Time aware task delegation in agent interactions for video-surveillance

Paolo Sernani[1], Matteo Biagiola[2,3], Nicola Falcionelli[1],
Dagmawi Neway Mekuria[1], Stefano Cremonini[4], Aldo Franco Dragoni[1]

[1] Dipartimento di Ingegneria dell'Informazione,
Università Politecnica delle Marche,
Ancona, Italy
{p.sernani, a.f.dragoni}@univpm.it,
{n.falcionelli, d.n.mekuria}@pm.univpm.it

[2] Fondazione Bruno Kessler,
Trento, Italy
biagiola@fbk.eu

[3] Università degli Studi di Genova,
Genova, Italy

[4] Site Spa, Bologna, Italy
s.cremonini@sitespa.com

Abstract. Cameras are everywhere and the interest towards distributed surveillance systems is growing both in academic research and commercial applications. Multi-Agent Systems (MASs) are ideal to design and develop such applications: distributed by nature, the capability of software agents to communicate using messages and interaction protocols can be exploited to coordinate and control distributed surveillance system. However, there is also the need to optimize the use of available resources as the bandwidth to achieve on-line performances for automatic analysis algorithms. In this regard, this paper presents a multi-agent distributed video surveillance system to perform face recognition. The main goal of the system is to reduce the need to transmit the frames to be analyzed over the network. Each node of the system checks the available elaboration time to decide whether the face recognition should be performed locally (in the node which detected the faces) or remotely (in other nodes of the system), delegating the task. To achieve the delegation, the agents interact via a market-based protocol, the Extended Contract Net Protocol (ECNP). The test results show a two order of magnitude decrease in the size of data transmitted over the network to perform the face recognition. In addition, the proposed agent architecture is a first step towards more general real-time compliant multi-agent systems, by using the elaboration time to regulate the agents' behaviours. Future steps include a deeper analysis of the interactions among agents to meet strict time constraints.

1 Introduction

Cameras are pervasive in everyday life: most of the things of the Internet of Things (IoT) are cameras [21]. Such pervasiveness is widening the interest towards intelligent distributed surveillance systems, i.e. systems aiming to perform real-time monitoring of persistent and transient objects within a specific scene [24]. Thus, distributed surveillance systems are challenged to exhibit online performances adequate to real surveillance scenarios in a cost-efficient manner, including the economical use of the available bandwidth [1]. Multi-Agent Systems (MASs) can address such challenges by exploiting the properties of software agents: distribution, autonomy, and social ability [27], i.e. the capability to asynchronously communicate via interaction protocols, can be applied to coordinate and control distributed surveillance systems [19].

In addition to distributed surveillance, MASs have been proven useful in several domains, including telerehabilitation [9], personalized medicine [12, 18] and Ambient Assisted Living [6], manufacturing [17], and energy systems [11]. However, MASs still lack the capability to address time constraints [7].

The presented research makes a first step in the direction of real-time compliant MASs in the field of distributed surveillance systems. Specifically, this paper presents a multi-agent distributed video surveillance system with the goal to perform face recognition on people detected in video streams. Each node of the system is directly connected to a camera and is capable to autonomously detect and recognize faces in the video stream. Each node is also capable to delegate the recognition tasks in case the available elaboration time is not enough to recognize all the detected faces and guarantee a predetermined frame rate. Moreover, such task delegation is fully decentralized: the agents of the system delegate their recognition tasks using a market-based protocol to interact. Therefore, the proposed system makes two steps beyond the state of the art of distributed video surveillance systems:

– it exploits the autonomy of the network nodes, reducing the need to transmit over the network frames for the elaboration; only the faces that a node cannot elaborate due to its limited time resources are sent to other nodes over the network.
– it introduces the use of time to distribute the workload among the nodes, making the agents aware of the time available for the elaboration, in order to preserve a predetermined frame rate for the face detection and recognition.

This paper also introduces some preliminary experiments run with the proposed system. The results show a two order of magnitude reduction of the data sent through the network in the proposed video surveillance scenario, with respect to a system where all the frames need to be sent to one or more remote servers.

The rest of the paper is organized as follows. Section 2 compares agent-based distributed surveillance systems available in scientific literature to the system proposed in this paper. Section 3 presents our multi-agent architecture for a distributed video surveillance system and the time-aware task delegation with a market-based interaction protocol. Section 4 evaluates the proof-of-concept implementation of the proposed system presenting the experimental settings and the test results. Finally, Section 5 concludes the paper and outlines the future works.

