Journal version available at https://doi.org/10.1016/j.ins.2015.03.021

A Hierarchical Neuro-Fuzzy Architecture for Human Behavior Analysis

Giovanni Acampora[b,*], Pasquale Foggia[a], Alessia Saggese[a], Mario Vento[a]

[a] *Dept. of Information and Electrical Engineering and Applied Mathematics, University of Salerno, 84084 Fisciano (SA), Italy*
[b] *School of Science and Technology, Nottingham Trent University, UK*

Abstract

Analysis and detection of human behaviors from video sequences has became recently a very hot research topic in computer vision and artificial intelligence. Indeed, human behavior understanding plays a fundamental role in several innovative application domains such as smart video surveillance, ambient intelligence and content-based video information retrieval. However, the uncertainty and vagueness that typically characterize human daily activities make frameworks for human behavior analysis (HBA) hard to design and develop. In order to bridge this gap, this paper proposes a hierarchical architecture, based on a tracking algorithm, time-delay neural networks and fuzzy inference systems, aimed at improving the performance of current HBA systems in terms of scalability, robustness and effectiveness in behavior detection. Precisely, the joint use of the aforementioned methodologies enables both a quantitative and qualitative behavioral analysis that efficiently face the intrinsic people/objects tracking imprecision and provide context aware and semantic capabilities for better identifying a given activity. The validity and effectiveness of the proposed framework have been verified by using the well-known CAVIAR dataset and comparing our system's performance with other similar approaches working on the same dataset.

Keywords: Behavior Understanding, Trajectories Analysis, Time Delay Neural Networks, Fuzzy Systems.

*Corresponding author: giovanni.acampora@ntu.ac.uk

1. Introduction

Human behavior analysis (HBA) is a new and hot research topic in the area of computer vision and artificial intelligence that is attracting the attention of several researchers, due its potential applications such as smart video surveillance, ambient intelligence and content-based information retrieval. The main task of HBA frameworks is to provide images and video with a semantic interpretation for trying to bridge the gap between their low-level representation in terms of pixels, and the high-level, natural language description that a human would give about them. In several application domains, this semantic-based approach may support human beings to overcome a well-defined *psychological overcharge* issue [1] that causes a decrease in human capabilities to analyze raw data flows from multiple sources of multimedia information (e.g. video surveillance) [2][3]; indeed, as stated by a study conducted by Security Solutions magazine, "After 12 minutes of continuous video monitoring, a guard will often miss up to 45% of screen activity. After 22 minutes of video, up to 95% is overlooked". Different analysis paradigms have been proposed to face this fascinating challenge [4][5][6][7] and, in particular, the approaches based on *bottom-up* and *top-down* analysis can be considered as the most commonly adopted in the current HBA literature. Precisely, bottom-up approaches analyze and interpret a human behavior based on low-level features of the video or image scene and try to semantically refine them by using a chain of methodologies from artificial intelligence and machine learning fields; vice versa, top down approaches try to recursively split a whole scene into a collection of semantically-defined events, whose joint analysis may enable the identification of the behaviors of humans populating the scene.

However, in spite of the growing interest in HBA, this research topic is still in its embryonic phase and, indeed, there are several challenges to be faced towards a fully suitable framework for detecting human activities. For instance, several approaches try to analyze a fixed collection of human behaviors and, as a consequence, they lack the scalability feature that could make the system robust enough to deal with novel and not yet considered human activities. Other proposals only analyze pixel-based information, such as trajectories, without taking into account the semantics of events, the scenario where human activity occurs and the dynamic relationships existing among different actors populating this scenario. Moreover, the uncertainty and vagueness that typically characterize human daily activities make HBA

frameworks hard to design and develop.

For all these reasons, we propose an effective and scalable HBA architecture capable of understanding human behaviors by analyzing motion information, through a context-awareness framework based on the joint exploitation of an enhanced tracking algorithm [8][9], neural computation and fuzzy logic theory. In particular, similarly to other approaches focusing on different application domains [10][11][12], we introduce a hierarchical architecture based on the integration of different computational intelligence techniques capable of providing benefits both from a *quantitative* and a *qualitative* point of view. In fact, the lower layer of the proposed architecture, based on suitably trained *time delay neural networks (TDNNs)*, analyzes the raw kinematic data obtained by a tracking algorithm, and provides a set of primitive behaviors (e.g. walking, running, stopping, loitering, etc.) performed by human beings moving in the scene under analysis; higher layers, based on a collection of *fuzzy inference engines (FIs)*, act as a context-aware process that semantically enriches the set of primitive behaviors and identify a collection of refined and context depending behaviors (e.g. meeting, walking together, money withdrawing, danger situation, etc.).

This architectural organization provides the resulting HBA system with high levels of scalability. Indeed, at low level, TDNNs can be opportunely re-trained in order to learn novel tracking information and, as a consequence, identify new primitive human behaviors. At high level, the modular approach used for designing the collection of FISs, enables our architecture to enhance its human behavior understanding capabilities by considering semantic relationships corresponding to new human activities. The separation between low and high level behaviors provides our system with improved capabilities from the point of view of behavioral learning. Indeed, only the lowest layer of our architecture, dealing with raw data, is depending upon some environmental features such as the camera view in terms of angle, distance and so on. Consequently, our system can be trained by using less data and a reduced computational effort than other learning-based HBA approaches.

Moreover, another significant benefit provided by the proposed architecture is related to its robustness to tracking imprecision; this feature is due to the exploitation of theories such as fuzzy logic and neural networks that naturally deal with vague or approximate information. The validity and efficiency of the proposed framework have been verified by using the well-known CAVIAR dataset and comparing our system's performance with other similar approaches working on the same dataset.

In the following, we will provide in Section 2 a panoramic summary of related works in the general area of human behavioral analysis; Section 3 introduces our proposal of hierarchical neuro-fuzzy HBA architecture; Section 4 will describe an experimental validation, performed on the standard CAVIAR database, and will discuss the obtained results; finally, in Section 5, we will present our conclusions and will outline some future developments.

2. Related Works

The challenging problem of human behavior analysis and understanding has been addressed by a large number of researchers in the last years. However, most of the research activities focused on the recognition of low level human actions [13][14] without taking into account how semantic relationships between a human and its surroundings can influence the detection of a complex behavior. For this reason, lastly, different analysis paradigms have been proposed to extend the conventional human behavior understanding paradigms, based on a low-level analysis, with enhanced context-aware capabilities for performing complex behavioral understanding. One of the first work on human behavioral analysis was introduced in 2004 by [15]; they present a framework, based on fuzzy self organizing maps, for learning patterns of object activities in image sequences for anomaly detection and activity prediction, and show that their method obtains a higher efficiency than Kohonen self organizing feature maps. Another pioneering HBA work is presented in [16] where the authors address the problem of learning and recognizing human activities of daily living by introducing the Switching Hidden Semi-Markov Model (S-HSMM), a two-layered extension of the hidden semi-Markov model (HSMM) for the modeling task. The aim of the HBA framework presented in [17] is to classify the speed of moving objects as normal or abnormal in order to detect anomalous events, taking into account the object class and spatio-temporal information such as locations and movements; this approach is based on a fuzzy knowledge base automatically generated through a learning algorithms based on a 3D description of the environment in which the system is installed. The research work presented by [18] deals with the idea of jointly modeling simple and complex behaviors to report local and global human activities in natural scenes; this system uses a state machine approach for activity representation incorporating knowledge about the problem domain in order to provide the systems with additional context-aware capabilities. In [19] the authors decompose a complex be-

havior pattern according to its temporal characteristics or spatial-temporal visual contexts. The decomposed behavior is then modelled using a cascade of Dynamic Bayesian Networks (CasDBNs). The work presented by [20] introduces a bottom-up approach for human behavior understanding based on multi-camera system. The proposed methodology, given a training set of normal data only, classifies behavior as normal or abnormal,using two different criteria of human behavior abnormality (short-term behavior and trajectory of a person). Within this system a one-class support vector machine decides short-term behavior abnormality, while we propose a methodology that lets a continuous Hidden Markov Model function as an one-class classifier for trajectories. In [21] the authors try to achieve knowledge discovery of people activity and also extract the relationship between the people and contextual objects in the scene by using the agglomerative hierarchal clustering to find the main trajectory patterns of people and relational analysis clustering to extract the relationship between people, contextual objects and events. In [22] the authors propose a suspicious behavior detection system, based on a context space model, a data stream clustering algorithm, and an inference algorithm, that makes more accurate detections, especially of those behaviors which are only suspicious in some contexts while being normal in the others.

