Learning and Nonlinear Models - Journal of the Brazilian Society on Computational Intelligence (SBIC), Vol. 18, Iss. 1, pp. 4-14, 2020

© Brazilian Computational Intelligence Society

Aerial Human Activity Recognition Through a Cognitive Architecture and a New Automata Proposal

Milena F. Pinto
Federal Center for Technological Education of Rio de Janeiro, Department of Electronics, Rio de Janeiro, Brazil
milenafaria@ieee.org)

Aurelio G. Melo, Andre L. M. Marcato, Camile A. Moraes
Federal University of Juiz de Fora, Electrical Department, Juiz de Fora, Brazil
{aurelio.melo, camile.moraes}@engenharia.ufjf.br, and andre.marcato@ufjf.edu.br

Abstract – Video surveillance often involves several actors in multiple interactions and modelling complex activities becomes a challenge, especially in real environments. When applying autonomous video surveillance, object recognition techniques are used to produce symbolic information related to the information present in a scene. An automaton is a specialized structure capable of accepting or rejecting those symbols producing an efficient computation structure for these types of data processing. This research work presents an innovative structure for the well-known Weighted Automata to organize the information from sensors, grouping these measurements into a symbolic representation of actions that are happening in the real world. This work also proposes a hierarchical architecture formed by a multilevel sensorial system comprised of low, middle and high levels to perceive the environment and to comprehend scenes. The proposed architecture is designed to operate in a decentralized way and onboard of the Unmanned Aerial Vehicles (UAVs). The experiments showed that this system updated effectively the semantic structure given the sequence of information and demonstrated the automaton and architecture effectiveness..

Keywords – Automata, Cognitive Systems, Human Interaction, Unmanned Aerial Vehicle.

1. INTRODUCTION

Recognition of human activities has been studied extensively mainly in computer vision and robotics areas, especially in the development of artificially intelligent systems to track human movements and to recognize their actions [1]. Besides, for surveillance of human activities, the use of cameras embedded in Unmanned Aerial Vehicles (UAVs) is becoming common in both indoor and outdoor environments, providing better perception, cost-effectiveness, and covering a larger area [2]. According to [3], a visual surveillance system works at detection, recognition, and behavior analyses. Thus, there is a need that this kind of system learns and understands scenes in an autonomous fashion. For example, several works individually analyzed the required components for those applications, such as computational vision, machine learning, among others. However, a few works have concerned with the integration and development of these architectures with the real-world. The cognitive system presented in [4] learns information about manipulation tasks performed by humans, where this information is provided by cameras and videos using high-level symbolic tools for interpreting the actions. This work has a strong representation capacity, but they do not represent time and location. In the context of automatic surveillance systems, the methodology presented in [5] and [6] is capable of representing the learning process using sensor data in association with elementary action detectors to feed a reasoning engine responsible for high-level formation. However, those works present a lack of integration and experimentation with real-world sensors.

Most of the works in the visual recognition area use learning techniques with large amounts of spatiotemporal data, such as the ones presented in [7] and [8]. Besides, several types of researches working with human activity recognition are based on learning by demonstration, remaining at the level of signal-to-signal mapping. They do not possess generalization capacity, as can be seen in [9] and [10]. On the other hand, in [11] and [12] are introduced grammatical rules for actions. However, these researches only involve object manipulation. In the context of cognitive systems architectures, some works were proposed, such as in [13], but none involving aerial application. Most of the researches using UAV for aerial surveillance is based on centralized control, where the processing is carried out in a central computer as in [7], or they are only focused on tracking strategies and not having mechanisms for understanding scenes, as in [14] and [15]. Additionally, most of them are tested only in simulation and not in real environments [16].

1.1 Main Contributions

This work presents a flexible and decentralized architectural concept to be embedded in a companion computer onboard the UAV. The examples discussed in this work will be focused on object search and suspicious human activity recognition in real-time. An automaton is proposed to organize the sensorial information and recognize actions. Additionally, to the best of the authors' knowledge, this work also presents an innovative automata approach to propagate the likelihood associated with each known instance to the high-level decision-maker. In this work, the following innovations are proposed:

- An innovative automata approach where the weights defined by an activation function are propagated through transitions.
- An organized paradigm to assign processing tasks for UAV applications.
- A validation of the propositions in the real-world scenario.

1.2 Organization

This research is organized as follows. Section 2 presents the main concepts of the proposed cognitive system. Besides, this section also describes the mathematical foundation of the proposed automata. Section 3 presents the results and a deep discussion of them. The concluding remarks and ideas of future works are conducted in Section 4.

2 PROPOSED COGNITIVE SYSTEM

The general concept of the decentralized surveillance architecture was already discussed in [17]. Figure 1 illustrates the simplified diagram of the system. A camera embedded in the UAV acquires a sequence of images. Then, the system organizes the sensory information in the middle-level through the new automata approach to produce actions in the scenes. This scene information is feed into the high-level where the Case-Based Reasoning (CBR) method takes decisions.