2 Related works

Using MAS and Agent-Oriented Programming to design and develop distributed video surveillance systems is not a new concept. For example, they have been already proposed for the coordination and control of systems composed of multiple and heterogeneous cameras [19]. However, new challenges are arising in IoT, especially to build MASs able to meet time constraints and deadlines [7, 8]. The proposed research makes a step towards time-aware agents in MASs, proposing to take into account the needed elaboration time in order to balance the workload of the video surveillance system at runtime.

In the realm of the distributed video surveillance systems, San Miguel et al. [20] propose a distributed analysis framework based on the client/server model: the cameras acquire the videos and all the frames are sent to the servers over an ethernet connection; the authors do not provide any description about the allocation of the frames to the available servers to perform the video analysis. Instead, in our system, we do not send all the frames: each node is capable to extract frames from the streams and locally performs face detection and recognition. In case the node does not have enough time to analyze all the detected faces to guarantee a predetermined frame rate, it sends only the extra faces to other nodes of the network. The allocation of extra faces to other nodes is based on a market-based protocol.

A multi-agent video surveillance system is presented by Lefter et al. [16]: a set of observation agents coupled with the cameras of the network recognizes and tracks detected objects. The extracted features are sent to a reasoning agent which uses data such as speed and position of objects to detect potential suspicious conditions. The authors test a proof-of-concept implementation based on Jade, focusing on the performances of the tracking algorithm and on the feasibility of the reasoning, without taking into account distribution and allocation of tasks to respect time constraints. On the contrary, our approach specifically focuses on the distribution of the face recognition task, using the time needed for the execution as a constraint for the frame rate of the elaboration.

The multi-agent system proposed by Chao and Jun [10] focuses on the capability of terminal nodes of a video surveillance network to process raw data, in order to reduce the transmission of data. Frames and video streams are transmitted only upon direct client request or if a suspicious event is detected. Similarly, the reduction of transmitted data guides our system. However, we also propose an approach to distribute the workload when the terminal nodes have limited resources.

Kumar et al. [15] present a distributed surveillance system based on the execution of mobile agents. In particular, agents are able to migrate from a node to another, for example for tracking purposes. The authors propose to encapsulate surveillance tasks into mobile agents which can migrate in the nodes of the network. The migration is used to make the system scalable and the nodes independent from the task they have to execute in a given moment. However, the criterion to move one agent is the specific task that needs to be executed in a certain node. Instead, in our system, agents delegate the task on the basis of the elaboration time they have.

3 A multi-agent architecture for video surveillance

The proposed distributed video surveillance system consists of peer nodes able to locally perform the tasks needed to identify known subjects in a video stream: the detection and recognition of faces in the frames of the video. The goal is to allow the devices which acquire the video streams, i.e. the nodes of the system, to perform locally most of the computation to recognize the detected faces. Hence, the system should achieve the reduction of the need to transmit the images to be processed in remote servers. The term of comparison for such reduction is a centralized system or a set of servers which need to receive all the frames from all the video streams.

Each node of the system hosts a MAS including four agents, namely the Extractor, the Detector, the Recognizer, and the Contractor. The Extractor is responsible for the extraction of frames from the video stream whenever the Detector asks for a frame to be analyzed. The Detector is responsible for the detection of faces in a frame of the video, applying an LBP-based cascade classifier according to the Viola-Jones method [25], as provided by the OpenCV library[5]. The Detector also asks for new frames when the recognition task is completed. The Recognizer is the agent which actually performs the face recognition task by applying the Local Binary Pattern Histogram (LBPH) classifier [2] available in the OpenCV library. It is worth to remark that the accuracy of the identification algorithm is beyond the goals of this paper: the focus is on proposing solutions to distribute the recognition tasks among the nodes, trying to limit the number of images that need to be transmitted over the network, with respect to a centralized recognition system. In the experiments described in Section 4, the ready-to-go LPBH implementation has been used, but it's possible to change the recognition algorithm in other tests. The Recognizer is responsible to recognize the faces received from the Detector running on the same node, and, in case it has enough time in its slot, the faces received from other nodes through the Contractor. In fact, the Contractor manages the interactions with the Contractor agents of the other nodes of the system. Specifically, the Contractor calls for proposals to delegate the recognition to other nodes, when the Recognizer in the same node is not able to perform the identification of all the faces detected in a frame. Therefore, the Contractor is also responsible for sending proposals to recognize faces in case the Recognizer in the same node has some free time in the current time slot. At runtime, the Detector agent asks for a frame to be analyzed to the Extractor and detects the faces in the frame. The Recognizer

[5] https://opencv.org/

evaluates its own capability to process all the detected faces. In case during the current time slot some extra faces cannot be recognized, the Contractor launches the call for proposals to the other nodes of the system.

