps to remove the false edges in the final graph. Once the graph G is constructed, next step is to query the graph for getting the trajectory relevant to a suspect. The investigator uploads the image of a suspect along with its spatial coordinates and timestamp. Let us call this the 'query tuple'. This query tuple after extracting the visual features of the image is represented as:

$$Q = (x_q, t_q, (sx_q, sy_q)),$$

where x_q is the image, t_q denotes the timestamp and (sx_q, sy_q) denote the spatial coordinates of the query q. The goal of the trajectory mining algorithm is to produce a trajectory which involves query tuple. The graph G is a DAG, thus a weighted longest path algorithm can be used to get the trajectory which maximizes the similarity starting from node Q [13] Since the query node is not necessarily the beginning of the trajectory and the end point of the trajectory is not known, the algorithm was run twice, first using Q as starting node, and then reversing the direction of the edges and running the algorithm backwards. Finally, the forward and backward trajectories to obtain the full trajectory have been concatenated. As remarked before, poor visual features quality may affect the knowledge graph construction steps, creating false edges in the graph. In the attempt of maximizing the maximum weighted length, the longest path algorithm may concatenate trajectories which belong to different persons and return a very long trajectory. In order to mitigate this problem, a modification of the algorithm has been proposed so that it maximizes the sum of the edge weights in the path divided by its length. Denoting $p = \{e_1, e_2, \dots, e_{L_p}\}$ the forward path starting from query node Q, the algorithm solves the following optimization problem:

$$\arg\max_p \sum_{e \in p} \frac{w_e}{L_p}$$

where w_e is the weight associated to edge e.

2.6 The reasoner and the semantic repository

The event reasoning module is based on an inferential approach to detect high level events using Semantic Web Rules (SWRL). This approach combines all the results collected by the lower level detectors, as entities, such as people and objects, and base events, such as persons running or walking, and their spatial trajectories over time, provided by the trajectory mining algorithm, with a rule based approach to identify potential crimes or high level events, such as pick-pocketing or vandalism. The low-level events and the spatial and temporal information collected from different cameras – including the computed trajectories – are indexed in an optimized semantic datastore, where rules for detecting events are manually defined using SWRL. When these rules are applied to the indexed information, high-level events can be detected. The approach

used is based on the standardization effort around the SWRL rule-based language. SWRL is a Semantic Web Rule Language based on a combination of the OWL DL and OWL Lite sub-languages of the OWL Web Ontology Language with the Unary/Binary Datalog RuleML sub-languages of the Rule Mark-up Language. SWRL includes a high-level abstract syntax for rules in both OWL DL and OWL Lite formalisms. A model-theoretic semantics is given to provide the formal meaning for OWL ontologies including rules written in this syntax. This approach has been successfully used in other domains as well, including law and biomedicine [14],[15],[16].

Deduction of the existence of the high-level events is computed using reasoning-based strategies, based on empirical rules, suggested and confirmed by the Madrid Municipal Police involved in the project. The proposed approach is able to recognize crime events like pick-pocketing attempts or street fights in a repository containing low- and middle-level metadata extracted from surveillance videos. The rules and the individuals are evaluated by a semantic reasoner, which infers suspicious events and persons, which can be further inspected by a human officer after the automatic analysis. To define the rules, the output of the Computer Vision algorithms and detectors has been considered as a base. These algorithms analyze the scene and extract information which is stored and indexed in a semantic repository in the ingestion phase, performed after an investigation has been started on the system. The extracted information includes objects like Bus, Car, Motorbike, Handbag and Person. Each object holds information relevant to the detection task, mainly the start and end frame of the object appearance and a collection of statuses for each frame in which it appears. Each frame contains the frame number (for temporal reasoning), position and size of the object (for spatial reasoning), and the low-level action the object is performing in that moment. Some of the low-level detected actions are Standing, Walking and Running for persons, Moving and Stopping for vehicles, etc. In addition, middle-level events, such as Entering or Exiting buildings, Falling or Lying down, Fighting and Graffiti Making are also correctly detected by the analyzers.

