An Event-Driven Approach to the Recognition Problem in Video Surveillance System Development

Nikita Bazhenov, Egor Rybin, Dmitry Korzun
Petrozavodsk State University (PetrSU)
Petrozavodsk, Russia
{bazhenov, rybin}@cs.petrsu.ru, dkorzun@cs.karelia.ru

***Abstract*—Many video surveillance systems (VSS) have been already developed for various application domains. Such systems are based on well-elaborated recognition algorithms of Artificial Intelligence (AI) and implemented as Ambient Intelligence (AmI) services in Internet of Things (IoT) environments. In particular, algorithms support such smart VSS functions of video data processing as human detection, human identification, object location within an image, human activity recognition. Many software tools have been developed to implement various recognition algorithms for VSS development. In this paper, we consider the following VSS development problems: a) a generic model of events in video data for a given problem domain, b) a hardware–software architecture for data processing with existing recognition algorithms, and c) a model to construct a required smart VSS function using existing software tools. We introduce our event-oriented approach to solve the above VSS development problems. The approach is experienced in several use cases. Our experimental study shows the applicability of the proposed approach in terms of the accuracy and performance.**

I. Introduction

Many smart video surveillance systems (VSS) have been already developed for various application domains, e.g., see public safety [1], cities and homes [2], [3], healthcare [4], industrial production [5], agriculture [6]. The VSS development is still progressing, when for each application problem specified recognition is required in given video data. Recognition algorithms come from Artificial Intelligence (AI); they need to be implemented as Ambient Intelligence (AmI) services in Internet of Things (IoT) environments [7]. In particular, the computer vision technology includes recognition algorithms to implement smart VSS functions of video data processing [8], [9] as human detection (as a physical entity), human identification (as a particular person, e.g., by face), object location within an image (e.g., in relation with other objects), human activity recognition (e.g., her/his pose detection and tracking).

The studied VSS functions belong to the progressing area of smart sensorics in IoT environments [10]. A sensing system is typically at Internet edges, e.g., sensors are embedded to physical human surroundings or integrated to human wearables. A VSS function can be considered as digital extension of human sensorics (human sensory organs). This type of technology is in demand in development of bionic suits and other assistive environments for robotics, smart textile, and Tactile Internet [11], [12].

The required recognition can be based on the event-driven approach for development smart IoT-applications [13]. Recognized information is derived as a set of facts that are semantically grouped into a common time-space structure (e.g., facts occurred in the same time period or related to the same physical object). This paper continues our work on semantic-oriented event-driven video data processing with essential role of edge IoT-devices [5], [14]. We consider three VSS development problems, which need novel methods for event modeling, distributed computing, and service construction.

First, recognition needs to operate with a sequence of images (frames in a video flow). Relation between images and interrelation of zones in the same images are semantics that must be taken into account in video data processing [15]. Situational video analytics provides algorithms to understand the surrounding context of a monitored object in a video surveillance scene. Recognition aims at finding typical situations and scenarios of interrelated long-term events over a certain period. A comprehensive view on a video stream is needed to prepare the input data for such recognition algorithms. We propose an event-based model of video data to solve the first development problem.

Second, video data processing is distributed [6]. VSS utilizes many IoT devices, small or large. Essential part of processing can be implemented directly at sensing devices (video cameras), so making them smart. High-complexity recognition algorithms are running on servers and other powerful computing systems (nearby or remote). We propose a microservice-based method to combine and configure devices into a data processing IoT system.

Third, many "basic" recognition algorithms are available on the software market and can be applied in VSS development [16], e.g., libraries with AI primitives. Various implementations of AI algorithms are accumulated, e.g., in Python libraries and in other components of the computer vision technology. When a particular VSS is developed, its recognition functions cannot be simply implemented using available implementations. The developer constructs a required VSS function as a composition of exiting algorithms. Furthermore, such a composition requires tuning the algorithms to the specifics of the problem domain (playing with the parameters of the composed algorithms). We propose a model to compose recognition algorithms into software implementation of a required VSS function.

In sum, the presented solutions to the three VSS development problems form a novel semantic-oriented event-driven

approach. The approach provides methods and tools that the VSS developer applies to implement required VSS functions. We tested the methods on several use cases. The experimental study shows the efficiency in terms of the accuracy and performance.

The rest of the test is organized as follows. Section II defines studied VSS development problems based on review of the existing VSS solutions. Section III introduces our methods to solve the VSS development problems. Section IV shows our software tools to implement VSS functions based on the proposed methods. Finally, Section V concludes the paper.

II. Development Problems

Let us consider existing solutions to VSS development. Basically, a recognition algorithm is event-oriented, recognizing such complicated parameters as *Who* has operated with the given object and *When*, *What* object is recognized and *Where* the object is located [1].

A. Existing Solutions

There are many approaches to the implementation of video surveillance systems [17]. Most of these systems are trying to solve one of the following tasks: detection and tracking of objects, identification of faces and human activity. Such systems are often narrowly focused and offer solutions only for a specific industry. At the same time, there are a small number of video surveillance systems that are trying to create a technological solution that includes a whole range of algorithms and methods, not only mathematical, but also related to the choice of Edge/Fog/Cloud paradigms and selecting hardware for a particular case.

Paper [18] proposes the key properties of the studied service class of how to operate with multiple data streams. Internet of Things technologies: Transforming Ordinary Urban Spaces into Cyber-Physical Spaces [19] with the next features: finding optimal IoT architectural configurations using video surveillance, drones, camera phones, analysis of heterogeneous configurations in terms of performance and results.

Machine Learning has substantially grown over a period of decade. Deep Learning the new field of Machine Learning is gaining ever-increasing interest in research due to its implicit capability appropriate for successful applications in the field of computer vision, speech processing, image processing, object detection, human and face sub-attribute detection, and many more analytical systems. Intelligent Video Systems and Video Analytics is managed by a wide collection in transportation, security, health care and customer analytics [8].