To decide whether to delegate the recognition of some of the detected faces to other nodes, the Recognizer evaluates the time available to perform the recognition task, using the Worst Case Execution Time (WCET) needed to run the algorithm over one face, and the number of detected faces, as explained in Subsection 3.1. In this work, the time needed by the Contractor to communicate the delegation or to propose to recognize additional faces has not been taken into account. The task delegation has been designed according to a market-based approach, as described in Subsection 3.2.

3.1 Task delegation for the recognition of faces

The Recognizer agent is responsible for the recognition task, i.e. it has to recognize the faces coming from the Detector in the same node. In addition, the Recognizer might be available to recognize the faces coming from other nodes which are not able to locally process all the faces detected in one time slot. Thus, once the Recognizer receives the faces from the Detector, it has to check if it is able to run the recognition algorithm in the available time slot.

Let's suppose that we want the node to be able to analyze one frame every period $T_f = T + T_d$ where T_d is the time used for the detection of faces. This means that the Recognizer has a slot T to recognize the detected faces. Given the execution time $ET = N_d \cdot T_r$ needed by the Recognizer to perform the recognition of N_d faces detected, where T_r is the WCET needed by the recognition algorithm to run over one image of a face, two are the possible conditions for the Recognizer:

1. $ET > T$. The Recognizer has not enough resources to run the recognition algorithm on the detected faces. Hence, the Contractor has to send a call for proposals to the other nodes of the system, looking for nodes capable to process N_{sf} extra faces, according to the Equation 1.

$$N_{sf} = \left\lceil \frac{ET - T}{T_r} \right\rceil \tag{1}$$

2. $ET \leq T$. The Recognizer has enough resources to run the recognition algorithm on the detected faces. In case $ET < T$ and a call for proposals is sent by other nodes, the Contractor of the node can propose to recognize N_{rf} extra faces, according to the Equation 2.

$$N_{rf} = \left\lfloor \frac{T - ET}{T_r} \right\rfloor \tag{2}$$

To make an example, let us suppose the system has the requirement to analyze, at least, one frame every three seconds. Supposing 0.5 s for the detection phase, the Recognizer would have 2.5 s to recognize all the faces. In case 9 faces are detected, and the recognition execute for 0.5 s per face, the Recognizer has the capability to recognize 5 faces locally. According to the Equation 1 the Contractor has to look for other nodes capable to recognize the 4 extra faces. Such number might be reached also involving more than one node of the system. For example, in case the recognizer of another node has the capability to recognize 3 extra faces in its time slot, the Contractor on the same node can participate to the call for proposals by offering to recognize 3 faces.

3.2 Agent interactions between the nodes

To delegate the recognition task, the Contractors running in the different nodes interact according to a market-based protocol: the Extended Contract Net Protocol (ECNP) as proposed by Aknine et al [3]. Using a market-based approach as the ECNP allows to allocate the recognition tasks in a fully distributed manner without the need of a central decisor [4, 14], making the interaction robust (no single points of failure) and modular (nodes can be added or removed from the system at runtime, without the need of changing the delegation process). In such market-based approach, the Contractor agents are selfish: they act to achieve the goal of recognizing all the faces detected in the nodes hosting them.

We choose the ECNP instead of the standard FIPA Contract Net [13] to allow the Contractors participating in several "call for proposals" in parallel. With the standard FIPA Contract Net protocol, a Contractor that proposes to recognize some faces and gets rejected could lose its chance to make a proposal to another Contractor. The "PreBid" and "Definitive Bid" performatives introduced by the ECNP avoid such danger [3].

Applying the ECNP to the distributed video surveillance system, a Contractor can play the role of the initiator of the protocol, in case the Recognizer in the same node has not enough resources to recognize all the faces (Equation 1). On the contrary, a Contractor plays the participant role in case the Recognizer has the capability to recognize faces from other nodes (Equation 2). Both the initiator and the participant can be represented by finite state machines (FSMs).

One might argue that the Recognizer could directly interact with other nodes to delegate the recognition of some faces, instead of the Contractor agent. However, separating the logic of the interactions among the nodes (thanks to the Contractor) from the local recognition task performed by the Recognizer has two advantages:

- the market-based interaction protocol among the nodes can be easily replaced by re-defining the logic of the Contractor agents, without interfering with the recognition (and vice versa), making the system modular;
- at runtime, different agents can run in different threads, allowing to execute the local recognition in parallel with the interactions to delegate the extra faces.