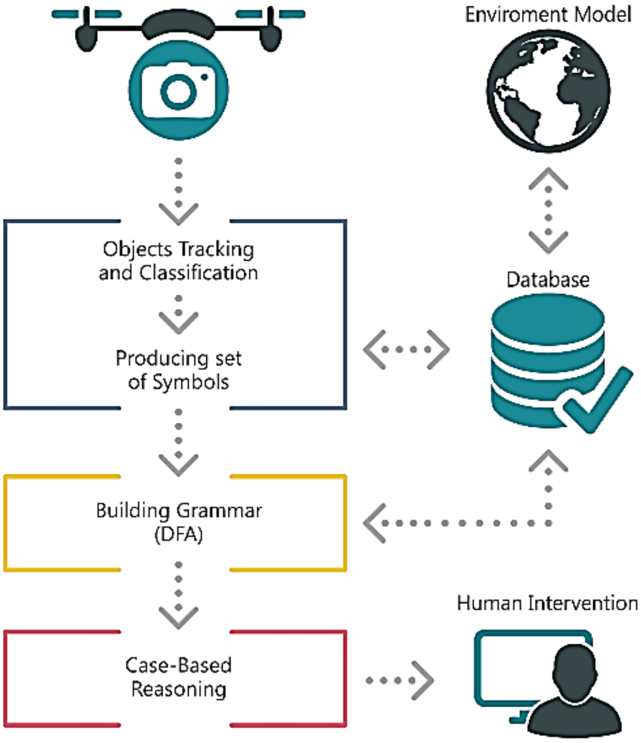


Figure 1: Simplified diagram of the proposed surveillance algorithm.

2.1 Low-Level

Before the image recognition algorithm, it is necessary to reduce the amount of data presented in the images. A few methods are applied to optimize the video capture. For more information see [18]. The low-level uses the Bag of Words (BOW) method to acquire perceptual knowledge using a camera. This method requires a moderated processing power. This enables the application onboard and online in a companion computer. Note that the processing power required for many algorithms is a restraining factor. For instance, the Convolutional Neural Networks (CNN) algorithm can be quite demanding in this aspect and being unfeasible despite its formidable recognition ability to recognize objects [19]. The BOW method [20] recognizes objects present in the image using a sequence of steps. First, the algorithm generates a certain number of descriptors. After, it organizes them and applies a classification technique to identify the objects. In this work, the set of descriptors is extracted by Scale-invariant feature transform (SIFT) [21] algorithm. The descriptors are submitted to a trained classifier that determines which classes are present in the images, building, then, a vocabulary of visual words. The Support Vector Machine (SVM) is the classifier responsible for learning the differences in each image class [22].

Once the SVM determines which object is present in a given image, it is necessary to find its position. Thus, to locate the objects, a sliding window is applied with a larger step, e.g., 30 pixels at a time in the first scan. This may result in a higher uncertainty in position. However, in case an actor is detected, that is, a human or car are present in the scene, the sliding window must have a smaller step in the area where the object was detected in the previous image, e.g., moving 10 pixels at a time. This engine performs more detailed object searches in areas where people or cars are present. It performs a sparser search in free zones, maintaining a minimum number of frames per minute without overloading the algorithm.

2.2 Middle-Level

As mentioned previously, the middle-level integrates the low-level information with the reasoning system at the high-level. Among the tasks allocated in this level, it can be highlighted the sensor data organization into useful information, performing specialized data-related tasks and data storage. In a fully autonomous architecture, this level will contain several parts. Those parts are responsible for specialized activities such as object recognition, path planning, mapping, among others. Despite the importance of each of those activities, this work will focus on the cognitive aspect of this level. This means to organize the incoming stream of sensor data into correlated standardized useful information. This information will be later applied to the decision-making at the high-level.

In previous work, the Deterministic Finite Automata (DFA) was applied to fulfill this task due to its predictability, performance, and flexibility [17]. However, the DFA has a relevant limitation, that is, the inability to process the low-level numerical information along with the acceptance states, such as the likelihood associated in each recognized instance, which can be an essential factor to the higher-level decision-maker. To exemplify this situation, it is supposed a case where an object, such as a knife, is recognized with very low certainty. In such a case, the UAV decision may not be to inform the user, but rather to follow the supposed knife before taking further action.

Different types of automata were defined to allow the development of the quantitative language, such as the weighted and probabilistic automata presented in [23] and [24], respectively. These languages use weights associated with accepted symbols as costs related to the transition execution or probabilities distribution of its occurrences. These approaches limit the application in certain real-world problems. Thus, we propose:

- Multiple information being propagated through the same automata;
- Automata have associated the likelihood of each instance recognized in the low-level layer;
- The probabilities can be mapped by the activation functions.

A few basic concepts are defined to aid the development of the proposed approach. Note that the concepts are based on the weighted automata, which associates a weight to every accepted word. However, in this new approach, the transitions are activated by the acceptance of the input symbol. Along with the symbol acceptance, a second condition is defined related to the transition probability of occurrence. The transition will only happen if the symbol is accepted, and its associated likelihood is higher than a threshold established for that transition.

Definition 1. In the proposed approach, Λ is a vector of n-values in such a way that $\Lambda \in \Re | \Lambda \geq 0$ and it is populated with initial conditions. This means that Λ is a single dimension vector in the $\Re$ space and it represents the automata initial conditions. M is a binary matrix that determines if each transition affects the values in Λ. This definition means that the execution of the automata propagates different information that can be affected by determinate transitions or not.

Definition 2. Given a manifold of functions δ defined over a set of states Q and a set of input symbols Σ with a real value associated $|\rho|$. Each input set $\{w, |\rho|\}$ is mapped into a value between $[0, \infty]$.

$$\delta : \{w, |\rho|\} \to [0, \infty], w \in \Sigma \tag{1}$$

Definition 3. The proposed automaton is a tuple:

$$A = (\Sigma, Q, \Delta, q_0, F, \Lambda, M) \tag{2}$$

With the following accepted language:

$$L(A) = w : \Delta(q, w) \in F \& \delta(\rho) > 0 \tag{3}$$

Where:

- Σ is a set of input symbols with associated values;
- Q is set of states;
- w is the input symbol;
- Δ is a table formed by input symbols accepted in determinate transition with respective function;

- q_0 is the set of initial states;
- F is the set of accepted states;
- Λ is a vector of initial values;
- M is the influence matrix.