The ADVISOR project [23] exploits a description logic language for trying to detect anomalous situations in complex scenes opportunely represented by means of three-dimensional models. The OBSERVER [24] project performs the detection of unusual elementary events and, on the basis of these events, it is able to predict the occurrence of complex abnormal behaviors in public space; this method uses a behavioral model generated through N-Trees classifiers that require a considerable computational effort for constructing a fixed, not further adjustable collection of behavioral patterns. The authors in [25] introduced the concept of *cognitive surveillance*, an ontology-based methodology that guides the identification of critical situations by using knowledge bases generated by exploiting a-priori human knowledge.

Different from previous approaches, the neuro-fuzzy nature of the proposed HBA method enables a high level of robustness respect to the intrinsic imprecisions that characterize the tracking algorithms, yielding better accuracy in human behavior recognition; moreover, the proposed system performs a multi-class behavior identification instead of the conventional two classes recognition (normal/abnormal) accomplished by previous approaches.

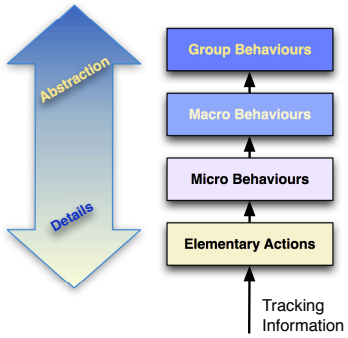


Figure 1: Behavioral Taxonomy.

3. A Hybrid Neuro-Fuzzy Approach for Human Behavioral Analysis

Our proposal for human behavioral analysis is based on a bottom-up computational intelligence hierarchical architecture whose different layers are devoted to discover the various components making up a complex human behavior and aggregate them in order to compute a complete description of the activity performed by the human. These layers work together to identify the behaviors of a collection of human beings populating a scene in a given temporal window, frame by frame, translating the raw kinematic data provided by a tracking algorithm [8] to a collection of semantic labels useful for describing both primitive and refined human behaviors. The mapping between labels and human activities is defined by introducing a so-called *behavioral taxonomy*, i.e. a hierarchical structure useful for identifying the different components characterizing a given behavior and how these components interact with the application scenario (see Fig. 1).

The behavioral taxonomy specifies human behaviors in a well-defined and structured way by depicting a collection of *behavioral components* whose detection and aggregation can enable a full scene understanding and characterization. In particular, the behavioral taxonomy is a layered structure (see Fig. 1): the lower layers manage behavioral components dealing with primitive behaviors (e.g. walking, running, etc.) that cannot be fully interpreted if not analyzed in a proper context; the higher layers refine primitive behaviors with contextual information in order to improve the human behavior interpretability (e.g. following, reaching, stealing, etc.). In detail, the behavioral taxonomy is a composition of the following behavioral component classes:

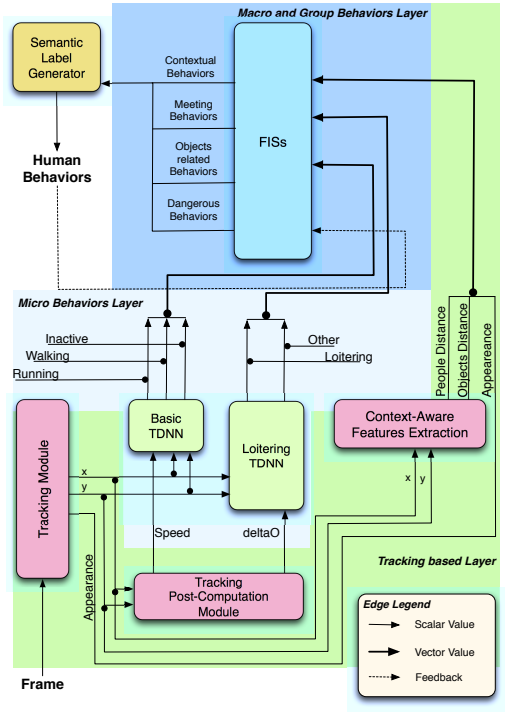


Figure 2: The Neuro-Fuzzy Human Behavioral Analysis Architecture.

- Elementary actions;
- Micro-behaviors;
- Macro-behaviors;
- Group behaviors.

Elementary actions represent the set of human instantaneous movements that are not capable of providing some information about human behaviors if they are not analyzed in an opportune spatio-temporal context (e.g. a walking step); *micro-behaviors* are obtained by sequencing a set of elementary actions in order to derive complex and repetitive actions but characterized by no semantics (e.g. walking); *macro-behaviors* refine micro-behaviors by adding contextual information and, as a consequence, they provide human behaviors with appropriate semantics (e.g. walking towards a significant object in a scene); macro-behaviors of two or more subjects can be simulta-

neously analyzed in order to recognize *group behaviors* (e.g. walking together towards a significant object in a scene).

Besides representing a suitable approach for formally modeling human activities, the behavioral taxonomy can be used as a reference model for developing human behavioral analysis systems. In fact, these systems could be based on a collection of rules able to analyze the behavioral taxonomy in a bottom up way from the human elementary actions to the human macro-behaviors and group behaviors. These collection of rules is named *behavioral semantics*. Our proposal defines its behavioral semantics by using a layered neuro-fuzzy approach (see Fig. 2) composed of the following components: *Tracking module*, *Post-Tracking module*, *Context-Aware Features Extraction*, *TDNNs Module*, *FISs Module* and *Semantic Label Generator*.

Our approach attempts to perform the behavioral identification by implementing a collection of *behavioral semantic rules* that *climb* the behavioral taxonomy and, for each level of this structure, compute the set of *behavioral semantic labels* better describing the behavior of humans in the scene. In order to achieve this aim, the behavioral semantics rules identify a human behavior by analyzing both *temporal* and *contextual* features or, in other words, they analyze how a human activity evolves in the time and how this activity is related to the context surrounding the human. The framework uses the theory of TDNNs and fuzzy logic for respectively defining the temporal and contextual rules. Hereafter the details related to the tracking algorithm, the behavioral taxonomy, and the neuro-fuzzy approach for recognizing human behaviors will be provided.

3.1. From pixels to physical and contextual information: the Tracking-based Layer

The main aim of the lower layer of the proposed architecture is to translate the pixels-based data computed by a tracking algorithm to physical and contextual information representing a kind of refined geometrical data on which the neural networks and fuzzy systems will better perform their behavioral analysis. This translation is performed through three subsystems: tracking module, post-tracking computation module and context-aware features extraction.

3.1.1. The Tracking Module

The tracking module aims at extracting the trajectory of the moving objects populating the scene (see Fig. 3). In particular, for a given time

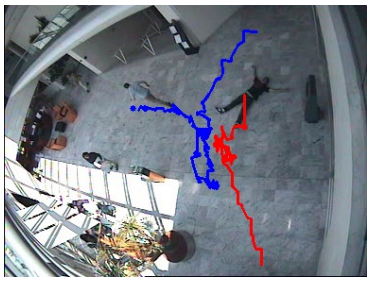


Figure 3: An example of application of the tracking algorithm to the CAVIAR dataset.

instant t, it is able to compute, for each moving object o in the scene, its position (x_o^t, y_o^t) and its $appearance_o$, that represents the class where the object belongs to (e.g. human, baggage, animal, and so on).

Although our framework can potentially work with any kind of tracking algorithm, in our experiments we use the algorithm recently proposed by [9], able to deal with complex occlusions involving a plurality of moving objects simultaneously. The rationale is grounded on a suitable representation and exploitation of the recent history of each single moving object being tracked. The object history is encoded using a state, and the transitions among the states are described through a Finite State Automata. This is the way for basing the tracking decisions not only on the information present in the current frame, but also on conditions that have been observed more stably over a longer time span. The object history can be used to reliably discern the occurrence of the most common problems affecting object detection, making this method particularly robust in complex scenarios. Since the tracking module is out of the scope of this paper, the reader can refer to [9] for a detailed description.