Initiator. Figure 1 depicts the FSM representing the initiator role for a Contractor in the ECNP. Once the Recognizer has faces which cannot be recognized in the current time slot, the Contractor on the same board plays the role of

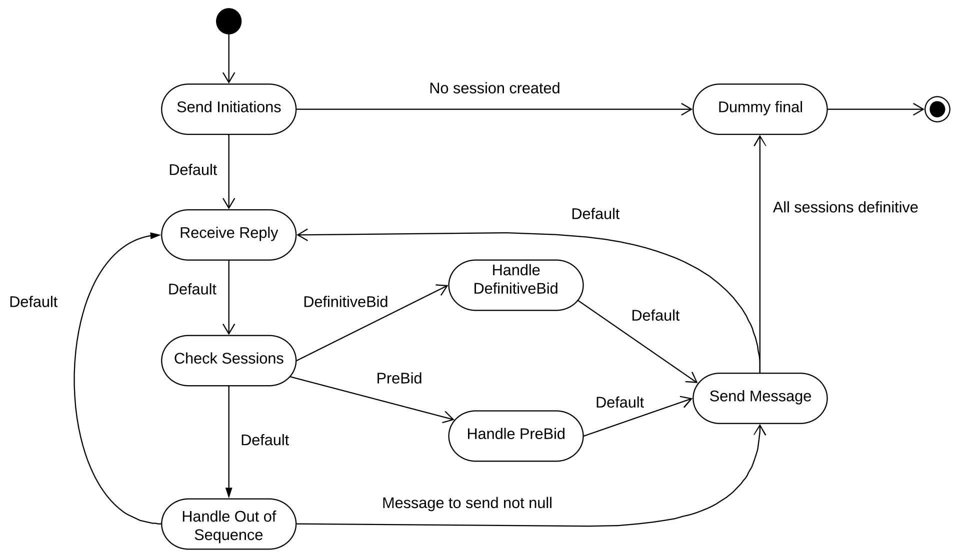

Fig. 1. The FSM for the initiator role played by a Contractor.

the initiator of the ECNP: it starts an instance of the protocol from the state "Send Initiations", sending an *Announce* message to the other Contractors of the system. The message actually indicates to the potential participants that the initiator is looking for nodes to recognize its extra faces. After sending the *Announce* message, the initiator waits for replies from the participants (state "Receive Reply"). As soon as a reply arrives, the initiator checks the received message (state "Check Sessions"): a participant might refuse to accept faces from the initiator, in case it has not enough resources in its time slot. In such case the initiator goes to state "Handle Out of Sequence" to terminate the protocol with such participant and wait for other messages (if there are other participants still involved in the protocol) or end the protocol without recognizing the extra faces. In fact, the possibility to send a *Refuse* message is not defined in the original formulation of the ECNP. However, taking inspiration from the FIPA Contract Net, we added the *Refuse* message for participants with no time to recognize extra faces, avoiding their involvement in the next phases of the protocol. In the "Receive Reply" state, a reply from a participant can have a *PreBid* performative, indicating that the participant is proposing to recognize some faces to the initiator. The content of a *PreBid* message is the number of faces that the participant can recognize. The initiator goes to the "Handle PreBid" state where two are the possible cases:

1. The initiator does not have extra faces to be recognized, since the already received *PreBids* were enough to recognize all the faces of the node. Then, the initiator send a *Definitive Reject* message (state "Send Message") to the participant which ends the protocol.
2. The initiator still has extra faces to be recognized.

(a) If the *PreBid* was for 0 faces, meaning a participant with free time already committed for the recognition with other nodes, the initiator sends a *PreReject* message to the participant. The participant can send again a *PreBid*, with a content greater than 0, in case the other nodes did not delegate it any face to recognize at the end of the protocol.
(b) If the *PreBid* was greater than 0, the initiator delegates to the participant a number of faces equal to the minimum between the *PreBid* and the faces to be recognized, by sending a *PreAccept* message.

Finally, a reply from a participant can have a *Definitive Bid* performative, to confirm the number of faces to be recognized according to the *PreAccept* message. The initiator sends a *Definitive Accept* message to the participant, with the face images to be processed. Once the initiator distributed all its extra faces to be recognized, it sends a *Definitive Reject* to all the other participants which previously sent a *PreBid*. When all the expected results from the participants arrive, the initiator ends the protocol (state "Dummy final").

Participant. Figure 2 shows the FSM of the participant role for a Contractor in the ECNP.