Definition 4. For each symbol accepted in the automata transition $("i", "j")$, the k_{th} value in Λ is updated accordingly to $M_{i,j}$ by a nonnegative binary operation with the function output δ.

$$\begin{aligned}\Lambda_{new}(k) &= \Lambda_{previous}(k)\delta(\rho),\ if\ M_{(i,j)} = 1\\ \Lambda_{new}(k) &= \Lambda_{previous}(k),\ if\ M_{(i,j)} = 0\end{aligned} \tag{4}$$

Example 1. Consider a system that has an event sequence $\Sigma = (a, b, c)$ associated to a single likelihood. The execution must happen in the following order: action "a" with a certainty of 0.8 or $(a\ \ 0.8)$, action b with certainty of 0.9 or $(b\ \ 0.9)$ and action c with a real value 5 $(c\ \ 5)$, where $c \in 1, 2, 3, 4, 5, 6$, i.e. there is no uncertainty associated with action c. As valid values are associated with the first two actions, a linear equation $\delta(\rho) = \rho$ can be used and once there is no uncertainty associated in the last action, the output will be always 1 for each input value. This language is accepted by the automata shown on Figure 2.

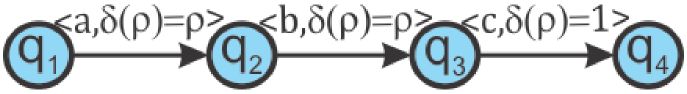


Figure 2: Automata for the Example 1.

An important property of this proposal is the execution equivalence with the DFA. This is important because if the automata execution is equivalent to the DFA, the mathematical evidence necessary to prove the automaton existence is simplified. To prove it, consider the DFA as tuple $A_D = (\Sigma_D, Q_D, \delta_D, q_{0D}, F_D)$ and the language accepted by this definition for a given symbol input set w^* as $L(A_D) = \{w^* : \delta_D(q_D, w^*) \in F_D\}$, the equivalence of the proposition A compared with the automata A_D can be determined if the languages accepted by both are equivalent, i.e. $L(A_D) = L(A)$ [18].

This is important because due to the simplified proof of the proposition. Then, two states are indistinguishable if for $\delta_{D'}(q_{1D}, w^*)$ and $\delta_{D''}(q_{2D}, w)$, the following relation is true $\{(q_{1D}, q_{2D}) : q_{1D} \in F\ \&\ q_{2D} \in F\}$. This means that for a set of inputs, the sequence of transitions results in the same final accepted state regardless the taken path. In other words, looking at the states with a black box interconnecting them and asking the question, "does the same inputs produces the same outputs?".

Theorem 1. If the execution of a set of states is indistinguishable, it can be said as equivalent.

Proof 1. The proof comes by contradiction since if there is a way to differentiate the execution between the first and second set of states, it must exist a string $w \in \Sigma$ that does not satisfy the proposed relation. The functions used in the proposed Weighted Finite Automata (WFA) are not directly supported by the DFA. Thus, to proper demonstrate the existence of an equivalent DFA, it is necessary to find a set of symbols to match both languages. Given the weighted automata defined above, the acceptance state is determinate by the associated input symbol w and the manifold function output $\delta(\rho) > 0$, as represented in Figure 3 (a). As both information always arrives at the same time, it is possible to determine which one is verified first and to build an equivalent automaton using Theorem 1, as presented in Figure 3 (b), which introduces a hidden state q_{01h}.

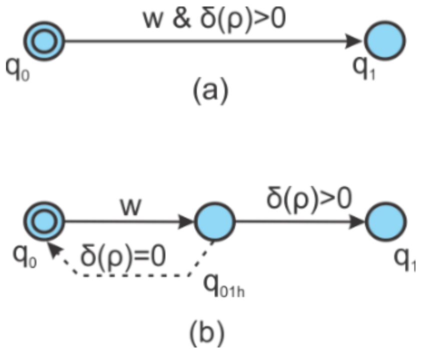
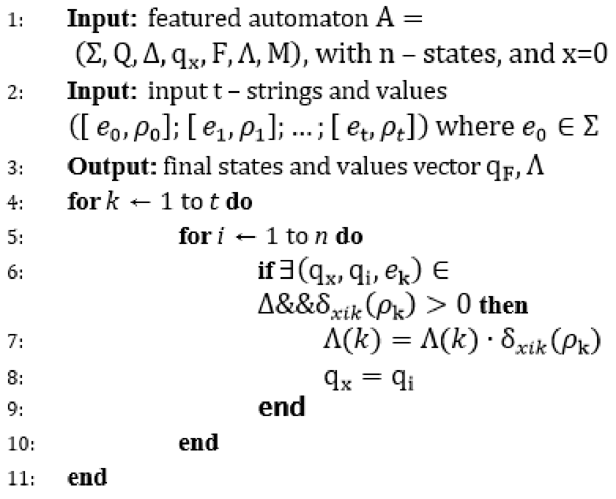


Figure 3: State Acceptance. (a) New Approach Transition. (b) Alternative Representation with Separated Conditions Tests.