3.1.2. The Post-Tracking Computation Module

The post-tracking computation module uses a subset of tracking module's outputs in order to compute two additional physical measures for each object o identified by the tracking algorithm and, precisely, its speed and its variation of direction respect to the previous frame. In particular, this module computes the speed of an object at time t by considering the position of the

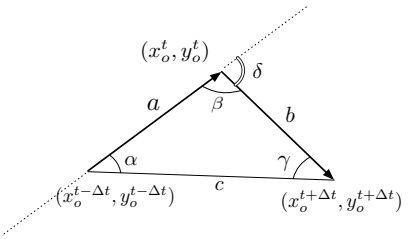


Figure 4: Computing δ by Carnot Theorem.

object at the current frame (x_o^t, y_o^t) and at previous frame $(x_o^{t-\Delta t}, y_o^{t-\Delta t})$:

$$speed_o = \frac{\sqrt{(x_0^t - x_0^{t-\Delta t})^2 + (y_0^t - y_0^{t-\Delta t})^2}}{\Delta t}$$

where the numerator represents the conventional euclidean distance between (x_o^t, y_o^t) and $(x_o^{t-\Delta t}, y_o^{t-\Delta t})$, and Δt is the frame period of the scene under analysis; this values is typically set to $1/25 sec.$, the common frame period in a video.

At the same time, this module computes the variation of direction δ of the object o located at (x_o^t, y_o^t) at time t by taking into account its previous and future locations, respectively $(x_o^{t-\Delta t}, y_o^{t-\Delta t})$ and $(x_o^{t+\Delta t}, y_o^{t+\Delta t})$. These information are used for making up the triangle shown in Fig. 4. Successively the Carnot's Theorem is used for computing the value of the angle β:

$$\beta = \arccos(\frac{a^2 + c^2 - b^2}{2ac})$$

where:

$$a = \sqrt{(x_o^t - x_o^{t-\Delta t})^2 + (y_o^t - y_o^{t-\Delta t})^2}$$

$$b = \sqrt{(x_o^{t+\Delta t} - x_o^t)^2 + (y_o^{t+\Delta t} - y_o^t)^2}$$

$$c = \sqrt{(x_o^{t+\Delta t} - x_o^{t-\Delta t})^2 + (y_o^{t+\Delta t} - y_o^{t-\Delta t})^2}$$

and, finally:

$$\delta_o = \pi - \beta$$

The value (x_o, y_o), $speed_o$ and δ_o will be used, together with the information computed by the Context-Aware Features Extraction Module, as inputs for a couple of TDNNs capable of mapping these data to a set of opportune human micro-behaviors.

3.1.3. The Context-Aware Features Extraction Module

The context-aware features extraction module is aimed at identifying some relationships between the collection of objects computed by the tracking algorithm and the context (the real environment) where these objects move. In particular, this subsystem computes a collection of values $d^h_{o_1,o_2}$ representing the euclidean distances between all pairs of "human objects" o_1 and o_2 identified by the tracking module. Moreover, this module computes a set of values $d^c_{o_1,o_2}$ for each pair of objects $o1$ and o_2 where o_1 is a human identified by the tracking module and o_2 is a so called *contextual object* as, for example an ATM in a bank scenario or an elevator in a hotel hall. The computation of the values $d^c_{o_1,o_2}$ is performed by providing the context-aware features extraction module with a XML description of the environment in which our architecture works. This description contains geometrical information about objects location. For example:

```
<Context name=‘‘Bank Hall’’>
...
        <Object name=‘‘ATM’’
                x=529
                y=340>
                <Description>
                Automatic system for
                money withdrawn
                located at ...
                <\Description>
        <\Object>
...
<\Context>
```

The grammar of the XML language used in the Context-Aware Features Extraction Module has been designed by means of the XML Schema tool [26].

3.2. Identifying Micro-Behaviors: the TDNN-based Layer

This layer performs a pattern recognition task to identify human micro behaviors starting from the collection of trajectory data returned by the lower layer. Precisely, this layer analyzes a sequence of information computed by the tracking module, namely speed, position and variation of direction of a moving object o, and maps them on the set of micro-behaviors defined in the behavioral taxonomy.

Conventional neural networks have been successfully used in different pattern recognition scenarios but, in spite of their extensive utilization, they lack of the temporal concept that characterize several applications, mainly in the fields of video analysis. For this reason, our architecture exploits a "time-oriented" neural approach, the TDNNs, capable of capturing the dynamic evolution of a given trajectory and opportunely classify it as a micro behavior. From this point of view, a TDNN can strongly improve the performance of a conventional neural network by enabling a learning mechanism based on the current data and the previous history of events. In fact, TDNNs are able for their nature to represent relationships between events in time, as well as to be invariant with respect to translations in time [27][28]. Thanks to these features, TDNNs can dynamically recognize complex trajectories as soon as a segment of sufficient length is available to the network.

In order to maximize the performance of TDNNs working at this layer, the collection of micro behaviors defined by the behavioral taxonomy has been divided in two different classes: the *speed-based micro behaviors* and *loitering micro behaviors*. Speed-based micro-behaviors represents the collection of behaviors depending upon the speed of the object under analysis, e.g. walking, running, etc.. In the same way, loitering micro-behaviors correspond to those behaviors whose identification depends upon rapid and abrupt changes of direction on the object. As a consequence, the layer is composed of two TDNNs, the *Basic TDNN* devoted to identify the speed-based micro behaviors, and the *Loitering TDNN* devoted to recognize the loitering micro behaviors. The main difference between these two networks is related to the input values they consider for performing the trajectory recognition; the Basic TDNN uses sequences of triple $(x_o, y_o, speed_o)$, i.e., positions and speeds of the object o in a given temporal window; the Loitering TDNN uses temporal sequences of triple (x_o, y_o, δ), i.e., locations and variation of direction in the same temporal window. Both networks need to be trained before they are used in a real scenario by providing them with real trajectories captured in the environment in which our HBA system will be installed. In particular,

both the networks will be trained by using two collection of trajectories, T_B for the Basic TDNN and T_L for the Loitering TDNN (both opportunely split in training, testing and validation set). In order to increase the classification effectiveness of the Loitering TDNN, the collection of real trajectories T_L is expanded by considering additional *dummy trajectories* obtained by applying a random-based translation and rotation to each trajectory in T_L. This approach can be applied only to the Loitering TDNN since a sequence of translation and rotation does not change the real nature of the initial trajectories; the resulting trajectory will be always characterized by a rapid and abrupt variation of direction (δ). The same approach cannot be applied to the Basic TDNN; indeed, a translation applied to a walking or running trajectory could significantly change the nature of this trajectory depending upon the location where the camera capturing the human movements is installed.

In order to more formally describe how the TDNN-based layer works, the schematic architecture of the Basic TDNN is shown in Fig. 5. The input vector is composed by 3 features, horizontal position, vertical position and speed corresponding, respectively, to the input channel 1, 2 and 3. Two hidden layers are considered, each composed by 15 neurons. Finally, the output layer is composed by 3 neurons, one for each considered micro behavior: running, walking and stopping. The delay is set to 20 frames. each input channel i to the TDNN is given by using a single line which is subject to delays by means of various time buffers. We named this kind of neural connections *timed delayed synapses* and they are used to connect the delayed inputs to the neurons of the second layers and, again, each neuron belonging to the second layer will be connected to the next layer through the same time delayed approach. In particular, the neurons at the second layer receives the delayed input related to a given channel by considering a proper temporal window Γ whose length is depending upon the application under design; in our case the length of Γ is 20 seconds. This approach is used for each hidden layer composing the TDNN. At the end, the output related to a given micro behaviors (running, walking, stopping) is computed by summing all activation values of previous layers stored over a period of time lasting Γ, multiplied by their connection weights. More formally, the activation level for each neuron in our Basic TDNN is, for a given trajectory γ [29]:

$$a_j^{\gamma}(t) = f(S_j^{\gamma}(t))$$

where

$$S_j^\gamma(t) = \sum_{i=1}^{3} \sum_{l=1}^{20} w_{j,i,k} \cdot a_i^\gamma(t - l\tau) + w_{j,0}$$

with $f(x) = \frac{1}{1+e^{-x}}$.