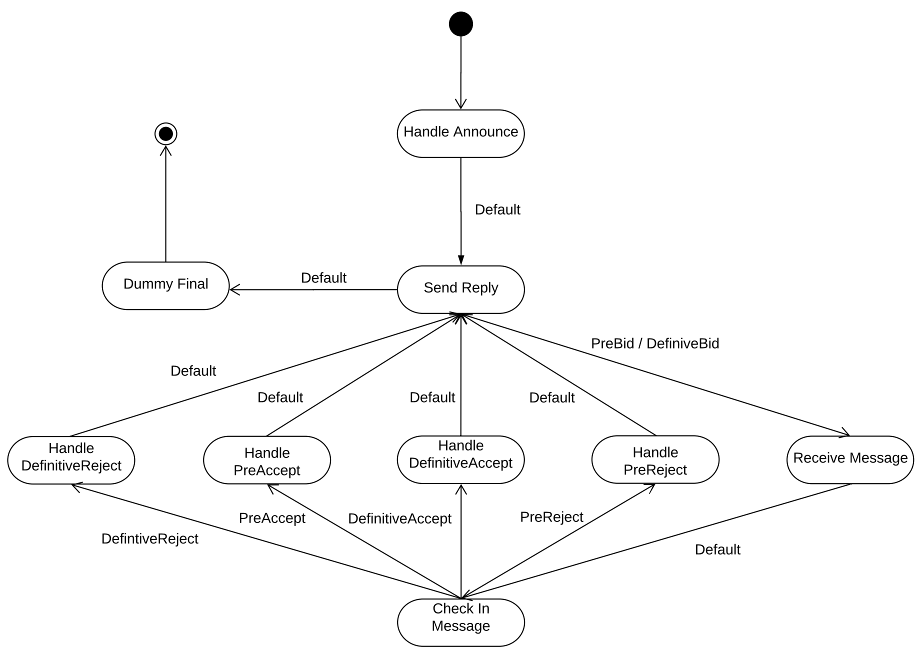

Fig. 2. The FSM for the participant role played by a Contractor.

A Contractor plays the role of a participant when it receives an *Announce* message from an initiator looking for nodes capable to recognize some faces.

Therefore, the participant starts from the state "Handle Announce". In case the participant has no free time in its slot for the recognition of faces from other nodes, it sends a *Refuse* message to the initiator (state "Send Reply") ending the protocol (state "Dummy Final"). On the contrary, in case the participant has free time in its slot, it sends a *PreBid* message to the initiator, using Equation 2 to propose the number of faces it can recognize. In making its *PreBid*, a participant has to take into account the *PreBids* already sent to different Contractors. This means that the *PreBid* could be sent to recognize 0 faces, in case the participant committed all its available recognition time to other nodes. After sending a *PreBid*, the participant goes to the "Receive Message" state, waiting for a reply from the initiator. As soon as a reply arrive, the participant checks the performative of the received message (state "Check In Message"). According to such performative, the participant changes to one of the following states:

– "Handle PreAccept". The participant sends a *Definitive Bid* proposing to recognize the number of faces requested by the initiator.
– "Handle PreReject". The participant sends another *PreBid* with the number of faces currently able to recognize, according to its available time and other *PreBids* it has already sent to different Contractors.
– "Handle Definitive Accept". The participant forwards the faces to be recognized to the Recognizer on the same node and sends the results back to the initiator of the ECNP, ending the protocol.
– "Handle Definitive Reject". The participant ends the protocol, since its recognition time is not needed by the initiator.

4 Evaluation

The primary goal of the proposed video surveillance system is to reduce the need to transmit the frames and the faces to be analyzed over the network. Instead of sending the frames from the video streams to remote servers for the elaboration, each node of the network should perform locally the majority of the tasks. A node with too many faces to recognize can send to other nodes the face images that cannot be processed locally, in order to guarantee a predetermined frame rate for the analysis of videos. For example, this might happen when a crowd is recorded by one camera, while no one is in front of a camera of another node, with the sender node being very busy and the receiver free. Therefore, in order to assess the system and the load balancing achieved through the use of the ECNP, we compare the network load of the proposed system to the case where all the analyzed frames are sent remotely for the face recognition.

4.1 Experimental setup

Figure 3 shows the experimental setup used to run the tests and evaluate the proposed system. Six Intel Galileo boards (single core i586 CPU @ 400 MHz, 256 MB DRAM, 100 Mb Ethernet) are the nodes of the system. The boards are

Fig. 3. The experimental setup.

connected into a LAN with two switches and a router (which is not visible in the picture).