One of the most progressive approaches to the development and support of technologies is Industry 4.0. This approach is based on the massive introduction of information technology (IT) in industry sectors using modern technologies of the Internet of things and AI. The article [20] plans to use a simple random sample questionnaire and will use structural equation modeling (SEM) to analyze the data. The study shows that the factors enabling Industry 4.0 and its implementation have a significant impact on the performance of firms. The proposed model helps define the enablers of Industry 4.0 when combined with the theories of Technology, Organization and Environment (TOE), Diffusion of Innovation (DoI) and Dynamic Opportunity (DO) theories. The study also helps government, policy makers and practitioners evaluate and implement their strategy to successfully improve the performance of Industry 4.0 companies and enterprises.

Therefore, a novel technology is needed to develop VSS with “smart” video services.

- Development of a unified data model: lack of description of events related to recognition.
- Development of a method for constructing a hardware configuration of video data processing system: lack of description of the hardware configuration on which the algorithms work.
- Development of algorithms for machine learning model construction and recognition in video data: lack of a choice of parameters for the algorithm for solving a specific problem.

B. Problem 1: Unified Data Model

One example of a comprehensive video analytics solution is given in [21]. Video engine, event engine, video surveillance ontology were used as technologies, which help in making decisions on video analytics tasks based on events and the context of what is happening. The use of an event-driven approach (for example, in UML language [22]) makes it possible to use not only mathematical recognition algorithms, but also semantically process data, especially in cases where a large number of recognition objects are used (for example, several people and machines). Moreover, simple events can be transformed into more complex information structures (complex events), which allows a more detailed description of the context of what is happening [14]. However, most of these systems use only Edge [5], [14] or Cloud [21] approaches, which leads to the next problem of choosing hardware and software.

Urban environments [23] and Smart City [24] deployments typically have thousands of surveillance and public cameras. Rapid advancements in computer vision techniques due to Deep Neural Networks enable using these camera feeds for performing non-trivial analytics. The knobs lend the application the ability to scale potentially to thousands of cameras. In this proposal two representative applications were designed; missing person tracking and priority signalling for emergency vehicles. The application scales to 1000 cameras on a Cloud-only deployment of 10 Azure VMs with 8 cores and 32GB RAM each were empirically verified.

C. Problem 2: Configuration of Video Data Processing

The choice of server hardware for deploying video analytic services depends on the organization of computing for one of the computing paradigms.

- Edge computing [25] – organization of computing within the reach of end devices. Such devices can be located within the same LAN (local area network) and are

represented by both low-performance computing components (Raspberry Pi, Nvidia Jetson) and medium-power components (local server).
- Fog computing [26] – organization of computing "nearby" end devices. Usually, this level of computation is necessary for additional data collection and analysis in cases where, for example, end devices are located in neighboring rooms, but their additional filtering is required.
- Cloud computing [27] – the organization of remote computing and data storage, without the active participation of the user. Usually represented by high-performance computing devices (DPC - data center, data center), consisting of a large amount of memory and graphics processors.

Each of these paradigms has its own advantages and disadvantages. For example, edge computing allows you to organize calculations near video cameras, but usually have limited computing resources.

D. Problem 3: Algorithms for Modeling and Recognition

The problem of choosing an algorithm [2], [28] for object recognition is the most difficult task, since it depends on the detected events and the availability of minimal hardware (server) support. Moreover, these algorithms have a large number of settings and parameters, which significantly affect the performance and accuracy of calculations. It is often difficult to determine what recognition accuracy a particular neural network will have at the output without conducting experiments. Including the results obtained during the experiments may differ (usually in a negative direction) when the system goes from laboratory conditions to production.

Monolithic centralized applications are becoming the most complex in their structure. Furthermore, this often leads to poor scalability and difficulty in understanding the interactions between processes. The article [29] proposes a general framework for a microservice system for an IoT application, which is the best scalable and maintainable architecture. The system design and associated microservices are introduced, and the focus is on core services and data exchange between devices from the service layer to the physical layer. There are more options to support the interaction and placement of dissimilar objects. In addition, this structure can easily provide additional integration of applications such as automation, analytics (including video analytics), big data analysis.

III. Proposed Methods

The literature review from the previous section shows that the existing solutions are still not well effective for the introduced VSS development problems. A novel approach is needed where appropriate methods and tools support VSS development. The following R&D problems are considered in this section towards constructing the required approach.

- Development of a unified model for representing events that occur in relation to heterogeneous objects and dynamic situations.
- Development of a hardware-software architecture with requirements for video sources and server hardware.
- Development of a method for selecting basic data processing algorithms and machine learning methods to construct smart VSS functions.

A. Video Data Processing in VSS

Modern real-time video analytics systems require a large amount of computing resources. The standard approach to organizing computing is to transfer all information and video streams to one centralized server. Such transmission and processing of information can be improved using on-board computing (on the same device that generates/receives data) or on local computers.

Work [30] suggests using the JCAB algorithm, which uses many parameters in the video analytics configuration, for example: bandwidth of devices at the periphery, network variability, video content dynamics. The use of such an algorithm can effectively balance between parameters such as analytics accuracy and power consumption.

Our approach develops a given VSS using several steps for video data processing. The steps are represented in Figs. 1–4.

Concept model of *Step 0* is shown in Fig. 1. As part of the approach, we propose the following concept model for VSS. Several heterogeneous cameras transmit video streams or image streams, thereby generating events. (A camera can be static, integrated to a smartphone, managed and IP-camera, built-in camera, etc.) Then, a chain of several events in a row or simultaneous registration of events leads to the appearance of complex events (complex events can be formed into even more complex structures, for example, systematic events). Each of these groups of events can be simultaneously processed at the edge, in the fog and in the cloud.

Most likely, the complexity of events will increase with a more integrated approach and in cloud solutions. Each of these paradigms is usually represented by a low-performance device (Edge), a medium computer (Fog), and a powerful clustered server (Cloud). However, depending on the wishes of the developer, low-performance devices can be located, for example, in the cloud. Or vice versa, powerful computers can be on the periphery (in the case when it is necessary to process large amounts of data in the shortest possible time). For each computing device, configuration and parameter, user input, and additional contextual information may be further allocated.