The Basic TDNN is trained by means of the conventional error back propagation paradigm with momentum. For instance, let the neuron r in output layer be the unit corresponding the running micro behavior, then its error value is:

$$\delta_r^\gamma = (T_r^\gamma - a_r^\gamma(t)) \cdot f'(S_r^\gamma(t))$$

where T_r^γ is the target value for the neuron r respect to the trajectory γ. At the same way, let j be a neuron in the hidden layers, this learning approach considers the weighted sum of the values δ related to each neuron that receive output from the unit j and computes the following error values:

$$\delta_j^\gamma(t) = \left[\sum_{l=1}^{20} \sum_{m=1}^{15} \delta_m^\gamma(t) w_{m,j,k} \right] \cdot f'(S_j^\gamma(t))$$

where m and k are, respectively, the neurons at the next layer and the delay element. Starting from these values, the adjustment of the connection weights is performed through a generalized delta rule where the moment term is employed to speed up the convergence:

$$\Delta w_{j,i,l}(t) = \eta \delta_{j,k}^\gamma(t) \cdot a_i^\gamma(t - l\tau) + \alpha \Delta w_{j,i,l}(t-1)$$

The learning phase continues until performance, measured by using root-mean-squared error, becomes satisfactory. The same design approach has been used for the Loitering TDNN.

3.3. Identifying Macro and Group Behaviours: The Fuzzy-based Layer

The tracking based layer and the micro behavior layer jointly work for providing our architecture with a quantitative classification capability translating the pixel data captured by the tracking algorithm in a collection of behavioral and contextual information such as micro behaviors, people distances, object distances and appearance. The main aim of the Fuzzy-based layer is to enhance our architecture with a qualitative skill that performs a data fusion task on the set of data coming from lower layer in order to

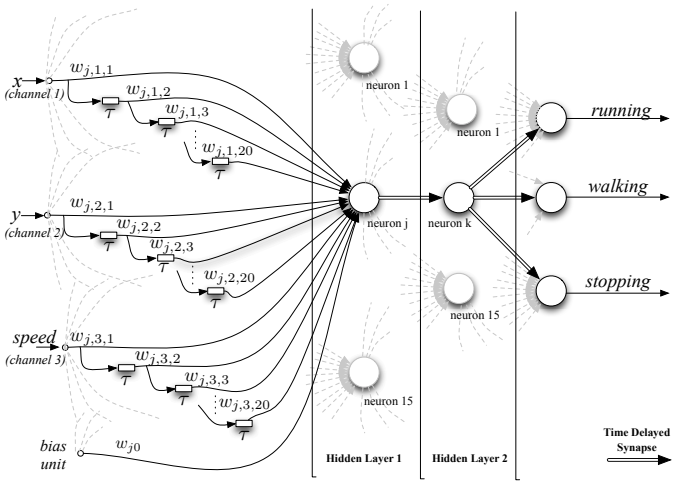


Figure 5: The Basic TDNN used for translating kinematic data to micro behaviors: running, stopping and walking.

detect the set of macro and group behavior defined by our behavioral taxonomy. In this scenario, fuzzy logic can represent the most methodology choice for achieving the aforementioned objective. Indeed, thanks to its ability in modeling uncertainty and vagueness, fuzzy logic allows for improving our architecture by means of a collection of fuzzy inference systems (FISs) whose linguistic rules are devoted to appropriately manage the set of imprecise information coming from the lower layers and, as a consequence, to show high level of accuracy in the identification of macro and group behaviors. The design of this architectural module has been performed by modeling each concept related to the human behavior identification by means of a proper fuzzy approach and an appropriate inference engine. In detail each concept (micro behaviors, macro behaviors, objects distances and appearances) has been modeled by using a collection of trapezoidal fuzzy sets, where each set has been defined on a given tolerance interval $[a, b]$ as follows (see Fig. 6):

$$A(x) = \begin{cases} 1 - \frac{a-x}{\alpha} & \text{if } a - \alpha \leq x \leq \alpha \\ 1 & \text{if } a \leq x \leq b \\ 1 - \frac{x-b}{\beta} & \text{if } b \leq x \leq b + \beta \\ 0 & \text{otherwise} \end{cases}$$

Besides enabling the HBA framework to recognize behaviors in a more efficient way than other kind of fuzzy sets, the trapezoidal fuzzy membership

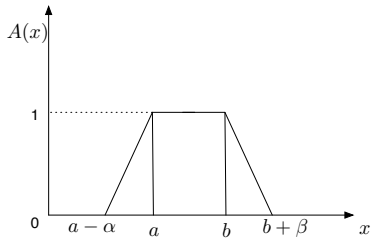


Figure 6: A Trapezoidal Fuzzy Set on tolerance interval [a, b] used for modeling HBA concepts.

can be implemented in a more simple way than other conventional shapes (e.g.Gaussian) on hardware with limited capabilities such as the embedded architectures where intelligent video surveillance systems are typically installed.

Once depicted how to "fuzzify" the collection of information coming from the lower layers of our architecture, it is necessary to make some design choices related to the implementation of different fuzzy engines composing our HBA system. Precisely, after some experimental investigations, a set of Mamdani fuzzy rule-based systems with feedback has been selected as main component of our HBA system's top layer [30]. The feedback is necessary to evaluate the current human behavior by taking into account the previous behavioral identifications and, consequently, avoiding that an "instantaneous" noisy evaluation can bring the whole system to a wrong state; this design choice improves the robustness of our framework. The conventional Mamdani approach has been preferred to the Takagi-Sugeno-Kang (TSK) thanks to its linguistic capabilities that, in spite of the higher precision shown by TSK models, enable a simple and direct writing of the rules involved in the recognition of a given behavior. This design choice enable our system to be enough scalable to deal with additional and not planned in advance human behaviors. Each Mamdani system, belonging to the HBA architecture, uses a typical *MaxMin* inference engine to compute its fuzzy output values and a common centre of gravity defuzzification operator is used to translate the fuzzy output in a real human behavior. Both the inference and defuzzification operators has been selected after some experimental investigation involving different fuzzy operators. In the current configuration, the fuzzy systems used in our HBA architecture are: *Contextual FIS*, *Meeting FIS*, *Left Object FIS* and *Danger FIS*.

The Contextual FIS analyzes data as object distance and the micro behavior values computed by the lower layers for identifying human macro behaviors that are strongly related to the context in which he/she is performing his/her activities (e.g. human o_1 is using an ATM). Some examples of fuzzy concepts related to this FIS are shown in Fig. 7 and 8, whereas, some contextual rules are:

$$\begin{aligned} &\text{IF (} distance_{o1,ATM} \text{ is } small\text{) AND (} stopping_1 \text{ is } yes \text{)} \\ &\text{AND (} stopping_2 \text{ is } yes \text{) THEN (} output \text{ is } moneyWithdrawn \text{).} \end{aligned} \tag{1}$$

The Meeting FIS inspects data like people distances and the micro behavior values computed by the TDNNs for identifying human macro behaviors depending upon micro behaviors related to two or more persons (e.g. meeting). Some examples of fuzzy concepts related to this FIS are shown in Figs. 7 and 8, whereas, some contextual rules are:

$$\begin{aligned} &\text{IF (} distance_{o1,o2} \text{ is } small \text{) AND (} stopping_1 \text{ is } yes \text{)} \\ &\text{AND (} stopping_2 \text{ is } yes \text{) THEN (} output \text{ is } meeting \text{).} \end{aligned} \tag{2}$$

The Left Object FIS analyzes data like object distance and the micro behavior values computed by the TDNNs for identifying human macro behaviors generated by a person leaving an object in the scene. Examples of fuzzy concepts related to this FIS are designed in the same way of those shown in Figs. 7 and 8, whereas, some contextual rules (o_1 is a human and o_2 is an inanimate object) are:

$$\begin{aligned} &\text{IF (} distance_{o1,o2} \text{ is not } small \text{) AND (} stopping_1 \text{ is not } yes \text{)} \\ &\text{AND (} stopping_2 \text{ is } yes \text{) AND (} appearance_2 \text{ is } bag \text{)} \\ &\text{THEN (} output \text{ is } leavingObject \text{).} \end{aligned} \tag{3}$$