Each board simulates a device connected to a camera and locally performs the frame extraction from the video stream, as well as the face detection and recognition. Instead of using real cameras, we simulated the video streams with the video files from the ChokePoint dataset[6] [26], a collection of 48 videos with a resolution of 800×600 pixels at 30 fps. The videos are designed for experiments in person identification/verification under real-world surveillance conditions and represent 25 and 29 subject walking through 2 portals, recorded by three different cameras. The use of video files instead of real cameras does not compromise the tests: with the OpenCV functions the Extractor agent on a node can take a frame and advance the video of a given milliseconds timeout, simulating the video stream going on. In our experiment, we considered the videos from portal 2 (P2), involving 29 subjects. We used the faces detected in sequence 4 (S4) to train the recognition algorithm (LBPH). Then, we stored other sequences of portal 2 in the SD cards of the boards, in order to run the test. Table 1 provides the list of the video files used in each board. At runtime, such videos simulate the video stream of cameras connected to the boards. In fact, the videos in the boards 1, 2 and 5 are composed of many subjects recorded together through the portal, simulating busy network nodes with the need to delegate the recognition of some faces. On the contrary, the videos played in the boards 3, 4, and 6 include less subjects, simulating nodes capable to receive faces from the busy ones.

[6] http://arma.sourceforge.net/chokepoint/

Table 1. The videos from the ChokePoint dataset [26] used on the Galileo at runtime, to simulate the video streams from real cameras.

Board	**Video File**
Galileo 1	"P2L-S5-C2-ext"
Galileo 2	"P2E-S5-C2-ext"
Galileo 3	"P2L-S1-C1"
Galileo 4	"P2E-S1-C2"
Galileo 5	"P2E-S5-C2-ext"
Galileo 6	"P2L-S1-C1"

We implemented the proposed multi-agent architecture using the Jade Framework[7] [5]. The agent platform consists of a Jade container for each board. Each container host one Extractor, one Detector, one Recognizer, and one Contractor. One of the boards hosts the main container, with the FIPA standard Agent Management System (AMS) and Directory Facilitator (DF) which allows the Contractors discovering each other services. We extended Jade's *FSMBehaviour* class to implement the FSMs for the initiator and the participant roles in the ECNP, since Jade does not provide an implementation of such protocol. To extract the frames from the videos, detect and recognize the faces, we used the algorithms provided by the OpenCV library: thanks to the Java bindings the Jade agents directly call the needed functions and elaborate the results.

4.2 Tests and results

To measure the network load of the proposed system we summed up the number of bytes of the messages[8] exchanged by agents hosted in different containers, i.e. the messages exchanged between the boards, including those with the face images in the content. The messages sent by agents to other agents in the same container are not sent through the network (i.e. the messages exchanged by the Extractor, the Detector, the Recognizer, and the Contractor in the same node do not count for the network load). Furthermore, the messages used in the Jade platform to create containers and host agents are part of the network traffic and contribute to the total amount of the network load.

In the videos used to simulate the cameras connected to the nodes of the video surveillance system, a subject is in the field of view for a median time of 3 seconds. Hence, to run the tests, we set the period to analyze the video frames to 3 seconds: at least one frame every 3 s has to be processed by each board.

Table 2 summarizes the results of the tests. The system analyzes 447 frames in total that occupy 33.5 MB. This means that in a centralized architecture

[7] http://jade.tilab.com/

[8] In Jade, each message is serialized in a byte sequence before being sent with the Java Remote Method Invocation (RMI). Hence, it is possible to measure the length of a message counting the bytes composing the sequence, without using an external tool.

where the nodes only acquire the frames to be sent to some servers for the face detection and recognition, there would be at least 33.5 MB to be transmitted over the network. Instead, the nodes of the multi-agent video surveillance system are able to analyze locally 114 of the 145 faces detected in the videos, without any need to transmit the frames. This means that the Contractor agents in the boards have to look for other nodes capable to recognize the extra 31 faces. To look for Contractors of nodes capable to recognize extra faces, negotiate such task, and manage the platform, the network load is 179 KB. Hence, there is a

Table 2. The test results: the system analyzed 447 frames, detecting 145 faces. The network has been occupied for 179 KB.

Analyzed frames	Network load	Faces		
		Analyzed locally	Analyzed remotely	Lost
447 (33.5 MB)	179 KB	114	26	5

two orders of magnitude difference between the proposed system and one using a remote server to recognize the faces. The only drawback is that 5 faces are lost, i.e. the Contractors were not able to find other nodes with enough resources (enough time) to recognize those 5 extra faces in their time slots. During the tests, the accuracy achieved by the Recognizer agents was 57.27%, using LBPH as the recognition algorithm.