Such data input can significantly affect the tracking of the desired events and the distribution of load between devices. On each device, methods and algorithms for the detection and recognition of various objects and entities can be applied, the requirements for such methods also increase depending on the computing resources. As an output, various events, processed images, graphs and diagrams can be detected. However, this kind of conceptual model only suggests possible processing of the data. Within such a model, it is possible, for example, to exclude completely complex events and leave only edge computations.

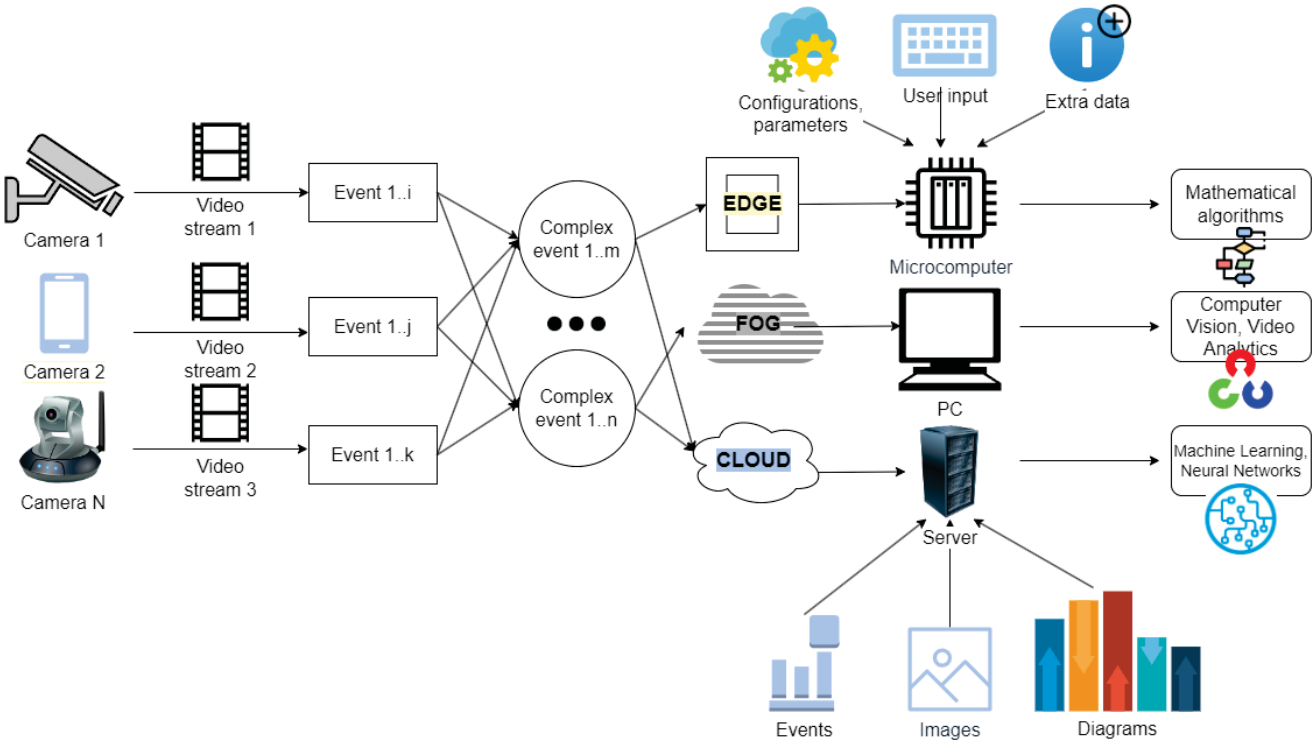

Fig. 1. Step 0. Concept model of proposed technology

B. Unified Event-Driven Model

Event representation model for *Step 1* is shown in Fig. 4. At this step, you need to describe the events that will be "tracked" based on the video streams received from the surveillance cameras. The following rules are proposed as basic ones: each video camera generates an individual video stream, which may contain many key elements that make up a basic event. At the request of the developer, the base event can change depending on time, location, and context. The number of basic events from one video stream can be unlimited.

Several video streams contain many basic events. Several basic events form a complex event (based on $1, 2, \ldots, n$ video streams). A complex event can include several simple consecutive or simultaneous basic events. Also, basic events can be independent and observed in different periods of time with different durations, but constitute one complex event. Complex events represent the total result, which can be presented in the form of a graph, diagram, text, or combined visual model. The result should be understandable and representative of the user.

C. Architecture of Video Data Processing System

Architecture with computing components for *Step 2* is shown in Fig. 3. This technology step can consist of all functioning devices, necessary computing modules and services, as well as actions and operations for interaction between components. To build such model, it is necessary to adhere on of the computational paradigms. The hardware-software architecture of the interaction of computing components within the framework can be represented as the next following paradigms: Edge (different types of cameras: Raspberry Pi, smartphone), Fog (databases, information systems, file processing), Cloud (data center, clusters of processors and powerful video cards).

D. Constructing Recognition Algorithms

Algorithms and their tunable parameters for *Step 3* are shown in Fig. 3. The idea of this step is a consistent description of the actions and processes taking place inside the event. The developer should choose the description of events according to his preference: these can be, for example, visual models or UML diagrams.

The proposed description (event model) includes the beginning and the end of the process. A single event can be described (then the base point will be the beginning of the event, and the end point will be the final event), and the whole algorithm with several events. This description may include several subcircuits in which an event is defined. The flowchart should reflect the algorithm of sequential actions, which in one way or another will lead to the execution of the algorithm and its end. The following notation is used.

- VA is video analytics (video analytics method).
- VCA is video content analytics (video content analysis method, situational video analytics).
- LIB is library, library (a collection of typical algorithms used repeatedly in several modules).
- PPE is personal protective equipment.