The Danger FIS analyzes data like people distance and the micro behavior values computed by the TDNNs for identifying human macro behaviors related to possible dangerous situations (e.g. fighting between two humans). Examples of fuzzy concepts related to this FIS are designed in the same way of those shown in Figs. 7 and 8, whereas, some contextual rules (both o_1 and o_2 are humans) are:

$$\begin{aligned} &\text{IF (} distance_{o1,o2} \text{ is } small \text{) AND (} loitering_1 \text{ is } yes \text{)} \\ &\text{AND (} loitering_2 \text{ is } yes \text{) THEN (} output \text{ is } fighting \text{).} \end{aligned} \tag{4}$$

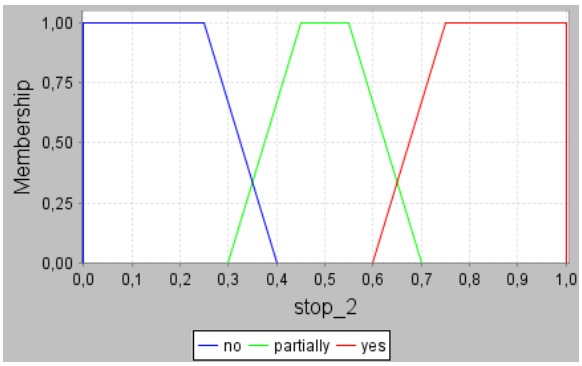


Figure 7: The fuzzy concept related to the trajectory class Stopping.

As shown by the architectural outline, each FIS can use, as one of its inputs, the last output returned by the system; in order to realize this feedback feature, we defined a set of fuzzy variables which identify the human behavior computed by the system at the previous time instant. Some examples of fuzzy variables encoding the state of the system are *PreviousBehaviorFighting* and *PreviousBehaviorLeavingObject*. This design choice allows FISs to perform the behavioral analysis not through a one shot evaluation but, vice versa, by taking into account the current situation and the last behavior computed for the same human. An example is shown below:

$$\begin{aligned} &\text{IF (} PreviousBehaviorFighting \text{ is } yes \text{) AND (} stopping_1 \text{ is } yes \text{)} \\ &\text{AND (} running_2 \text{ is } yes \text{) THEN (} output \text{ is } fighting \text{).} \end{aligned} \tag{5}$$

Each FIS computes a collection of macro behaviors by applying a defuzzification operator on the fuzzy set returned by its fuzzy rules. Successively, these collections of values are analyzed by a so called Semantic Label Generator that computes the maximum membership value and returns a semantic label corresponding to it. In order to make our system robust and tolerant to eventual FIS errors, the labels generated by the semantic label generator are "temporally" analyzed by the last module of our hierarchy: Temporal Behavioral Analysis. This module considers the behavioral labels computed during the last second and returns, as output of the whole systems, the behavior corresponding to the most frequent label.

As will be shown in the next section, the feedback feature and the temporal analysis strongly improve the accuracy of the proposed approach.

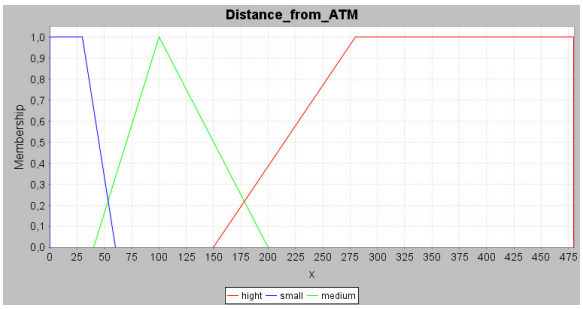


Figure 8: The fuzzy concept related to the context variable Distance from ATM.

4. Experimental results

The proposed method has been evaluated on a standard dataset, the first section of the Caviar (Benchmark Data for PETS-ECCV 2004) [31]. The twenty-eight videos composing the database have been acquired from a wide angle camera in the entrance lobby of the INRIA Labs at Grenoble, France. Six different scenarios occur: walking, browsing, collapsing, leaving object, meeting and fighting. A more detailed description is provided in Table 1. Since in this paper we focus on behavioral analysis, we do not extract moving objects trajectories by using our tracking method, but we use the ground truth provided with the dataset. This choice allows us to compare the proposed approach with other state of the art methods in a proper way, by starting from the same input data.

In the following, three different evaluations will be provided. In the first (Subsection 4.1) we provide a quantitative evaluation of the TDNN modules, in charge of recovering micro-behaviors. In Subsection 4.2, we evaluate the performance of the entire proposed framework and a comparison with other state of the art approaches is carried out. In particular, the methods proposed by [20] and [22] have been considered, since they evaluate their systems on the CAVIAR dataset, rather than a proprietary dataset, and the results are publicly available. It is worth noting that in the Caviar dataset there is not an explicit definition of *normal* and *abnormal*. For this reason, in order to achieve a proper comparison, the different scenarios, and then the different macro-behavior that we consider in our system, have been divided into two classes according to their semantic meaning. In particular, as in [20] and [22], the fighting scenarios are considered *abnormal* while all the other ones

Scenario	Number of Sequences	Number of Frames
Walking	3	3045
Browsing	6	6665
Collapsing	4	4227
Leaving object	5	5848
Meeting	6	4135
Fighting	4	2499

Table 1: Benchmark Data for PETS-ECCV 2004.

normal.

Finally, in Subsection 4.3, we focus on a single abnormal scenario, in order to clarify to the reader the overall behavior of the proposed method. In particular, a fight scenario is considered.

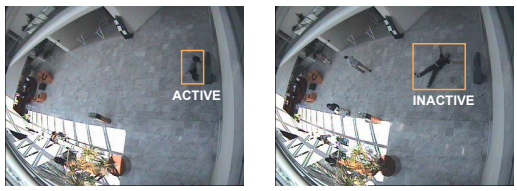
An example of the proposed method at work over the CAVIAR dataset is available at https://www.dropbox.com/s/0otpe9q0uwfr50a/HBA.avi.

4.1. Micro-behaviors evaluation

The Caviar dataset identifies the following four different classes:

- *inactive*: the person is visible but not moving;
- *active*: the person is making movements but he is not translating across the image;
- *walking*: the person is visible, moving and translating across the image slowly;
- *running*: the person is visible, moving, translating across the image quickly.

Usually the main problem arises in distinguishing *active* and *inactive* classes, since, as it happens in real scenarios, detailed information related to the pose of people are not available: objects are in a far-field or video has a low-resolution, and the only information that a video analytic system is reliably able to extract is a noisy trajectory. For such a reason, we decided to merge the classes *inactive* and *active* in a single micro-behavior *stop*. An example is shown in Fig. 9.



(a) (b)

Figure 9: Example of active (a) and inactive (b) objects

This choice is also confirmed by the ambiguity in labeling these behaviors in multiple observers. In [32] it has been shown that the overlap in the classification of the class *inactive* between three different observers is only 68 %, confirming the difficulty, also for a human observer, to identify and correctly classify a behavior.

In order to evaluate the performance of the TDNN, the Caviar dataset has been partitioned into two main folds, as shown in Table 2.

Furthermore, the training set has been partially extended by duplicating the *loitering* trajectories, which have been properly shifted and scaled. As a matter of fact, in [29] it is shown that this simple tuning allows a generic TDNN to obtain approximate shift-invariance, which is a desired property for the *loitering* class.

The entire system has been realized in MATLAB (R2011b) and the Neural Network and Fuzzy Logic Toolbox have been used respectively for the TDNNs and for the fuzzy engines. The realized basic TDNN is composed by two hidden layers, each with 15 neurons, and one output layer with 3 neurons. The input window has been set to 20 frames (approximatively 1 second). The loitering TDNN has the same structure, excepts for the fact that the output layer is composed by 2 neurons. The training set was presented to the networks until the minimum error was reached.