At this stage, it is possible to consider the manifold function requirement $\delta(\rho) > 0$ as a symbol $w' \in \Sigma^{'}$ which can be defined belonging to the previously defined set of symbols or $\Sigma \in \Sigma^{'}$. As result, if $\Sigma_D = \Sigma^{'}$ and $Q = Q_D$, it must be possible to build a DFA that is equivalent to the proposed model. Finally, the DFA properties must be valid once every transition in the proposed model can be individually mapped into a DFA. After the basic definitions, a simplified algorithm is built to show the automata execution, presented in Fig 4. Note that the proposed structure barely increases the computational complexity of the automata

© Brazilian Computational Intelligence Society



```
Input: featured automaton $A = (\Sigma, Q, \Delta, q_x, F, \Lambda, M)$, with n – states, and x=0
Input: input t – strings and values ($[e_0, \rho_0]; [e_1, \rho_1]; \ldots; [e_t, \rho_t]$) where $e_0 \in \Sigma$
Output: final states and values vector $q_F$, $\Lambda$
for $k \leftarrow 1$ to $t$ do
    for $i \leftarrow 1$ to $n$ do
        if $\exists (q_x, q_i, e_k) \in \Delta \&\& \delta_{xik}(\rho_k) > 0$ then
            $\Lambda(k) = \Lambda(k) \cdot \delta_{xik}(\rho_k)$
            $q_x = q_i$
        end
    end
end
```

Figure 4: Automata Execution.

Table 1: Primary Activities for some Patterns.

Pattern	Primary Activity
Activity 1	Pose 1, Pose 2 and Pose 3
Activity 2	Pose 1, Pose 2 and Pose 4

algorithm. The extra computational cost corresponds only to the activation function calculation and an extra condition in the algorithm required to verify the output function.

Example 2. Human activity recognition is a prominent application for the proposed structure. In this field, a very common methodology consists in detecting human poses to determine which activity is being recognized, such as in [25]. The first example will be based on [25]. This example assumes that a human activity is determined by observing a set of three individual poses. For simplification, it is also assumed that a strict sequence is not required. The poses can be recognized by several different methods, such as computational vision, accelerometers, among others. However, a limitation in the recognition process arises due to the lack of accuracy to determine the pose. This is explained by the interferences of the sensors that may affect the measurements. This results in a likelihood measurement for each individual detected pose. The activities are presented in the Table 1.

One simple way to represent the proposed scheme is using a simple DFA, as shown in Figure 5. In this method, the recognition certainty of each pose is not processed by the automata. The binary output indicates which pose was accepted at the end of the automata execution. Also, it is not possible to estimate which activity has more likelihood of recognition before the sequence is completely accepted.

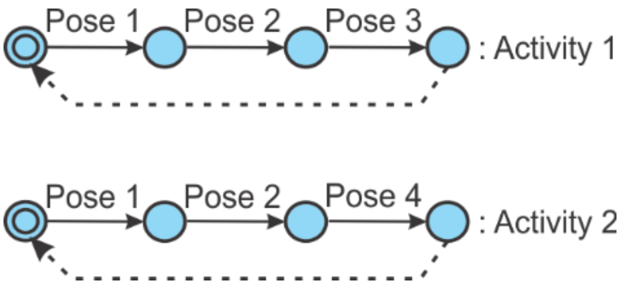


Figure 5: Activity Recognition Automata.

A similar WFA is defined to represent the same problem to show the powerful representation of the proposed method. First, two activation functions must be defined to determine a model for the WFA. The first function is $\delta(\rho) = 0.1$ is presented in Figure 6 (a) and the second one is selected as $\delta(\rho) = \rho$ (Figure 6 (b)). The first function will be used to penalize the recognition probability when a pose is not part of the activity. The second one enables direct transfer of the likelihood input and allows its computation.

Now, it is possible to convert the DFA structure of Figure 5 to the proposed approach. The following notation is used $< input, \ activation \ function \ associated >$ to represent the activations functions associated with each transition. Another difference is the initial likelihood representation noted in brackets. Figure 7 exhibits the proposed topology for this problem for Activity 1.

One way to simplify the automata representation is to use a matrix that correlates the transitions for each state. For the Automata represented in Figure 7, the matrices of $\Delta_{activity1}$ uses this representation and is shown in Table 2. Similarly, for

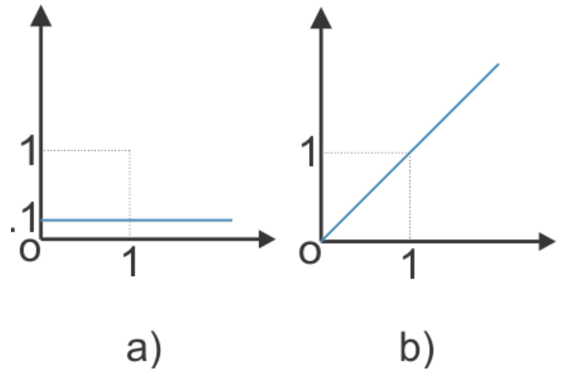


Figure 6: Acceptance Function. (a) Function $\delta(\rho) = 0.1$. (b) $\delta(\rho) = \rho$.

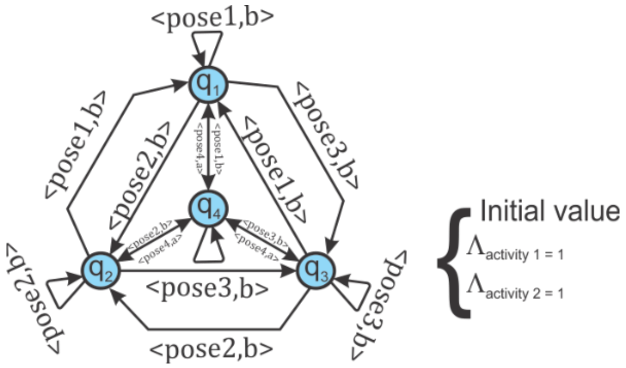


Figure 7: Proposed Automata for Activity 1 of Example 1.

Activity 2 is possible to build $\Delta_{activity2}$ that is shown on Table 3.