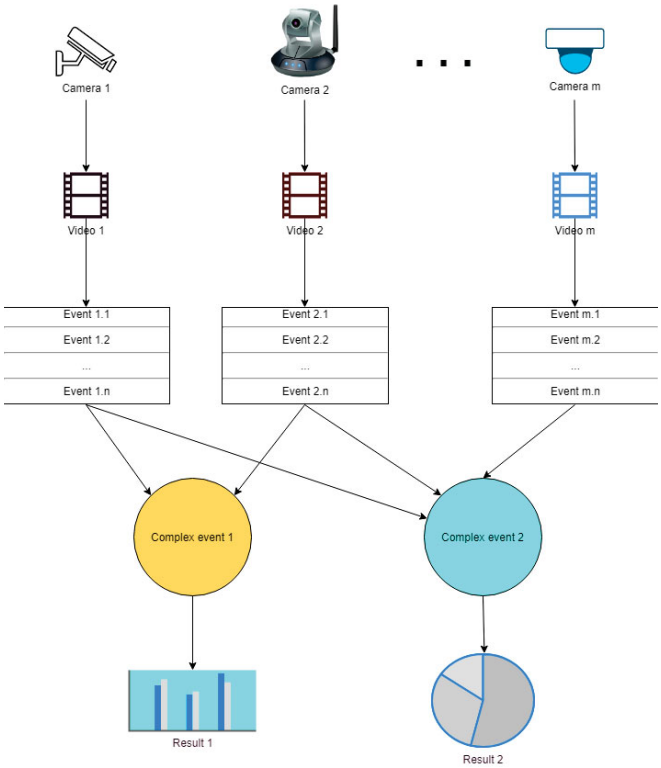

Fig. 2. Step 1. Unified event model based on different events from video streams

Therefore, the output va module provides a frame and possible configurations/parameters. The output vca module provides situational video analytics events.

Fig. 4 shows the architecture that organizes a required interaction of modules. As part of the interaction we describe libraries and modules based on recognition methods:

- Video processing library: lib-va-video-processing.
- Video customization and transformation module: va-video-transform.
- Personnel control module, counting and accounting of people on the territory: vca-personal-control.
- Module for identifying a person by face, recognizing faces, wearable PPE: vca-face-ppe-detection.
- Module for person trajectory tracking: vca-trajectory-tracking.
- The module for recognizing human activity in dynamics by its skeleton: vca-activity-recognition.

The introduced steps describe the methods that the proposed approach includes. The methods are used in development of a particular VSS with particular smart services. The next section describes the tools that the proposed approach exploits.

IV. Software Modules

The proposed methods are supported with implemented software modules. Each module provides a tool for VSS development to apply the methods according the proposed approach. We implemented the following modules.

- Video streaming module. Used for restreaming video form other sources to other modules.
- Person tracking module. Used for presence detection in particular areas.
- Face recognition module. Used to recognize a person by face from the given video stream.
- Trajectory computing module. Used to compute the trajectory that a person is walking in the given video frames.
- Human activity recognition module. Used to detect particular activity a person is performing at the moment.

A. Software Modules for VSS Development

Video Data Processing Library: The library contains various support functions to access databases and to send events.

Video Streaming Module: Video streaming module allows for single connection to camera to be retransmitted into arbitrary number of services.

1) Read config with information on how to connect to cameras and more.
   Config contains information about:
   a) Connection type (IP-camera, Web-camera, local file).
   b) URL to camera (camera device or file path).
   c) ZeroMQ port for decoded frame publication.
   d) Number of frames per second that needs to be published from original stream.
   e) If streamed video should also be saved on disk.
   f) Directory where video should be saved.
   g) Camera name.
2) For each connection separate process created.
3) Check to see if camera available.
   Test connection performed used only to get single frame from stream and some additional information. If module unable create connection it will wait for 5 minutes and try again. As result this step yields us resolution of an image and confirmation that stream available.
4) Connect to camera and start receiving video stream.
   This step is start of main loop of Video streaming module. Connection to camera done with help of FFmpeg command.
5) Publish decoded frames via ZeroMQ.
6) Poll if FFmpeg command is still running and connection still active.
   If it turns out that connection dropped wait for 5 minutes and try to connect again.

Human Tracking Module: YoLoV4 was used to detect person in this module and in trajectory tracking.

1) During preparation area for detection should be selected. Bounding box as top left coordinates and bottom right.
2) At runtime, human gets detected on image.
   Image received from subscription to ZeroMQ port for a camera from video streaming module. YoLov4 neural network used to detect human on incoming frame. Bounding box for human on image (if any) returned as result.