For each class $c_i \in \{\text{stopping, walking, running, loitering}\}$, the following metrics have been computed in order to quantitatively evaluate the performance of the method: the *precision*, the *recall* and the *f-score* . The former is the the fraction of retrieved instances that are relevant. It can be computed as:

$$precision_i = \frac{TP_i}{TP_i + FP_i}. \tag{6}$$

Training Set Tr
Browse2.mpg, Browse3.mpg, Browse WhileWaiting2.mpg, Fight Chase.mpg, Fight OneManDown.mpg, LeftBox.mpg, LeftBag PickedUp.mpg, Meet WalkSplit.mpg, Meet WalkTogether1.mpg, Rest SlumpOnFloor.mpg, Rest WiggleOnFloor.mpg, Split.mpg, Walk3.mpg
Test Set Ts
LeftBag.mpg, Rest InChair.mpg, Meet Crowd.mpg, Walk1.mpg, Fight RunAway2.mpg, Browse1.mpg, Browse4.mpg, Browse WhileWaiting1.mpg, Fight OneManDown.mpg, Fight RunAway1.mpg, LeftBag AtChair.mpg, LeftBag BehindChair.mpg, Meet Split 3rdGuy.mpg, Rest FallOnFloor.mpg, Walk2.mpg.

Table 2: Partition of the CAVIAR dataset into training and test sets.

The recall is the fraction of relevant instances that are retrieved and can be evaluated as:

$$recall_i = \frac{TP_i}{TP_i + FN_i}. \tag{7}$$

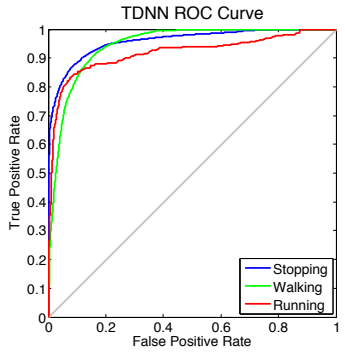
TP_i, FP_i and FN_i respectively represent the number of True Positive (the number of items correctly labeled as belonging to the class c_i), False Positive (the number of items incorrectly labeled as belonging to the class c_i) and False Negative (the number of items belonging to the class c_i but incorrectly labeled). Finally, the f-score is a measure of a test's accuracy which combines in a single value both recall and precision; it is computed as the harmonic mean of precision and recall:

$$f\text{-}score_i = 2 \cdot \frac{precision_i \cdot recall_i}{precision_i + recall_i}. \tag{8}$$

The considered metrics, computed for each class, are summarized in Table 4: the class that our system is most reliably able to recognize is the class *walking*. This result is also justified by the number of samples given to the

network for the training (13407 samples for the class *walking*, 486 samples for the class *running*, 7297 samples for the class *stopping* and 532 samples for the class *loitering*).

A more complete evaluation of the proposed approach is presented in Table 3, where the Receiver operating characteristic (ROC) curve and the misclassification matrix are reported, the last one obtained by optimizing the performance on the training set.



(a)

		Predicted Class		
		stop	***walking***	***running***
GT	***stop***	**85.87**%	2.45%	0.19%
	walking	13.55%	**96.44**%	12.53%
	running	0.58%	1.11%	**87.28**%

(b)

Table 3: Performance of the proposed *micro-behavior* detector, both in terms of ROC curve and Misclassification Matrix.

Finally, in order to further confirm the effectiveness of the proposed method, two different comparisons have been carried out. First, the traditional speed-based approach has been considered: the real velocities (expressed in meters/seconds) are computed thanks to a preliminary calibration of the scene and a GMM classifier is used for recognizing the different classes. On the other side, the method proposed by [33] has been taken into account. The results are reported in Fig. 5(a) and Fig. 5(b) respectively.

Although a punctual comparison is not possible since in [33] both the class *active* and *inactive* are considered, we can note that a significant improvement of the performance, especially in terms of *running* recognition. This consideration is also confirmed by analyzing the results obtained by traditional speed-based approach, which confirms not to be robust with respect

	Precision	***Recall***	***f-score***
stop	0.9471	0.8587	0.9007
walking	0.9058	0.9645	0.9342
running	0.9367	0.8728	0.9036
loitering	0.8345	0.9419	0.8850

Table 4: Precision, Recall and f-score of the proposed *micro-behavior* detector.

		Predicted Class		
		stop	***walking***	***running***
GT	***stop***	**78.13**%	21.65%	0.22%
	walking	10.82%	**87.67**%	1.51%
	running	2.75%	84.99%	**12.26**%

(a)

		Predicted Class					
		active	***inactive***	***walking***	***running***	***multiple***	***none***
GT	***active***	**37%**	15%	27%	0%	10%	11%
	inactive	7%	**78%**	3%	0%	8%	4%
	walking	3%	2%	**90%**	0%	2%	3%
	running	0%	0%	53%	**45%**	0%	2%

(b)

Table 5: Misclassification Matrix of the speed-based *micro-behaviors* detector (a) and of the method proposed in [33].

to errors of the tracking phase.

4.2. Macro-behaviors Evaluation

A deeper evaluation has been conducted over the entire architecture in order to confirm the efficiency of the proposed method. Two different experimentations has been carried out: first, the misclassification matrix has been computed in order to evaluate the ability of the system in distinguishing the different macro-behaviors (staying-together ST, money-withdrawal MW, left-baggage LB, fighting F, none N). Furthermore, a comparison with other state-of-the-art methods using the standard Caviar dataset has been performed, confirming the usability of the proposed method in real applications.

The confusion matrix computed over the test set Ts, in terms of frames, is shown in Table 6a. The ground truth has been manually made by a human expertise without any knowledge about the proposed method, so avoiding to influence his decisions. However, it is worth pointing out that it is not

a simple task to define the exact frame when two persons start fighting or when they stop fighting, also for a human operator. For this reason, although the performance achieved in terms of frames by the proposed method are convincing, a further experimentation is needed: rather than reasoning with frames, we prefer to focus on events. A generic event has been considered correctly detected if there is at least a 50% overlap between the detected frames and the ground truth ones. The obtained confusion matrix is shown in Table 6b: all the events have been correctly classified, except a *fighting* event, wrongly associated to a *staying-together* behavior. This particular error is due to the fact that the TDNN modules show, for both the involved persons, high values of *loitering* and *stopping*; it means that the output of the Fuzzy Engine exhibits high values for both macro-behavior classes and the Semantic Label Generator (wrongly) chose the highest one.

Furthermore, it is worth noting that a single frame is associated to the event *none* whereas there are not any events of interest occurring at the same time instant. This is the reason why it does not make sense to signal an event *none* to the human operator.

Finally, the effectiveness of the proposed method has been confirmed by comparing it with two other state of the art approaches [20][22] which use the same standard Caviar dataset. The results are summarized in Table 7. Note that a lot of methods recently proposed perform the test over proprietary datasets, not freely available, making these systems not easily comparable.

The main issue to manage lies in the fact that [20] and [22] only detect if a behavior is *normal* or *abnormal* behaviors, without explicitly identifying the particular behavior. In more details, the fighting scenarios are considered abnormal, while all the other ones are considered normal.

A comparison in terms of precision, recall and f-score is shown in Table 7: in particular, the second row shows the results obtained by applying the method proposed in [20] when the entire trajectory is available (offline), while third row summarizes the results computed in real time, that is by informing the human operator at the i-th frame about the event occurring at the i-th frame. Although the proposed method works in real time, we can note that it clearly outperforms [20] in both its applications, real time and offline.