Few outputs will be detailed for specific given inputs to illustrate the automata operation. The automaton is then run for three pose sequences to show its operation. The first sequence is: $(pose1 \;\; 1)$, $(pose2 \;\; 0.8)$ and $(pose3 \;\; 0.72)$. Note that the execution generates a higher likelihood for one of the proper activities (i.e. in this case activity 1), as shown in Table 4.

In case that the last pose changes, i.e., $(pose1 \;\; 1)$, $(pose2 \;\; 0.8)$ and $(pose4 \;\; 0.72)$ the activity 2 is recognized as having a higher likelihood, as shown in Table 5.

It is also important to note that this approach makes the automata output more flexible. In the examples, an improper sequence that contains the inputs pose4 or pose3 will also tends towards selecting one of the activities as outputs. For the surveillance problem in this research work, the authors propose an implementation that is capable to receive the elements presented in each scene, namely actors (C), actions(A), objects(O) (as can be seen in the green region of Figure 8), scene features such as time (T) and location(P) (blue part of Figure 7). Those inputs will carry likelihoods calculated by its specific recognition algorithms. Those input elements are processed by the proposed automata. In the representation, the exclamation and interrogation symbols are used as synchronized transitions. Then, the input signals are accepted producing specific synchronization signals. At the end of each input sequence, a probability is computed and feed into the high-level. To end the execution, the automaton resets through the dashed lines to receive the next data sequence.

This process is executed at each sampling of the vision recognition algorithm, which will be considered a scene. For each detected action, the automata will calculate the occurrence likelihood based on all the components. This likelihood can be used in the high-level for decision-making, increasing the algorithm reliability.

2.3 High-Level

The automata produce a valid case containing a non-repetitive structure with the important objects and their relations in each scene. This allows the algorithm in the high-level (i.e., CBR) to analyze the information and to produce a partial view of the UAV

Table 2: Matrix for $\Delta_{activity1}$.

	q1	q2	q3	q4
q1	(pose1 b)	(pose2 b)	(pose3 a)	(pose4 a)
q2	(pose1 b)	(pose2 b)	(pose3 0.1)	(pose4 a)
q3	(pose1 b)	(pose2 b)	(pose3 0.1)	(pose4 a)
q4	(pose1 b)	(pose2 b)	(pose3 0.1)	(pose4 b)

Table 3: Matrix for $\Delta_{activity2}$.

	q1	q2	q3	q4
q1	(pose3 b)	(pose2 b)	(pose4 a)	(pose4 b)
q2	(pose2 b)	(pose2 b)	(pose4 a)	(pose4 b)
q3	(pose1 b)	(pose2 b)	(pose4 a)	(pose4 b)
q4	(pose1 b)	(pose2 b)	(pose4 a)	(pose4 b)

Table 4: Automata Operation for First Input Sequence.

Input Likelihood				
Pose 1	**Pose 2**	**Pose 3**	**Final Likelihood**	
1	0.8	0.9	0.72	**Activity 1**
1	0.8	0.1	0.08	**Activity 2**

Table 5: Automata Operation for Second Input Sequence.

Input Likelihood				
Pose 1	**Pose 2**	**Pose 3**	**Final Likelihood**	
1	0.8	0.1	0.08	**Activity 1**
1	0.8	0.9	0.72	**Activity 2**

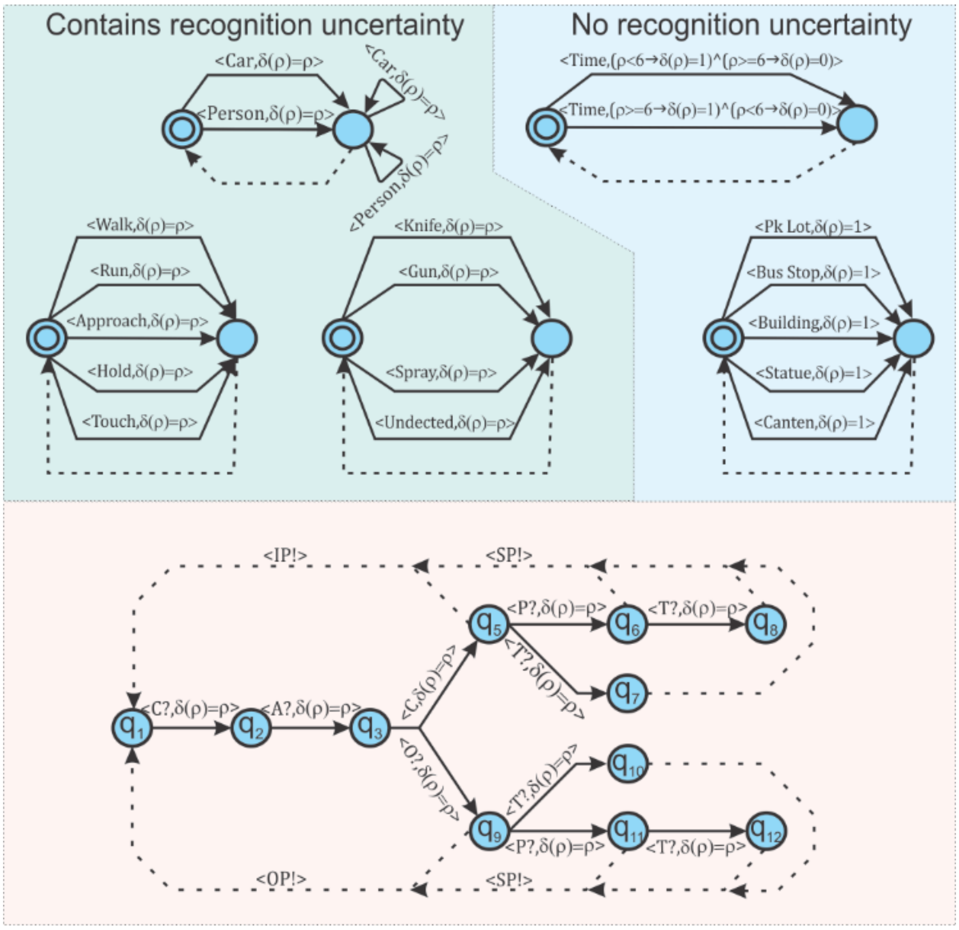


Figure 8: Midle level automata.(a) Function $\delta(\rho) = 0.1$. (b) Function $\delta(\rho) = \rho$.