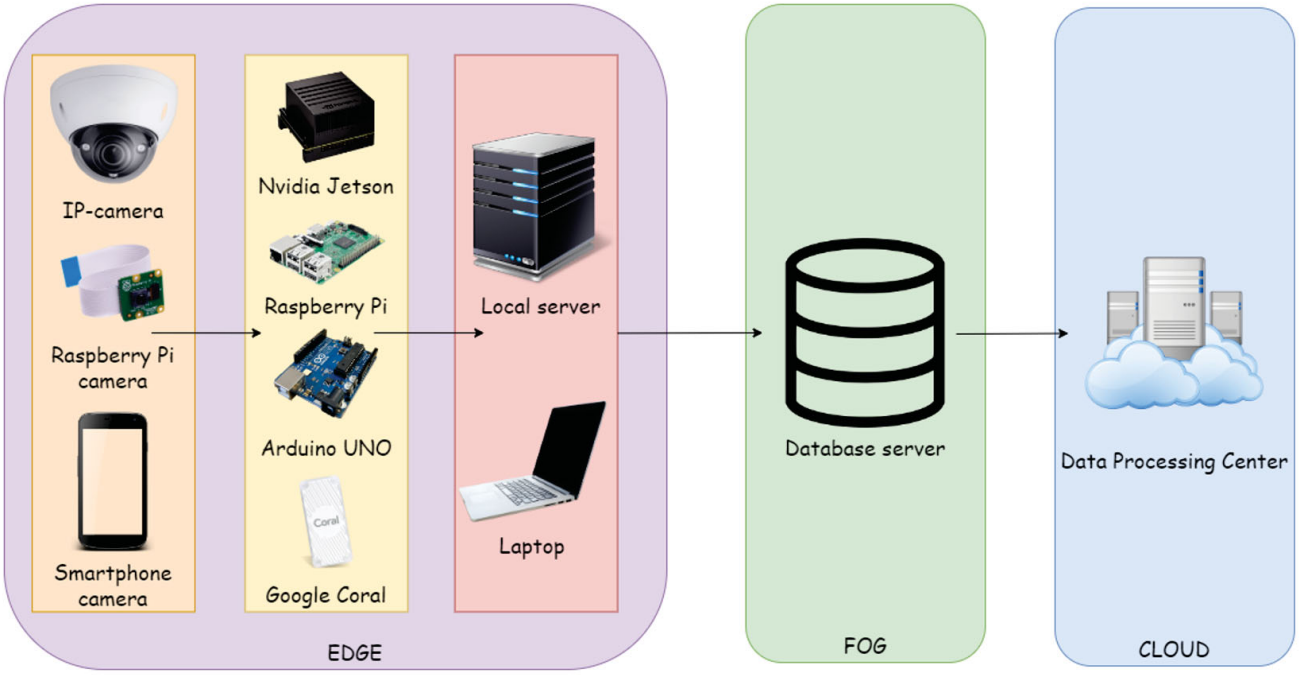

Fig. 3. Step 2. Architecture for video data processing

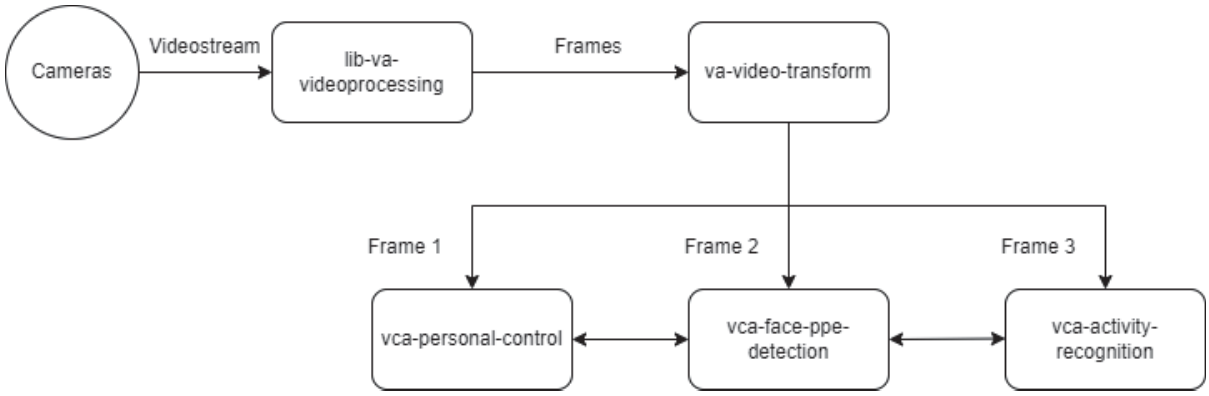

Fig. 4. Step 3. Composing recognition using several existing algorithms and tools

3) Overlap between found human bounding box and selected are during preparation compared.
   If overlap exceeds threshold value, then it would mean that human present in the selected area.

Face Recognition Module: The algorithms of face recognition are based on such existing methods as Facenet512 and OpenCV Face Detector.

1) Before face recognition module started it is necessary train it on portrait photos of humans it would need to recognize.
   Photos can be organized as flat files in folder with descriptive names. Photos would be processed by FaceNet512 to get feature vectors from them, that would be stored in database for later use.
2) At the beginning of module this feature vectors loaded from database.
3) At runtime, face gets detected with bounding box and feature vectors.
   Same FaceNet512 as in preparation step used to detect face on image here, and to compute feature vectors that could be compared to ones in the database.
4) As double check measure, OpenCV Detector runs to confirm that face found correctly.
   Such step allows to remove imperfection images, low quality photos, blurred, moving humans, not looking into cameras humans.
5) Distance between found and stored feature vector compared.
   Euclidean and cosine distances used to compare vectors.

Closest match reflects vectors that come from same human, with exception of too far away vectors.

Trajectory Computing Module: Human detection leads to a position of the person on every image. The module constructs a trajectory based on the recognized position points.

1) Human gets detected on image and their bounding box saved locally.
2) Human gets detected again on next image with their bounding box saved locally again.
3) Change in location between this two bounding boxes allow to calculate how person moves.
   To determine if found on current image human is same as on previous distance to previous saved position could be used: if new position is close enough to previous one, then they possibly are from the same track.
   If necessary to track multiple humans whose tracks are expected to overlap such methods as SORT [31] could be used.

Human Activity Recognition Module: The PoseNet neural network was used as a method for recognizing key points [32].

1) Human recognition.
2) Recognition of key points of the human body.
3) Building limb primitives.
4) Calculation of key angles.
5) Activity type definition.

If there is a sufficiently large frame in which a person occupies only a small part, it may be necessary to use human localization methods to simplify the work of the key point recognition method. In a situation where the person is the center of the frame, this step can be skipped.

Recognizable keypoints are the following (Fig. 5).

0) Mouth;
1) Left eye;
2) Right eye;
3) Left ear;
4) Right ear;
5) Left shoulder;
6) Right shoulder;
7) Left elbow;
8) Right elbow;
9) Left brush;
10) Right hand;
11) Left thigh;
12) Right thigh;
13) Left knee;
14) Right knee;
15) Left foot;
16) Right foot;

When constructing limb primitives, the following key points are used: hands, elbows, shoulders, hips, knees, feet. Based on the obtained key points, the primitives of the limbs are determined. Recognizable angles between limb primitives are as follows: elbows, shoulders, hips, and knees.

To recognize activity, information about the position of key points and the angles between them is used. For example, if a person is sitting, then his knees are bent at about 90 degrees. Thus, it is possible to set a specific activity through the description of the angles between the limbs.