Furthermore, the fourth row shows the results obtained by [22]: the proposed method outperforms it in terms of f-score over the detection of *normal* behaviors. However, it is worth considering that the proposed method is also able to identify the particular behavior occurring, which is an important and not negligible feature in real applications, where, for instance, the macro-

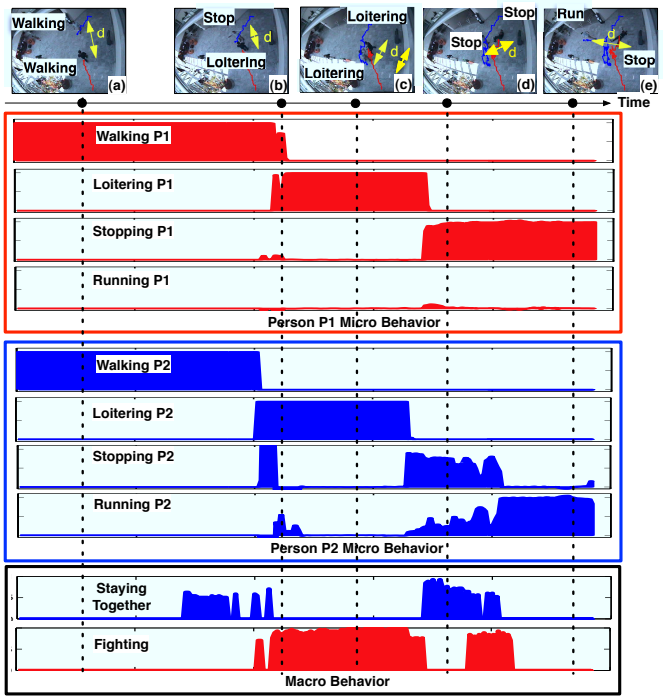


Figure 10: Output of the proposed method on a fighting sequence of the Caviar dataset.

behavior *left-baggage* could be considered an *abnormal* behavior rather than a *normal* one.

In light of the obtained results, we can state that the proposed hierarchical architecture is able to reliably identify and label the different components of complex behaviors.

4.3. Case study: a fight scenario

This Section is devoted to clarify the proposed architecture through the description of a simple example. In particular, a fighting scenario is considered. The behavioral evolution of the system in the above mentioned example, both in terms of micro-behaviors of each person populating the scene and of macro-behavior, is detailed in Fig. 10.

In the first row of Fig. 10 five different significant time instants have been selected: two persons walk, maintaining a significant distance between each other (a); they meet, stop talking (b) and after a few seconds they start

		Predicted Class				
		MW	***ST***	***F***	***LB***	***N***
GT	***MW***	**139**	0	0	0	40
	ST	0	**2586**	0	0	106
	F	0	103	**241**	0	15
	LB	0	0	0	**871**	10
	N	29	479	72	12	**9046**

(a)

		Predicted Class				
		MW	***ST***	***F***	***LB***	***N***
GT	***MW***	**2**	0	0	0	0
	ST	0	**5**	0	0	0
	F	0	1	**2**	0	0
	LB	0	0	0	**2**	0
	N	0	0	0	0	**0**

(b)

Table 6: Confusion Matrix of the proposed architecture for the following behaviors: staying-together *ST*, money-withdrawal *MW*, left-baggage *LB*, fighting *F*, none *N*. The evaluation is performed in terms of frames (a) and of events (b).

fighting: the distance between them remains small and their trajectories have a very irregular shape (c). Finally, one person falls down on the floor (d) and the other one runs away (e). Each frame is labeled with the micro-behavior of the subjects, obtained by the TDNN modules, and with the distance between them, computed by the Context-Aware Features Extraction. Furthermore, the trajectories extracted until that moment overlay the image.

The second row shows the temporal evolution of the system: the first set, composed by four different bars (Walking P1, Loitering P1, Stopping P1 and Running P1), refers to the micro-behavior of the person identified by id 1, which enters the scene from the bottom; the following set, also composed by four bars (Walking P2, Loitering P2, Stopping P2 and Running P2), refers to the micro-behavior of the person entering the scene from the top and identified by id 2. Finally, the last set, composed by only two bars, identifies the macro-behaviors activated during this sequence (Staying together and

	Precision		Recall		F-Score	
	Normal	**Abnormal**	**Normal**	**Abnormal**	**Normal**	**Abnormal**
Our Proposal	0.9912	0.7699	0.9946	0.6713	0.9929	0.7172
[20] offline	0.8882	0.3129	0.775	0.5125	0.8277	0.3886
[20] real-time	0.7625	0.2273	0.7309	0.2582	0.7464	0.2418
[22]	1.0	0.6666	0.9693	1.0	0.9844	0.7999

Table 7: Precision and Recall obtained by the proposed method and by other state-of-the art techniques [20][22].

Fighting). Note that the other macro-behaviors (money-withdrawal and left-baggage) have not been shown in the figure since there are not any rules activating them in the considered example.

A more detailed description of the state of the system will be provided at the time instant (c). Starting from the object kinematic data (spatial coordinates x and y) computed by the Tracking module and from derived information (*speed* and *delta orientation*), computed by the Tracking Post-Computation Module, the TDNNs calculate trajectory memberships for each of the considered micro-behavior class. For instance, the classes membership for the first person is:

$$p_1 = \begin{bmatrix} 0 & 0.97 & 0.08 & 0 \end{bmatrix}, \tag{9}$$

where the elements of vector p_1 respectively refer to class *walking*, *loitering*, *stopping* and *running*. As for the second person, the classes membership vector obtained by the TDNNs is the following:

$$p_2 = \begin{bmatrix} 0 & 0.99 & 0 & 0.02 \end{bmatrix}. \tag{10}$$

The maximum value of each vector is used in order to infer the persons' micro-behavior: in such a situation, we can note that both the persons are *Loitering*.

The kinematic data are also processed by the Context-Aware Features Extraction in order to obtain context information. In particular, such module calculates the distance among all the objects populating the scene and the distance between each object and one or more fixed object in the scene (an ATM, on the right, in the considered example). In the situation we are analyzing, the module obtains the following results:

$$d_{p1,a} = 47 \quad pixels; \tag{11}$$

$$d_{p2,a} = 56 \quad pixels; \tag{12}$$

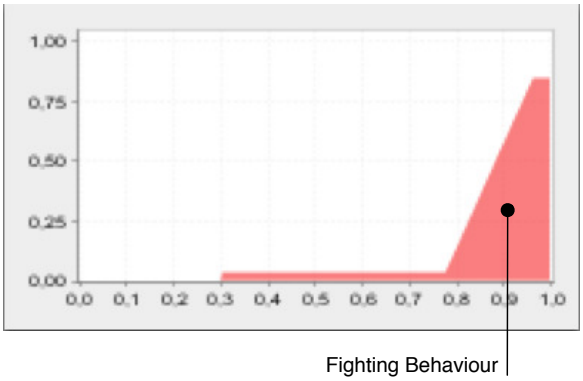


Figure 11: The fuzzy area inferred by the HBA fuzzy module.

$$d_{p1,p2} = 8 \quad pixels, \tag{13}$$

where $d_{p1,a}$ is the distance between the person 1 and the ATM, $d_{p2,a}$ is the distance between the person 2 and the ATM and finally $d_{p1,p2}$ is the distance among the two persons.

All this information will be used by our fuzzy engines in order to identify macro-behaviors or group-behaviors. In particular, it fires a collection of fuzzy rules inferring the fuzzy area, shown in Fig. 11, whose defuzzified value corresponds to the Fighting Group Behaviour (corresponding to the situation (b) reported in Fig. 10).

Successively, the Semantic Labels Generator analyzes the set of defuzzified values computed by the fuzzy engines during the last second (25 frames) and it finally identifies the Fighting Group Behavior as the most appropriate behavior for describing the scene under analysis.

5. Conclusions

The research in the field of human behavioral analysis and intelligent video surveillance systems has shown a human inadequacy to simultaneously monitor multiple sources of visual data and conceptualize the behavior of all the observed objects for quickly and correctly detecting danger situations. In this paper we reduced this drawback by proposing a behavioral taxonomy and a related neuro-fuzzy architecture aimed at supporting human beings in this stressing activity. The proposed approach detects human behaviors by means of a hierarchical analysis capable of identifying and labeling the different components of a complex behavior. Our experiments preliminarily

validated the soundness of our architecture by correctly detecting some human behaviors in a complex scene. In the future, more sophisticated tests and comparisons with other literature approaches will be conducted in order to further prove the quality of our system.

Acknowledgment

This research has been partially supported by A.I.Tech s.r.l., a spin-off company of the University of Salerno (www.aitech-solutions.eu).

References

[1] A. H. Tickner, E. C. Poulton, A. K. Copeman, D. C. V. Simmonds, Monitoring 16 television screens showing little movement, Ergonomics 15 (3) (1972) 279–291. doi:10.1080/00140137208924430.