The description of objects and events is encoded in a RDF-based ontology, implemented specifically for the SURVANT system, in which each individual can be expressed in RDF-based triple format. In the ingestion phase, the data indexed in the semantic repository can easily reach billions of tuples. One of the implemented features is the ability to summarize the raw temporal data, taking into account only the start and end time of events that can be meaningful for the inference performed by the SWRL rules created manually by the users. In addition, the spatial data of each person can be filtered and summarized, taking into account the position and distance of groups of people that are near enough to potentially interact between each other. The advanced search functionalities provided by the reasoner are able to offer the investigators not only a concise and effective representation of the events detected inside the scene of a crime, but also more high-level abstractions. In particular, if a criminal event, such as a theft or pickpocketing, is reported in one of the roads covered by the video surveillance system, the system should be able to provide investigators with potential "suspicious" situations, combining appropriately several simple basic events. Starting from the position data, and from mid-level events such as "Walking", "Running", "Meeting", high-level events as "potential pickpocketing" or "probable fight" can be recognized

by applying rules based on spatial, temporal and empirical criteria defined in behavioral patterns for improving the capability to identify suspicious actions. An example of one of the event “Robbery” is presented in **Fig. 5**.

"If a person has luggage, another person comes close to him, grabs his luggage, and starts running, and the owner of the luggage runs after him, then there has been a robbery."

Fig. 5. Example of a “Robbery” event as captured by a surveillance camera (images degraded to preserve the privacy of the actors).

One of the most challenging tasks of the event reasoning module has been to find a way to overcome the inherent scalability problems of the reasoning systems, with an innovative approach based on highly parallelized computing. The adoption of the Stardog repository [17] (with embedded reasoning capabilities), the commercial version of Clarks & Parsia Pellet reasoner [18] (previously used), allowed introducing several enhancements in this direction; in particular, the ability to use an in-memory storage for heavy computational scenarios with huge amounts of triples present and the introduction of a high availability and performance clustering, based on Zookeeper and other HA techniques. Stardog is a commercial semantic repository that supports the RDF graph data model, the SPARQL query language, the property graph model and the Gremlin graph traversal language, as well as OWL 2 and user-defined rules for inference and data analytics, distributed by Complexible, an innovative and relatively recent USA company, focused on inference solutions. Stardog reasoning is based on the OWL 2 Direct Semantics Entailment Regime. Stardog performs reasoning in a lazy and late-binding fashion: it does not materialize inferences; but, rather, reasoning is performed at query time according to a given reasoning level. This allows for maximum flexibility while maintaining excellent performance and scalability. After the start of each investigation, when the underlying analyzers have completed the detection process of objects and low-level events, the Reasoner module is invoked by the Vision Service, which signals the presence of data to be ingested in the semantic repository. The reasoner responds immediately with a simple ACK and starts loading the data from a queue, in a typical producer/consumer modality. After having performed deductions and logical consequences from low-level events with the defined set of SWRL rules, the module is directly involved in the SURVANT project pipeline to perform queries on the underlying knowledge base in order to present the aggregated computed results to the investigators in the GUI of the system (see **Fig. 6.**)

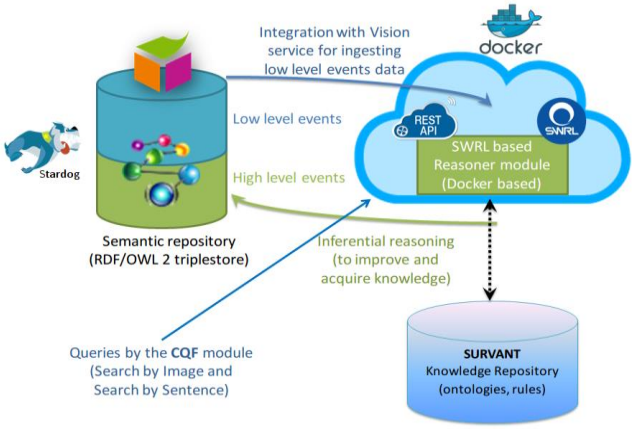

Fig. 6. Interactions between the semantic repository and SURVANT modules.

3 Use cases

The use cases that have been defined for the SURVANT product are the following: i) Aggression (Beat and run away), ii) Theft (Pick pocketing), iii) Vandalism (against parked vehicles, Defacing of buildings, iv) Scene Monitoring (Building monitoring), v) Missing Person (Vulnerable individual reported missing), vi) People Tracking (Assault on a Person, Person of Interest Tracking, Detect subsequent criminal behavior). To test the above use cases the Municipal Police of Madrid has recorded 5 days of videos with actors that followed specific plots of some possible scenarios that the consortium has defined for the testing and validation of those use cases. The SURVANT algorithms have been tested against almost 100GB of footages.