The test results highlight the advantages of distributing the detection and recognition tasks in the nodes of a video surveillance application. In addition to the decreasing of the network load, designing the system as a MAS using a market-based approach to distribute the load allows a fully decentralized allocation of the tasks that cannot be executed locally. Moreover, such market-based approach is robust to the addition or removal of nodes of the network.

4.3 Threats to the validity of the experiments

Being a first step towards real-time compliant multi-agent systems, the proposed experiments do inevitably suffer from threats to validity. Therefore, future works will address the identified limitations.

Concerning the internal validity, being based on Jade, the system runs in the Java Virtual Machine: hence, the WCET to recognize a face is approximated by tests on the Galileo Boards using the least squares method. A more precise analysis on the WCET should be performed. In addition, the time needed to interact using the ECNP is ignored. This is not an issue on the platform used for the tests, where the time to recognize a face is much higher than the time needed for the interaction.

Concerning the external validity, there is a lack of datasets tailored on video-based face recognition in video surveillance scenarios, especially to tests the network load balancing and the task delegation when many subjects are recorded at the same time. However, more experiments would be needed to generalize the results to different scenarios and use cases.

5 Conclusions and future works

We presented a multi-agent distributed video surveillance system to perform face recognition in video streams. Each node of the system is a camera plus an elaboration unit capable to locally process the video stream in order to extract frames as well as detect and recognize faces. In case a node cannot recognize all the detected faces due to a limited amount of elaboration time, it looks for other nodes in the network in order to delegate the task, starting a call for proposals with the ECNP, a market-based protocol. The proposed systems goes in the direction to make the agents in a MAS aware of the time, in order to meet time constraints. In addition, using the ECNP, the task delegation is fully decentralized. The tests on a proof-of-concept implementation of the proposed system highlighted the potential reduction of the network load: only the extra faces that a node is not able to recognize are sent over the network, instead of sending all the frames in the case where the elaboration is performed remotely.

Future works will address the threats to the validity of the presented experiments. Considering hard real-time scenarios also the interactions and the time needed to exchange messages should be part of the analysis in addition to the recognition WCET, going in the direction of real-time compliant interactions [7]. In fact, the ECNP does not comply with real-time requirements [8]. Therefore, an alternative has to be found, in order to make the entire process (including the interactions) capable to meet strict time constraints. Moreover, we used a ready-to-go LBPH implementation for the face recognition, since the presented work is focused on the distribution of the workload rather than on the recognition accuracy. Nevertheless, to exploit its potential in unconstrained scenarios, the recognition should be based on deep neural networks to obtain state-of-the-art accuracy [22, 23].

Acknowledgements

The authors thank Site SpA for the support provided for the presented research.

References

1. Abas, K., Porto, C., Obraczka, K.: Wireless smart camera networks for the surveillance of public spaces. Computer 47(5), 37–44 (2014)
2. Ahonen, T., Hadid, A., Pietikainen, M.: Face description with local binary patterns: Application to face recognition. IEEE transactions on pattern analysis and machine intelligence 28(12), 2037–2041 (2006)
3. Aknine, S., Pinson, S., Shakun, M.F.: An extended multi-agent negotiation protocol. Autonomous Agents and Multi-Agent Systems 8(1), 5–45 (2004)