world as well as to enable the respective system to comprehend scenes and minimizing the requirement for human supervision. The CBR stores the acquired knowledge (i.e., cases) and searches for similar previous cases in the database to solve new inputs and, then, an action can be inferred from the adaptation of the most suitable solutions found in the database [26]. Note that the cases are formed in each processed frame by the low-level. The last decisions are taken into consideration in the next scene analysis. The main components of a case are (1) People, objects and relations; (2) Time and UAV position; (3) The last set of decisions; (4) Decision that UAV should take. The authors well describe all the processes in [27] and [17].

3 RESULTS AND DISCUSSION

Thirteen scenes were recorded using a UAV at a fixed height of 8 meters to evaluate the proposed architecture. Table 6 presents the collected scenes. The scenes contain specific events to be detected by the high-level mechanism. The data consists of the video recorded by the UAV along with Flight Control Unit (FCU) sensors data. All codes and interfaces were developed in a Linux compatible environment aiming to easy integration with the Robot Operating System (ROS). Figure 9 presents the UAV used in the experiments. This UAV was develop by the authors and it contains a LogiTech Webcam c920 camera, a single-board computer Intel Atom Z3775 1.46GHz responsible for all processing related to the hierarchical architecture, a Pixhawk 2.4.8 32bit flight controller with an ArduPilot software and a NEO-8M GPS.

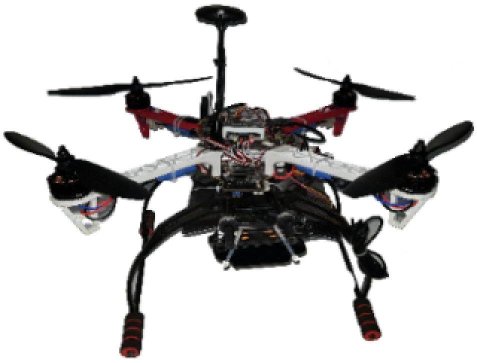


Figure 9: Developed
UAV used in the experiments.

Table 6: Collected Scenes.

	Contents			
Scenes	**Car**	**People**	**Objects**	**Event**
1, 2, 3	Absent	Present	Gun	Theft
4, 5	Present	Absent	Gun	Theft
6, 7	Absent	Present	Spray	Graffiti
8, 9	Present	Absent	Absent	Irregular parking
10, 11	Present	Present	Absent	People/Car Interactions
12, 13	Absent	Present	Absent	People Interactions

Figure 10 presents the developed onboard HMI interface responsible for transmitting to the operator the CBR analysis and the UAV flight details through the X11VNC server. In the recorded scene, the object ”Gun” is recognized, and the proposed architecture decides to and to follow and to inform the operator. The interface intention is to allow the operator to learn new cases, to visualize the UAV primary data, and to receive feedback about the system detection.

This section also evaluates the prediction results of the data. The results showed here are the outputs of the CBR reasoning based on the information processed by the automata in the middle-level. Therefore, a confusion matrix is used for this evaluation [28]. Table 7 shows the confusion matrix for the model. Table 8 and Table 9 present the parameters obtained through the confusion matrix of Table 7. The videos from Table 6 were manually annotated, frame by frame, to produce these results. Using the annotations, it was possible to compare the results with the automata output.

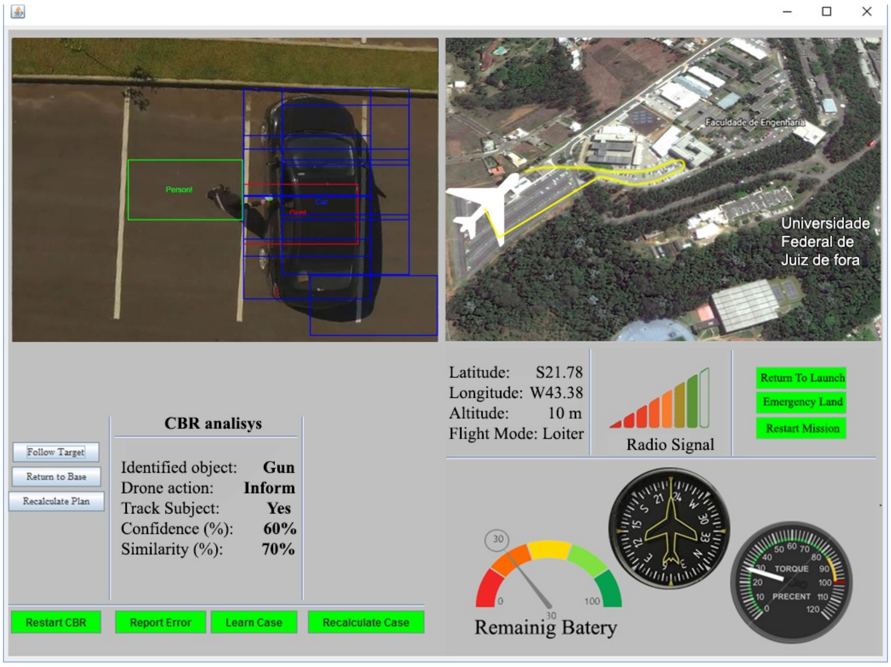
Table 7: Class car (Instances: 216) / Class People. (Instances: 247). / Class Gun. (Instances: 270).