4 Results

4.1 Video analysis for object tracking and event detecting

Object tracking is performed in SURVANT using the track-by-detection paradigm. For object detection, a modified PVANet model is used that is faster by 15% and smaller by 79%, producing comparable results to the original PVANet under the same training procedure. However, the performance was subpar for surveillance footage.

A new dataset was created to cover the needs of the use cases described above, involving public and challenging real surveillance data. The dataset has almost 430K samples from the following classes: person, handbag, backpack, suitcase, car, truck, bus, motorbike, bicycle, cell phone, laptop, and graffiti. The object detection network was re-trained using the dataset and some augmentation techniques to better simulate the real situation introducing motion blur.

Fig. 7. Example detections on a MET CCTV video by the default PVANet.

In order to produce trajectories of objects, SURVANT utilized a DL-based detection association model using multiple information cues including appearance (see **Fig. 7**), position, interaction, volume and velocity. The network was able to give predictions for batches of images with speed that approximates 1522 FPS and precision of 90%. Moreover, SURVANT utilized a tracklet association model that was able to identify and connect tracklets of the same person in real time with an accuracy of 91%.

Regarding the performance of the event recognition module, a collection of public datasets and surveillance videos were collected and annotated with events of interest for our use cases. The network described in Section 3.1 was trained and tested on separate parts of the dataset and the results are provided in **Table 1**.

Table 1. Recognition accuracy for different event classes.

Overall Accuracy	Average Accuracy	Stand	Walk	Run	Fight	Make Graffiti	Lie down	Enter building	Get in vehicle
89.37	88.37	77.33	95.06	72.73	81.48	94.12	95.89	95.7	94.68

4.2 Event Reasoning

The experiments on the event reasoning in SURVANT have been carried out using the output of the low-level detectors and of the trajectory miner module. This information was passed to the Stardog-based semantic repository component, where triples were stored and reasoned with. The visual analysis tracks objects frame by frame and identifies low- and middle-level events and actions that each object is performing. This information is then passed to the semantic component, where information is trans-formed to the form of Subject-Predicate-Object triples and stored in the repository. After a video is processed, the reasoning is performed on the stored triples in the datastore and new relations are inferred, containing the detected abnormal events. The performed tests show encouraging results in the recognition of potential suspicious events: on one hand, the system produces some "false positives" in the recognition, but on the other hand the lack of recognition of potentially suspicious events and persons is very rare. The tests have been performed on a relatively small set of surveillance videos (almost 100GB of files), provided by the Madrid Police; the statistical performance of the reasoner in detecting crimes in real world situations is very encouraging. Often in these videos the crime scene is positioned at the extreme boundaries of the video or even

outside of the camera coverage area. In this case it was not feasible for the reasoner to detect the crime correctly. By taking into account this factor and the relatively low quality of the videos recorded and analyzed by the system, a success percentage of 2 detections every 3 crimes (67%) is to be considered very good, as well as the percentage of "false positives", which is around 15%. For each high-level crime type, some rules used by the reasoner perform very well indeed and can be considered already optimized, whereas some other rules are less precise in the detection phase. This may lead towards future refinements of the less performing rules.

4.3 Trajectory miner

Trajectories of people walking in Cork City, Ireland were simulated as an initial test of the algorithm. The topological graph is obtained from OpenStreetMap XML data [19] and uses $m = 75$ people of the test portion of the publicly available MARS dataset [20] as camera detections. Topological graph consists of $N = 4000$ nodes, representing straight walking roads, which are connected if the corresponding roads are connected. $N_c = 300$ nodes equipped with simulated cameras have been selected. For each person in the dataset, source and destination nodes and a random path between them have been selected. Timestamps are obtained by computing Haversine distance between the nodes and assuming the walking speed to be a Gaussian random variable with mean 1.4 m/s and standard deviation 0.1. Whenever a person, during his trajectory, crosses a camera node, a frame f_i from the corresponding image dataset is used to simulate a detection. The list of the crossed camera nodes is our target trajectory.

As a performance metric, the F-score on the detected nodes was computed in the estimated trajectory as compared to the nodes in the ground truth trajectory. A comparison of our results with simple k_{nn} nearest neighbor search on the visual features as a baseline, with k_{nn}=15 to 100 and a step of 10 has been performed. For most of the cases the performance of our algorithm outperforms k_{nn} algorithms, up to 350% over k_{nn}-15 baseline for class 37. Moreover, for the proposed approach, the standard deviation is much lower as compared to the other algorithms, mainly because of the false positives avoided due to the spatiotemporal refinement of the visual similarity.