4. Badreldin, M., Hussein, A., Khamis, A.: A comparative study between optimization and market-based approaches to multi-robot task allocation. Advances in Artificial Intelligence 2013, 1–11 (2013)
5. Bellifemine, F.L., Caire, G., Greenwood, D.: Developing multi-agent systems with JADE. John Wiley & Sons (2007)
6. Calvaresi, D., Cesarini, D., Sernani, P., Marinoni, M., Dragoni, A.F., Sturm, A.: Exploring the ambient assisted living domain: a systematic review. Journal of Ambient Intelligence and Humanized Computing 8(2), 239–257 (2017)
7. Calvaresi, D., Marinoni, M., Lustrissimini, L., Appoggetti, K., Sernani, P., Dragoni, A.F., Schumacher, M., Buttazzo, G.: Local scheduling in multi-agent systems: getting ready for safety-critical scenarios. In: 15th European Conference on Multi-Agent Systems EUMAS 2017 (2017)
8. Calvaresi, D., Marinoni, M., Sturm, A., Schumacher, M., Buttazzo, G.: The challenge of real-time multi-agent systems for enabling iot and cps. In: Proceedings of the International Conference on Web Intelligence. pp. 356–364. WI '17, ACM, New York, NY, USA (2017)
9. Calvaresi, D., Schumacher, M., Marinoni, M., Hilfiker, R., Dragoni, A.F., Buttazzo, G.: Agent-based systems for telerehabilitation: Strengths, limitations and future challenges. In: Agents and Multi-Agent Systems for Health Care: 10th International Workshop, A2HC 2017, São Paulo, Brazil, May 8, 2017, and International Workshop, A-HEALTH 2017, Porto, Portugal, June 21, 2017, Revised and Extended Selected Papers. pp. 3–24. Springer International Publishing, Cham (2017)
10. Chao, W., Jun, X.M.: Multi-agent based distributed video surveillance system over ip. In: 2008 International Symposium on Computer Science and Computational Technology. vol. 2, pp. 97–100 (2008)
11. Coelho, V.N., Cohen, M.W., Coelho, I.M., Liu, N., Guimarães, F.G.: Multi-agent systems applied for energy systems integration: State-of-the-art applications and trends in microgrids. Applied Energy 187, 820–832 (2017)
12. Falcionelli, N., Sernani, P., Brugués, A., Mekuria, D.N., Calvaresi, D., Schumacher, M., Dragoni, A.F., Bromuri, S.: Event calculus agent minds applied to diabetes monitoring. In: Agents and Multi-Agent Systems for Health Care: 10th International Workshop, A2HC 2017, São Paulo, Brazil, May 8, 2017, and International Workshop, A-HEALTH 2017, Porto, Portugal, June 21, 2017, Revised and Extended Selected Papers. pp. 40–56. Springer International Publishing, Cham (2017)
13. FIPA Contract Net: Fipa contract net interaction protocol specification. http://www.fipa.org/specs/fipa00029/ (2017), [Online; accessed 22 December 2017]
14. Khamis, A., Hussein, A., Elmogy, A.: Multi-robot task allocation: A review of the state-of-the-art. In: Koubâa, A., Martínez-de Dios, J. (eds.) Cooperative Robots and Sensor Networks 2015. pp. 31–51. Springer International Publishing (2015)
15. Kumar, P., Pande, A., Mittal, A., Mudgal, A., Mohapatra, P.: Distributed video surveillance using mobile agents. In: IEEE International Conference on Digital Convergence (ICDC 2011). pp. 199–204 (2011)
16. Lefter, I., Rothkrantz, L., Somhorst, M.: Automated safety control by video cameras. In: Proceedings of the 13th International Conference on Computer Systems and Technologies. pp. 298–305. CompSysTech '12, ACM, New York, NY, USA (2012)
17. Leitão, P., Barbosa, J., Trentesaux, D.: Bio-inspired multi-agent systems for reconfigurable manufacturing systems. Engineering Applications of Artificial Intelligence 25(5), 934–944 (2012)

18. Montagna, S., Omicini, A.: Agent-based modeling for the self-management of chronic diseases: An exploratory study. Simulation 93(9), 781–793 (2017)
19. Natarajan, P., Atrey, P.K., Kankanhalli, M.: Multi-camera coordination and control in surveillance systems: A survey. ACM Trans. Multimedia Comput. Commun. Appl. 11(4), 57:1–57:30 (2015)
20. San Miguel, J.C., Bescós, J., Martínez, J.M., García, Á.: Diva: a distributed video analysis framework applied to video-surveillance systems. In: Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on. pp. 207–210. IEEE (2008)
21. Satyanarayanan, M., Simoens, P., Xiao, Y., Pillai, P., Chen, Z., Ha, K., Hu, W., Amos, B.: Edge analytics in the internet of things. IEEE Pervasive Computing 14(2), 24–31 (2015)
22. Schroff, F., Kalenichenko, D., Philbin, J.: Facenet: A unified embedding for face recognition and clustering. In: 2015 IEEE Conference on Computer Vision and Pattern Recognition. pp. 815–823 (2015)
23. Taigman, Y., Yang, M., Ranzato, M., Wolf, L.: DeepFace: Closing the gap to human-level performance in face verification. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition. pp. 1701–1708 (2014)
24. Valera, M., Velastin, S.A.: Intelligent distributed surveillance systems: a review. IEE Proceedings-Vision, Image and Signal Processing 152(2), 192–204 (2005)
25. Viola, P., Jones, M.: Rapid object detection using a boosted cascade of simple features. In: Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. pp. 1–9 (2001)
26. Wong, Y., Chen, S., Mau, S., Sanderson, C., Lovell, B.C.: Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In: IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops. pp. 81–88. IEEE (2011)
27. Wooldridge, M., Jennings, N.R.: Intelligent agents: theory and practice. The Knowledge Engineering Review 10, 115–152 (1995)