[2] A. Amato, V. Di Lecce, V. Piuri, Neural network based video surveillance system, in: Computational Intelligence for Homeland Security and Personal Safety, 2005. CIHSPS 2005. Proceedings of the 2005 IEEE International Conference on, 2005, pp. 85 –89. doi:10.1109/CIHSPS.2005.1500617.

[3] D. Duque, H. Santos, P. Cortez, Prediction of abnormal behaviors for intelligent video surveillance systems, in: Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on, 2007, pp. 362 –367. doi:10.1109/CIDM.2007.368897.

[4] A. Oikonomopoulos, I. Patras, M. Pantic, Spatiotemporal salient points for visual recognition of human actions, IEEE Transactions on Systems, Man and Cybernetics - Part B 36 (3) (2006) 710–719.

[5] P. Dai, H. Di, L. Dong, L. Tao, G. Xu, Group interaction analysis in dynamic context, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 39 (1) (2009) 34–42. doi:10.1109/TSMCB.2008.2009559.

[6] O. Brdiczka, J. Crowley, P. Reignier, Learning situation models in a smart home, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 39 (1) (2009) 56–63. doi:10.1109/TSMCB.2008.923526.

[7] K. Huang, D. Tao, Y. Yuan, X. Li, T. Tan, View-independent behavior analysis, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 39 (4) (2009) 1028–1035. doi:10.1109/TSMCB.2008.2011815.

[8] R. Di Lascio, P. Foggia, A. Saggese, M. Vento, Tracking interacting objects in complex situations by using contextual reasoning, in: Computer Vision Theory and Applications (VISAPP), 2012 International Conference on, 2012.

[9] R. Di Lascio, P. Foggia, G. Percannella, A. Saggese, M. Vento, A real time algorithm for people tracking using contextual reasoning, Computer Vision and Image Understanding.
URL http://dx.doi.org/10.1016/j.cviu.2013.04.004

[10] Y. Lin, Affective driving, in: S. Fukuda (Ed.), Emotional Engineering, Springer London, 2011, pp. 263–274.

[11] D. Karras, On improved mri segmentation using hierarchical computational intelligence techniques and textural analysis of the discrete wavelet transform domain, in: Intelligent Signal Processing, 2007. WISP 2007. IEEE International Symposium on, 2007, pp. 1–6. doi:10.1109/WISP.2007.4447513.

[12] A. T. Lawniczak, B. N. D. Stefano, Computational intelligence based architecture for cognitive agents, Procedia Computer Science 1 (1) (2010) 2227 – 2235, {ICCS} 2010. doi:http://dx.doi.org/10.1016/j.procs.2010.04.249.
URL http://www.sciencedirect.com/science/article/pii/S1877050910002504

[13] J. Aggarwal, M. Ryoo, Human activity analysis: A review, ACM Comput. Surv. 43 (3) (2011) 16:1–16:43. doi:10.1145/1922649.1922653.
URL http://doi.acm.org/10.1145/1922649.1922653

[14] R. Poppe, A survey on vision-based human action recognition, Image and Vision Computing 28 (6) (2010) 976 – 990. doi:10.1016/j.imavis.2009.11.014.
URL http://www.sciencedirect.com/science/article/pii/S0262885609002704

[15] W. Hu, T. Tan, L. Wang, S. Maybank, A survey on visual surveillance of object motion and behaviors, IEEE Transactions on Systems, Man and Cybernetics 34 (2004) 334–352.

[16] T. V. Duong, D. Q. Phung, H. H. Bui, S. Venkatesh, Efficient duration and hierarchical modeling for human activity recognition., Artif. Intell. 173 (7-8) (2009) 830–856.

[17] J. Albusac, D. Vallejo, J. Castro-Schez, L. Jimenez-Linares, Oculus surveillance system: Fuzzy on-line speed analysis from 2d images, Expert Systems with Applications 38 (10) (2011) 12791 – 12806. doi:10.1016/j.eswa.2011.04.071.
URL http://www.sciencedirect.com/science/article/pii/S0957417411005872

[18] A. Fernández-Caballero, J. C. Castillo, J. M. Rodríguez-Sánchez, Human activity monitoring by local and global finite state machines, Expert Syst. Appl. 39 (8) (2012) 6982–6993. doi:10.1016/j.eswa.2012.01.050.
URL http://dx.doi.org/10.1016/j.eswa.2012.01.050

[19] C. C. Loy, T. Xiang, S. Gong, Detecting and discriminating behavioural anomalies, Pattern Recogn. 44 (1) (2011) 117–132. doi:10.1016/j.patcog.2010.07.023.
URL http://dx.doi.org/10.1016/j.patcog.2010.07.023

[20] P. Antonakaki, D. Kosmopoulos, S. J. Perantonis, Detecting abnormal human behaviour using multiple cameras, Signal Process. 89 (9) (2009) 1723–1738. doi:10.1016/j.sigpro.2009.03.016.
URL http://dx.doi.org/10.1016/j.sigpro.2009.03.016

[21] L. Patino, H. Benhadda, E. Corvee, F. Bremond, M. Thonnat, Extraction of activity patterns on large video recordings, Iet Computer Vision 2. doi:10.1049/iet-cvi:20070062.

[22] A. Wiliem, V. Madasu, W. Boles, P. Yarlagadda, A suspicious behaviour detection using a context space model for smart surveillance systems, Comput. Vis. Image Underst. 116 (2) (2012) 194–209. doi:10.1016/j.cviu.2011.10.001.
URL http://dx.doi.org/10.1016/j.cviu.2011.10.001

[23] N. T. Siebel, S. J. Maybank, The advisor visual surveillance system, in: in ECCV 2004 workshop Applications of Computer Vision (ACV, 2004.

[24] D. Duque, H. Santos, P. Cortez, Prediction of abnormal behaviors for intelligent video surveillance systems, in: Computational Intelligence and Data Mining, 2007. CIDM 2007. IEEE Symposium on, 2007, pp. 362 –367. doi:10.1109/CIDM.2007.368897.

[25] C. Fernández, P. Baiget, F. X. Roca, J. Gonzílez, Determining the best suited semantic events for cognitive surveillance, Expert Syst. Appl. 38 (4) (2011) 4068–4079. doi:10.1016/j.eswa.2010.09.070.
URL http://dx.doi.org/10.1016/j.eswa.2010.09.070

[26] C. S. McQueen, H. Thompson, Xml schema (2014).
URL http://www.w3.org/XML/Schema

[27] A. Yazdizadeh, K. Khorasani, Adaptive time delay neural network structures for nonlinear system identification, Neurocomputing 47 (1-4) (2002) 207–240.

[28] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, K. Lang, Phoneme recognition using time-delay neural networks, Acoustics, Speech and Signal Processing, IEEE Transactions on 37 (3) (1989) 328–339. doi:10.1109/29.21701.

[29] D.-T. Lin, J. Dayhoff, P. Ligomenides, Trajectory recognition with a time-delay neural network, in: Neural Networks, 1992. IJCNN., International Joint Conference on, Vol. 3, 1992, pp. 197 –202 vol.3. doi:10.1109/IJCNN.1992.227170.

[30] C.-S. Lee, M.-H. Wang, A fuzzy expert system for diabetes decision support application, Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on 41 (1) (2011) 139–153. doi:10.1109/TSMCB.2010.2048899.

[31] R. B. Fisher, The pets04 surveillance ground-truth data sets, in: n Proc. 6th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance, 2004.

[32] T. List, J. Bins, J. Vazquez, R. Fisher, Performance evaluating the evaluator, in: Visual Surveillance and Performance Evaluation of Tracking and Surveillance, 2005. 2nd Joint IEEE International Workshop on, 2005, pp. 129–136.

[33] A. Sridhar, A. Sowmya, P. Compton, On-line, incremental learning for real-time vision based movement recognition, in: Proceedings of the 2010 Ninth International Conference on Machine Learning and Applications, ICMLA '10, IEEE Computer Society, Washington, DC, USA, 2010, pp. 465–470. doi:10.1109/ICMLA.2010.75.
URL http://dx.doi.org/10.1109/ICMLA.2010.75