5 Conclusions

The SURVANT product that is the main output of the SURVANT project is a fully-customizable, scalable and robust system that is ready to hit the market. SURVANT presents an architecture that is horizontally and vertically scalable. The reliability of its output depends on the positive results that have been produced by SURVANT from a scientific point of view. The adopted CNN was able to give predictions with a precision of 90%. SURVANT utilized a tracklet association model that was able to identify and connect tracklets of the same person in real time with an accuracy of 91%. From the semantic reasoner perspective SURVANT got a success percentage of 2 detections every 3 crimes (67%), as well as a false positive rate around 15%. Thanks to the approach adopted with the trajectory miner, false positives were avoided due to the

spatiotemporal refinement of the visual similarity. This work has been supported by the SURVANT project that received funding from the EU Horizon 2020 Fast Track to Innovation (FTI) programme under Grant Agreement No n° 720417.

References

1. K.H. Kim, et al. “Pvanet: Deep but lightweight neural networks for real-time object detection.” arXiv preprint arXiv:1608.08021, 2016
2. D. Anastasios, et al. “Multi-target detection in CCTV footage for tracking applications using deep learning techniques.” *2016 IEEE Int. Conf. on Image Processing (ICIP)*, 2016
3. E.A. Ahmed. An improved deep learning architecture for person re-identification. *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, p. 3908-3916, 2015
4. K. He, et al. “Deep residual learning for image recognition”. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, p. 770-778, 2016
5. A.G. Howard, et al. “Mobilenets: Efficient convolutional neural networks for mobile vision applications”. arXiv preprint arXiv:1704.04861, 2017
6. G. Hinton, O. Vinyals, and J. Dean, “Distilling the knowledge in a neural network”. arXiv preprint arXiv:1503.02531, 2015
7. L. Zheng, et al. “Scalable person re-identification: A benchmark”. *IEEE International Conference on Computer Vision*, p. 1116-1124, 2015
8. E. Ristani, et al. “Performance measures and a data set for multi-target, multi-camera tracking". *Eur. Conf. on Comp. Vision, w. on Benchmarking Multi-Target Tracking*, 2016
9. Y. Zheng, “Trajectory data mining: an overview”. *ACM Transactions on Intelligent Systems and Technology (TIST)* 6.3, p. 1-41, 2015
10. D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, “The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains.” *IEEE signal processing magazine*, 30.3, p. 83-98, 2013
11. L. McInnes, J. Healy, and J. Melville, “Umap: Uniform manifold approximation and projection for dimension reduction”. arXiv preprint arXiv:1802.03426, 2018
12. R.W. Bohannon, “Comfortable and maximum walking speed of adults aged 20-79 years: reference values and determinants”, *Age and ageing*, 26, p. 15–19, 1997
13. R. Sedgewick, Algorithms, Pearson Education India, 2016
14. N. Capuano, A. Longhi, S. Salerno, and D. Toti. “Ontology-driven generation of training paths in the legal domain”, *International Journal of Emerging Technologies in Learning (iJET)*, 10, p. 14-22, 2015, DOI: 10.3991/ijet.v10i7.4609
15. G. Arosio, G. Bagnara, N. Capuano, E. Fersini, and D. Toti. “Ontology-driven Data Acquisition: Intelligent Support to Legal ODR Systems”, *Frontiers in Artificial Intelligence and Applications*, 259, p. 25-28, 2013, DOI: 10.3233/978-1-61499-359-9-25
16. D. Toti. “AQUEOS: A system for question answering over semantic data”, *Proceedings - 2014 International Conference on Intelligent Networking and Collaborative Systems, IEEE INCoS 2014*, p. 716-719, DOI: 10.1109/INCoS.2014.13
17. Stardog. https://www.stardog.com/
18. Clark & Parsia Pellet reasoner. https://www.w3.org/2001/sw/wiki/Pellet
19. OpenStreetMap contributors, “Planet dump retrieved from https://planet.osm.org”, https://www.openstreetmap.org
20. L. Zheng, et al. “Mars: A video benchmark for large-scale person re-identification”. *European Conference on Computer Vision*, p. 868-884, 2016