Predicted class		Positive	Negative
Class Car	Positive	100	5
	Negative	0	111
Class People	Positive	67	2
	Negative	34	144
Class Gun	Positive	44	13
	Negative	46	167

The CBR algorithm was applied to the data collected in Table 9. For each video, a segment comprised of 180 frames was selected. The high-level analyzed this segment against the expected result, considering the presence of the event (positive) or not (negative). Table 9 illustrates the obtained results.

The calculation used to produce Table 8 is shown in Table 10. The basic terminology used in the previous confusion matrices is given by:

- **Condition Positive (P):** The number of real positive cases in the data;



CBR analisys

Identified object: Gun
Drone action: Inform
Track Subject: Yes
Confidence (%): 60%
Similarity (%): 70%

Latitude: S21.78
Longitude: W43.38
Altitude: 10 m
Flight Mode: Loiter

Figure 10: Action Recognizing through the system interface.

Table 8: Confusion Matrix Characteristics

	Classes				Classes		
Value	**Car**	**People**	**Gun**		**Car**	**People**	**Gun**
ACC	0.98	0.85	0.78	**TPR**	1	0.66	0.48
FDR	0.05	0.03	0.23	**FNR**	0	0.33	0.51
NPV	1.00	0.81	0.78	**FPR**	0.04	0.01	0.07
DOR	0.00	0.19	0.22	**TNR**	0.96	0.99	0.93
LR+	23	48	6.8	**LR-**	0	0.3	0.6

Table 9: Detected events for each occurrence.

Class		**Decision Accuracy**
Theft	**Positive**	70%
	Negative	95%
Vandalism	**Positive**	65%
	Negative	93%
Violation	**Positive**	92%
	Negative	95%

- **Condition Negative (N):** The number of real negative cases in the data;
- **True Positive (TP):** Condition positive detected as positive;
- **True Negative (TN):** Condition negative detected as negative;
- **False Positive (FP):** Equivalent with false alarm;
- **False negative (FN):** Equivalent with miss;

The scenarios analyzed in this work showed that the proposed methodology could operate in these test scenarios. This is a relevant result, once expands into the current state-of-the-art of the automata development. Despite the positive results, the work did not produce a fully operational algorithm to be deployed in any real situation, which means that for deployment in a product, the automata algorithm from Figure 8 still need to be improved. However, the proposed model is still valid.

4 CONCLUSIONS AND FUTURE WORK

This research work presented an innovative automata approach that enables multiple information propagated through the same automata. The proposed automata allowed proper processing of the likelihoods or other numeric data that is propagated in its structure. In the proposed application, the object recognition certainty of each instance in the low-level layer is processed with the automata. As a result, the high-level can perform reasoning based on the certainty of the object presence.

Table 10: Confusion Matrix Parameters.

ACC	$\frac{\Sigma True\ Positive+\Sigma True\ Negative}{\Sigma Total\ Population}$	$\frac{TP+TN}{TP+TN+FP+FN}$
TPR	$\frac{\Sigma True\ Positive}{\Sigma Condition\ Positive}$	$\frac{TP}{TP+FN}$
FDR	$\frac{\Sigma False\ Positive}{\Sigma Predicted\ Condition\ Positive}$	$\frac{FP}{FP+TP}$
FNR	$\frac{\Sigma True\ Negative}{\Sigma Condition\ Negative}$	$\frac{TN}{TN+FP}$
NPV	$\frac{\Sigma True\ Negative}{\Sigma Predicted\ Condition\ Negative}$	$\frac{TN}{TN+FN}$
FPR	$\frac{\Sigma False\ Negative}{\Sigma Condition\ Positive}$	$\frac{FP}{TN+FP}$
TNR	$\frac{\Sigma True\ Negative}{\Sigma Condition\ Negative}$	$\frac{TN}{TN+FP}$
LR+	$\frac{True\ positive\ rate}{False\ positive\ rate}$	$\frac{TPR}{FPR}$
LR−	$\frac{False\ negative\ rate}{True\ Negative\ Rate}$	$\frac{FNR}{TNR}$
DOR	$\frac{Positive\ likelihood\ ratio\ (LR+)}{Negative\ likelihood\ ratio\ (LR-)}$	$\frac{LR+}{LR-}$

Also, this work proposed an architecture to perform reasoning. Note that the architecture is capable of encoding complex activities and relationships in a simple and manageable structure. The experiments showed the effectiveness of the architecture and the semantic automata structure given a sequence of information computed by the onboard camera. The results of this work can be applied in security footage for surveillance applications.

A few extensions are foreseen for this work. First, a definition of more specialized structures in each architecture level allowing more than a recognition. It is also expected to improve the implementation of the proposed automata to provide more information to the high-level, making the architecture operation flexible to other scenarios. Moreover, implementation of standardized interfaces allowing code distribution.

REFERENCES

[1] H. Qian, Y. Mao, W. Xiang and Z. Wang. "Recognition of human activities using SVM multi-class classifier". *Pattern Recognition Letters*, vol. 31, no. 2, pp. 100–111, 2010.

[2] Z. Li, Y. Liu, R. Hayward, J. Zhang and J. Cai. "Knowledge-based power line detection for UAV surveillance and inspection systems". In *2008 23rd International Conference Image and Vision Computing New Zealand*, pp. 1–6. IEEE, 2008.