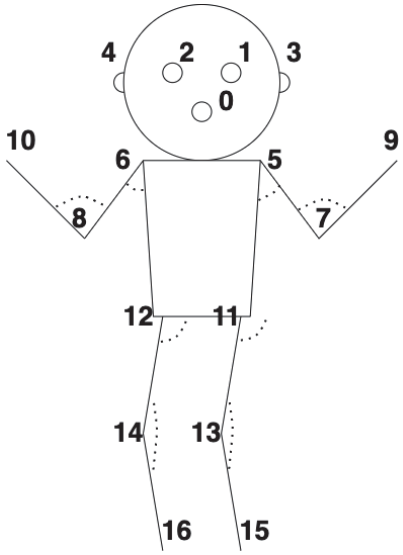

Fig. 5. Recognizable keypoints of human skeleton

1) Recognizable pose activity needs to be prepared beforehand.
2) At runtime human body keypoints are detected;
   Result of this step would be XY coordinates of each keypoints.
3) Limb primitives and key angles need to be calculated from keypoints as described earlier.
   For limb primitives result would be starting and end XY coordinates, and for key angles it would be angle between limb primitives.
4) Composition of such primitives compared to prepared activities.
   If the person is in the same position as on example, then the distance between the recognized pose would be low.

B. Experiments with Recognition Algorithms

Use Cases: The presented modules are exploited in several use cases, which show the efficiency of the technology.

One of the studied use cases is VSS for terrain security monitoring. Developed modules was used to check that only authorized personal entering restricted area (such as staff only rooms or closed perimeter objects). Also, it is used to monitor people check-in and check-out on the pass-through point. The use case is essentially based on event recognition [14].

Another use case is VSS systems for well-being and sport. Physical activity of a person (athlete, sportsmen) is under monitoring. The person uses training equipment when performing physical exercises. Movement of the person and the equipment is recognized to collect statistics of the equipment usage or to evaluate the quality of training. The latter case is suitable for edge-oriented VSS [5].

Experimental Hypothesis: The following hypothesis are tested in the experimental study.

Tests for the overall architecture.

1) How many cameras can be server at the time.
2) How many frames per second average.

Tests for the face recognition algorithms.

1) Measure accuracy.
2) Measure speed of execution.
3) How easy it is to add new faces.

Tests for activity recognition algorithms.

1) Measure accuracy.
2) Measure speed of execution.
3) How easy it is to add new activity.

Experimental Setup: A typical Personal Computer (PC) is used for running computations within the implemented modules. This configuration is oriented to the event-driven property [14] for smart VSS functions. The edge computing property [5] is not considered since relatively powerful computer is required for the recognition algorithms. The PC characteristics are the following.

- OS: Ubuntu 20.04;
- Net: 100Mb/s local;
- CPU: Intel Core i9-9900K (8 core, 16 threads);
- GPU: Nvidia RTX 2060;
- RAM: 32GB;
- SSD: 2TB m.2;

We use Hikvision as No 1 camera with the following characteristics (Fig. 6).

- Model DS-2TD2617B-3/PA.
- Resolution 2688×1520.
- Bitrate 16000 kbit/s;
- Frame rate 20 fps;

We use Hikvision as No 2 camera with the following characteristics (Fig. 7):

- Model DS-2CD2H83G2-IZS.
- Resolution 3840×2160.
- Bitrate 4000 kbit/s (dynamic).
- Frame rate 20 fps.

Human Recognition: In this experiment, the IP-camera is set up on ceiling (Camera No 1, Fig. 6). The person is walking in front of the camera. As a result, the algorithm recognizes events when the person is moving in or out of the observable area (i.e., either in or out of each frame). The recognition goal is to detect human as a physical entity, not to identify the person.

Similarly, the observable area can be divided into multiple zones (in advance). In each frame, the event is recognized in runtime when the person is in or out each zone.

The accuracy is evaluated as ratio of the number of frames with correct recognition of an in/out event to the number of all frames in the video stream. The recognition rate is about 15 ms per frame (one zone per camera as in Fig. 6). The recognition algorithm shows the accuracy approximately 95%. Similar result is for Camera No 2 that observes humans in a long hall (Fig. 7).

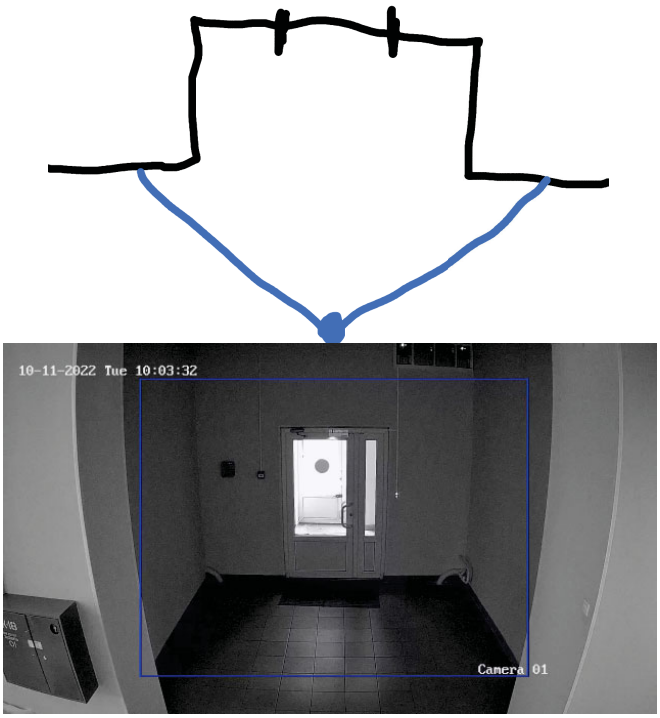

Fig. 6. Camera setup No 1 is monitoring a door

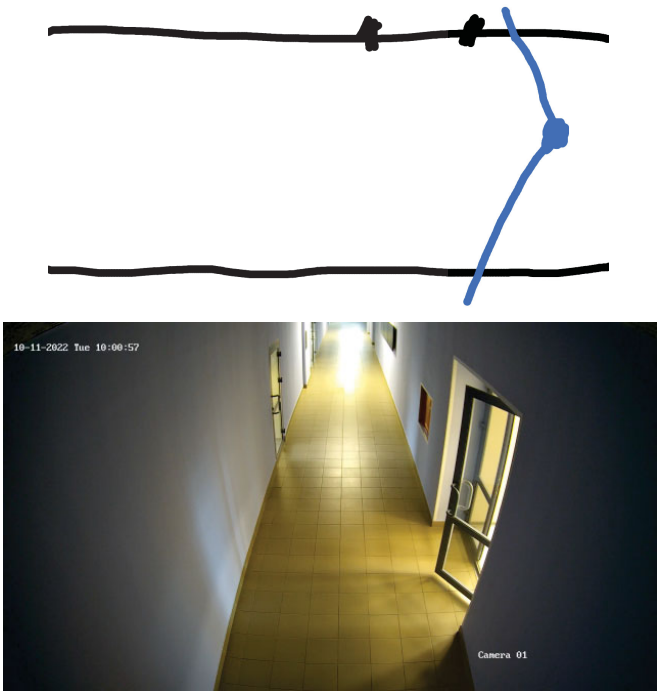

Fig. 7. Camera setup No 2 is monitoring a long hall

Face Recognition: In this experiment, the IP-camera observes an entrance zone in some building (Camera No 1, Fig. 6). Five persons are registered in the system in advance,

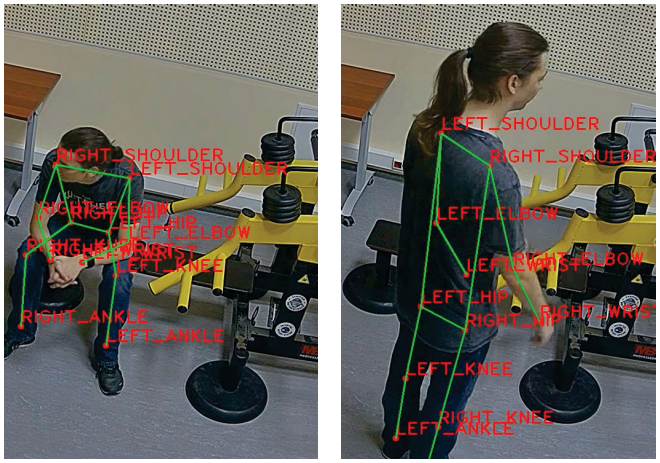

Fig. 8. An example of human activity recognition: a) the person is standing, b) the person is sitting

i.e., appropriate photos with faces are stored. Then the photos are used for identification of a registered person in runtime.

The recognition rate is about 130 ms per frame. The accuracy depends essentially on the quality of a) the registered photos (for the five persons) and b) the frames coming from the camera in runtime.

The recognition algorithm has reasonable accuracy if the following requirements to registered photos are satisfied.

- A photo should be taken from the front.
- The person stands in front of white background.
- Light is distributed uniformly on the person face.

Similar result is for Camera No 2 that observes humans in a long hall (Fig. 7).

The system supports extension. Additional photos with faces can be selected from the frames captured in runtime, either for already registered person or for a new person. Appropriate feature vectors of faces are added for further comparison with features observed in runtime.

Human Activity Recognition: In this experiment, the following activity types are selected for recognition: a) a person stands or walking around, b) a person sits or lays down.

The recognition results are demonstrated in Fig. 8. The person (Egor Rybin as a coauthor of this paper) is moving around the given training equipment. The algorithm constructs keypoints of human skeleton. An event is detected when one of the two activity types is recognized in each frame (based on the shape that the keypoints formed).

The recognition rate is about 20 ms per frame, i.e., the performance is 50 fps. The accuracy of the algorithm is about 93% for the tested video stream.

The system supports extension. New activity types can be added for the recognition. Appropriate template (model) is constructed to describe possible shapes (keypoints and angles between limb primitives). The feature vectors are rebuilt for further comparison with shapes observed in runtime in frames. The addition is relatively easy compared with training a neural network.

V. Conclusion

This paper considered AI-based technologies of VSS development. We studied several development problems that cannot be effectively solved using existing technologies: a) a generic model of events in video data for a given problem domain, b) a hardware–software architecture for data processing with existing recognition algorithms, and c) a model to construct a required smart VSS function using existing software tools. We introduce a novel semantic-oriented event-driven approach to solve the above VSS development problems. The approach provides methods and tools that are applied in given VSS development to implement a required smart VSS function. Our software modules include recognition algorithms for human detection and tracking, face recognition, trajectory computing, and activity recognition. The modules are experimentally used in several use cases, which showed the applicability of the proposed approach in terms of the accuracy and performance.

Acknowledgment

This R&D study is implemented with financial support by Russian Science Foundation, project no. 22-11-20040 (https://rscf.ru/en/project/22-11-20040/) jointly with Republic of Karelia and funding from Venture Investment Fund of Republic of Karelia (VIF RK). The work is in collaboration with the Artificial Intelligence Center of PetrSU.

References

[1] Q. Zhang, H. Sun, X. Wu, and H. Zhong, "Edge video analytics for public safety: A review," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1675–1696, 2019.

[2] B. Janakiramaiah, K. Gadupudi, and A. Jayalakshmi, "Automatic alert generation in a surveillance system for smart city environment using deep learning algorithm," *Evolutionary Intelligence*, vol. 14, 2021.

[3] T. Ayuningsih, A. Suhendar, and S. Suyanto, "Feasibility study of artificial intelligence technology for home video surveillance system," in *2022 1st International Conference on Information System & Information Technology (ICISIT)*, 2022, pp. 210–215.

[4] R. Rajavel, S. K. Ravichandran, K. Harimoorthy, P. Nagappan, and K. R. Gobichettipalayam, "IoT-based smart healthcare video surveillance system using edge computing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 13, no. 6, pp. 3195–3207, mar 2021.

[5] N. Bazhenov, A. Harkovchuk, and D. Korzun, "Edge-centric video data analytics for smart assistance services in industrial systems," in *Proc. 14th Int'l Conf. on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM)*. IARIA XPS Press, Oct. 2020.

[6] S. Trilles Oliver, A. Gonzalez-Perez, and J. Huerta, "An IoT platform based on microservices and serverless paradigms for smart farming purposes," *Sensors*, vol. 20, 2020.

[7] D. Korzun, E. Balandina, A. Kashevnik, S. Balandin, and F. Viola, *Ambient Intelligence Services in IoT Environments: Emerging Research and Opportunities*. IGI Global, 2019.