[3] A. Puri. "A survey of unmanned aerial vehicles (UAV) for traffic surveillance". *Department of computer science and engineering, University of South Florida*, pp. 1–29, 2005.

[4] Y. Yang, Y. Li, C. Fermuller and Y. Aloimonos. "Robot learning manipulation action plans by" Watching" unconstrained videos from the world wide web". In *Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015.

[5] L. Snidaro, M. Belluz and G. L. Foresti. "Representing and recognizing complex events in surveillance applications". In *2007 IEEE Conference on Advanced Video and Signal Based Surveillance*, pp. 493–498. IEEE, 2007.

[6] L. Snidaro, M. Belluz and G. L. Foresti. "Modelling and managing domain context for automatic surveillance systems". In *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 238–243. IEEE, 2009.

[7] Z. Li, Y. Liu, R. Walker, R. Hayward and J. Zhang. "Towards automatic power line detection for a UAV surveillance system using pulse coupled neural filter and an improved Hough transform". *Machine Vision and Applications*, vol. 21, no. 5, pp. 677–686, 2010.

[8] A. Qadir, W. Semke and J. Neubert. "Vision based neuro-fuzzy controller for a two axes gimbal system with small UAV". *Journal of Intelligent & Robotic Systems*, vol. 74, no. 3-4, pp. 1029–1047, 2014.

[9] P. Dollár, V. Rabaud, G. Cottrell and S. Belongie. "Behavior recognition via sparse spatio-temporal features". VS-PETS Beijing, China, 2005.

[10] B. D. Argall, S. Chernova, M. Veloso and B. Browning. "A survey of robot learning from demonstration". *Robotics and autonomous systems*, vol. 57, no. 5, pp. 469–483, 2009.

[11] K. Pastra and Y. Aloimonos. "The minimalist grammar of action". *Philosophical Transactions of the Royal Society B: Biological Sciences*, vol. 367, no. 1585, pp. 103–117, 2012.

[12] A. Guha, Y. Yang, C. Fermu, Y. Aloimonos *et al.*. "Minimalist plans for interpreting manipulation actions". In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 5908–5914. IEEE, 2013.

[13] J. Wiedermann. "Towards computational models of artificial cognitive systems that can, in principle, pass the turing test". In *International Conference on Current Trends in Theory and Practice of Computer Science*, pp. 44–63. Springer, 2012.

[14] J. Pestana, J. L. Sanchez-Lopez, S. Saripalli and P. Campoy. "Computer vision based general object following for gps-denied multirotor unmanned vehicles". In *2014 American Control Conference*, pp. 1886–1891. IEEE, 2014.

[15] V. Cichella, I. Kaminer, V. Dobrokhodov and N. Hovakimyan. "Coordinated vision-based tracking for multiple uavs". In *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 656–661. IEEE, 2015.

[16] E. Semsch, M. Jakob, D. Pavlicek and M. Pechoucek. "Autonomous UAV surveillance in complex urban environments". In *Proceedings of the 2009 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology-Volume 02*, pp. 82–85. IEEE Computer Society, 2009.

[17] M. F. Pinto, A. G. Melo, A. L. Marcato and C. Urdiales. "Case-based reasoning approach applied to surveillance system using an autonomous unmanned aerial vehicle". In *2017 IEEE 26th International Symposium on Industrial Electronics (ISIE)*, pp. 1324–1329. IEEE, 2017.

[18] M. Almeida, N. Moreira and R. Reis. "Testing the equivalence of regular languages". *arXiv preprint arXiv:0907.5058*, 2009.

[19] R. S. Andersen, C. Schou, J. S. Damgaard and O. Madsen. "Using a flexible skill-based approach to recognize objects in industrial scenarios". In *Proceedings of ISR 2016: 47st International Symposium on Robotics*, pp. 1–8. VDE, 2016.

[20] Y. Benezeth, P.-M. Jodoin, B. Emile, H. Laurent and C. Rosenberger. "Comparative study of background subtraction algorithms". *Journal of Electronic Imaging*, vol. 19, no. 3, pp. 033003, 2010.

[21] D. G. Lowe *et al.*. "Object recognition from local scale-invariant features." In *iccv*, volume 99, pp. 1150–1157, 1999.

[22] H. Bay, T. Tuytelaars and L. Van Gool. "Surf: Speeded up robust features". In *European conference on computer vision*, pp. 404–417. Springer, 2006.

[23] M. P. Schützenberger. "On the definition of a family of automata". *Information and control*, vol. 4, no. 2-3, pp. 245–270, 1961.

[24] K. Chatterjee, L. Doyen and T. A. Henzinger. "Probabilistic weighted automata". In *International Conference on Concurrency Theory*, pp. 244–258. Springer, 2009.

[25] E. Kim, S. Helal and D. Cook. "Human activity recognition and pattern discovery". *IEEE pervasive computing*, vol. 9, no. 1, pp. 48–53, 2009.

[26] S. Vacek, T. Gindele, J. M. Zollner and R. Dillmann. "Using case-based reasoning for autonomous vehicle guidance". In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4271–4276. IEEE, 2007.

[27] M. F. Pinto, F. O. Coelho, J. P. De Souza, A. G. Melo, A. L. Marcato and C. Urdiales. "EKF Design for Online Trajectory Prediction of a Moving Object Detected Onboard of a UAV". In *2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO)*, pp. 407–412. IEEE, 2018.

[28] H. Lewis and M. Brown. "A generalized confusion matrix for assessing area estimates from remotely sensed data". *International Journal of Remote Sensing*, vol. 22, no. 16, pp. 3223–3235, 2001.