[8] V. Kulkarni and K. Talele, "Video analytics for face detection and tracking," in *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2020, pp. 962–965.

[9] A. L. Nair S. and R. K. Megalingam, "Human action recognition: A review," in *2021 10th International Conference on System Modeling & Advancement in Research Trends (SMART)*, 2021, pp. 249–252.

[10] M. A. Bouras, A. Ullah, and H. Ning, "Synergy between communication, computing, and caching for smart sensing in internet of things," *Procedia Computer Science*, vol. 147, pp. 504–511, 2019.

[11] E. Coronado, T. Kiyokawa, G. A. G. Ricardez, I. G. Ramirez-Alpizar, G. Venture, and N. Yamanobe, "Evaluating quality in human-robot interaction: A systematic search and classification of performance and human-centered factors, measures and metrics towards an industry 5.0," *Journal of Manufacturing Systems*, vol. 63, pp. 392–410, 2022.

[12] U. Guler, T. B. Tufan, A. Chakravarti, Y. Jin, and M. Ghovanloo, "Implantable and wearable sensors for assistive technologies," in *Reference Module in Biomedical Sciences*. Elsevier, 2021.

[13] O. Petrina, S. Marchenkov, and D. Korzun, "A semantic space-time event representation model in production equipment monitoring of technical state and utilization condition," in *2022 8th International Conference on Control, Decision and Information Technologies (CoDIT)*, 2022, pp. 1362–1367.

[14] N. Bazhenov and D. Korzun, "Event-driven video services for monitoring in edge-centric internet of things environments," in *Proc. 25th Conf. Open Innovations Association FRUCT*, Nov. 2019, pp. 47–56.

[15] S. S. A. Zaidi, M. S. Ansari, A. Aslam, N. Kanwal, M. Asghar, and B. Lee, "A survey of modern deep learning based object detection models," *Digital Signal Processing*, vol. 126, 2022.

[16] J. L. P. Diaz, A. Dorn, G. Koch, and Y. Abgaz, "A comparative approach between different computer vision tools, including commercial and open-source, for improving cultural image access and analysis," in *2020 10th International Conference on Advanced Computer Information Technologies (ACIT)*, 2020, pp. 815–819.

[17] V. Tsakanikas and T. Dagiuklas, "Video surveillance systems-current status and future trends," *Computers and Electrical Engineering*, vol. 70, 11 2017.

[18] H. Zhang, N. Yang, Z. Xu, B. Tang, and H. Ma, "Microservice based video cloud platform with performance-aware service path selection," in *Proc. 2018 IEEE Intl Conf. on Web Services (ICWS)*, Jul. 2018, pp. 306–309, 8456365.

[19] D. Pascale, G. Cascavilla, M. Sangiovanni, D. Tamburri, and W.-J. Heuvel, "Internet-of-things architectures for secure cyber-physical spaces: the visor experience report," 04 2022.

[20] M. N. Hassan Reza, C. Agamudai Nambi Malarvizhi, S. Jayashree, and M. Mohiuddin, "Industry 4.0–technological revolution and sustainable firm performance," in *2021 Emerging Trends in Industry 4.0 (ETI 4.0)*, 2021, pp. 1–6.

[21] D. Eneko Ruiz de Gauna, E. Irigoyen, I. Cejudo, H. Arregui, P. Leskovsky, and O. Otaegui, "Video analytics architecture with metadata event-engine for urban safe cities," ser. ICCTA 2021. New York, NY, USA: Association for Computing Machinery, 2021, pp. 46–52. [Online]. Available: https://doi.org/10.1145/3477911.3477919

[22] W. Naiyapo and W. Jumpamule, "An event driven approach to create UML models," in *Proc. 22nd Intl Computer Science and Engineering Conference (ICSEC 2018)*, Nov. 2018, pp. 1–5, 8712621.

[23] A. Hampapur, R. Bobbitt, L. Brown, M. Desimone, R. Feris, R. Kjeldsen, M. Lu, C. Mercier, C. Milite, S. Russo, C.-F. Shu, and Y. Zhai, "Video analytics in urban environments," 09 2009, pp. 128–133.

[24] A. Khochare, K. Sheshadri, R. Shriram, and Y. Simmhan, "Dynamic scaling of video analytics for wide-area tracking in urban spaces," in *2019 19th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID)*, 2019, pp. 76–81.

[25] A. D. and R. I. Minu, "Edge computing based surveillance framework for real time activity recognition," *ICT Express*, vol. 7, 05 2021.

[26] I. Bedhief, L. Foschini, P. Bellavista, M. Kassar, and T. Aguili, "Toward self-adaptive software defined fog networking architecture for iiot and industry 4.0," in *2019 IEEE 24th International Workshop on Computer Aided Modeling and Design of Communication Links and Networks (CAMAD)*, 2019, pp. 1–5.

[27] A. Alam, I. Ullah, and Y.-K. Lee, "Video big data analytics in the cloud: A reference architecture, survey, opportunities, and open research issues," *IEEE Access*, vol. 8, pp. 152 377–152 422, 2020.

[28] H. Pan, Y. Li, and D. Zhao, "Recognizing human behaviors from surveillance videos using the SSD algorithm," *The Journal of Supercomputing*, vol. 77, no. 7, pp. 6852–6870, Jan. 2021. [Online]. Available: https://doi.org/10.1007/s11227-020-03578-3

[29] L. Sun, Y. Li, and R. Memon, "An open iot framework based on microservices architecture," *China Communications*, vol. 14, pp. 154–162, 02 2017.

[30] S. Zhang, C. Wang, Y. Jin, J. Wu, Z. Qian, M. Xiao, and S. Lu, "Adaptive configuration selection and bandwidth allocation for edge-based video analytics," *IEEE/ACM Transactions on Networking*, vol. 30, no. 1, pp. 285–298, 2022.

[31] N. Wojke and A. Bewley, "Deep cosine metric learning for person re-identification," in *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*. IEEE, 2018, pp. 748–756.

[32] A. Kendall, M. Grimes, and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization," 2015. [Online]. Available: https://arxiv.org/abs/1505.07427