Learning and Classification of Events in Monitored Environments

J. Albusac[1] J.J Castro-Schez[2] L.M Lopez-Lopez[2] D. Vallejo[2] L. Jimenez-Linares[2]

1.E.U.P.A, University of Castilla-La Mancha
Almaden, Spain
2.Escuela Superior de Informatica, University of Castilla-La Mancha
Ciudad Real, Spain
Email: {javieralonso.albusac, josejesus.castro, luis.jimenez, lorenzomanuel.lopez, david.vallejo}@uclm.es

Abstract—
This paper presents a prototype system to automatically carry out surveillance tasks in monitored environments. This system consists in a supervised machine learning algorithm that generates a set of highly interpretable rules in order to classify events as normal or anomalous from 2D images without needing to build a 3D model of the environment. Each security camera has an associated knowledge base which is updated when the environmental conditions change. To deal with uncertainty and vagueness inherent in video surveillance, we make use of Fuzzy Logic. The process of building the knowledge base and how to apply the generated sets of fuzzy rules is described in depth for a virtual environment.

Keywords— Automated Video Surveillance, Visual Information Analysis, Machine Learning, and Fuzzy Logic.

1 Introduction

The problem of intelligent surveillance deals with the perception, interpretation, and identification of activities and situations that occur in monitored environments [1]. A typical surveillance scenario consists of a set of CCTV cameras deployed on different places in order to perform explicit surveillance on behalf of people and information storage to carry out forensic analysis if needed [2]. When a security guard watches an environment by means of video cameras, he is able to detect if something is going wrong. In other words, he perceives video events and classifies them as normal or anomalous. If he detects an anomalous behaviour or event, he makes the appropriate decisions to solve the problem as soon as possible. However, the system depends on the human component to classify events and to continuously pay attention to the video stream, which is very tiring and, therefore, error-prone [3]. This is our motivation to face the design of a system able to classify behaviours, that is, to propose a method for automating this task in different domains and scenarios.

Events in real environments can be classified as simple or composite. Really, a composite event is a sequence of simple events which are temporally related [4, 5]. However, the analysis of object characteristics and spatial properties may be enough to understand simple events, i.e, without needing a temporal analysis. In this context, there are many elements which can be learned and identified from data sensors. For instance, the object physics characteristics to determine its class or object type [6], trajectories followed by moving objects [7], entry/exit areas [8], allowed proximity relationships between objects and areas, allowed speed for each type of object, etc.

On the other hand, it is necessary to study how to deal with the uncertainty in real-time to carry out surveillance. Normally, an artificial surveillance system cannot totally ensure what is happening from data sensors in most cases. Uncertainty, imprecision, and vagueness are frequently present when these systems try to solve real world problems. For this reason, one of the main problems of this type of systems is the high number of false alarms due to incorrect interpretations.

Several authors have addressed these issues in the literature. Foresti et al. [9] proposed a visual-based surveillance system for real-time event detection and classification, which is based on adaptive high order neural trees. This system carries out object classification, object tracking, and event recognition for understanding normal, suspicious, and dangerous events in parking lots. Buxton and Gong [10] also provided solutions to the problem of event recognition. They proposed techniques based on Bayesian networks for interpreting traffic situations in dynamic scenes. Remagnino et al. [11] described events in a surveillance scenario by using a Bayesian classifier instead of the Hidden Markov Model. The different techniques proposed in this work were employed to model the common event behaviours in car park. On the other hand, fuzzy logic [12, 13] may provide another interesting approach for dealing with the same problem [14]. In [15], a prototype fuzzy system for describing human activity in natural language is described. This description is made by analysing the possible relations among objects in a monitored environment. To do that, the linguistic labels and the fuzzy rules are defined by an expert to classify people behaviour.

The crafting of detailed scene models may provide an effective means of interpreting situations and detecting anomalous behaviours in a static scene, but it is not an appropriate approach in dynamic scenes where the environmental conditions change over time [7]. This is why in this work we use a fuzzy-based machine learning algorithm to generate a set of highly interpretable set of rules to classify real-time events. To do that, the system analyses spatial data from 2D images obtained by cameras in outdoor/indoor environments, and it generates sets of highly interpretable fuzzy rules. The system analysis situations in a similar way as human beings do, that is, it obtains conclusions without needing precise data. For instance, a person does not need to know the exact speed or the absolute position of an object to determine if it is moving fast on the zone in which it is located.

The remainder of this paper is organised as follows. Section 2 describes the problem statement and overviews the processes required to build the knowledge base of a camera and how to apply this knowledge to classify events. Section 3 de-

ISBN: 978-989-95079-6-8

scribes in depth the machine learning algorithm employed to build the different sets of fuzzy rules that compose the knowledge base. Section 4 studies how to apply the system in a well-defined environment. Finally, conclusions are presented in Section 5.

2 Problem Statement

As previously described, one of the main objectives of advanced surveillance systems is to interpret events in an environment from data sensors. Besides, events not only must be identified, but the system must be also able to classify events as normal or abnormal in order to make adequate decisions. This task covers several research areas, we attack two of them: i) how to build the surveillance knowledge base in a concrete scenario and ii) how to apply this knowledge to classify events as normal or abnormal.

On the other hand, a simple event can be defined as a concrete action that occurs at a time instant, and an anomalous simple event may be enough to activate an alarm in a monitored environment. The method proposed in this work determines the normality of a scene according to this type of events. Examples of anomalous simple events are as follows: a vehicle moving too fast in a concrete time instant or an object located in a forbidden zone. We characterise our surveillance domain by means of the following assumptions:

- Video stream is obtained from cameras placed on certain locations of the environment.
- Each camera has an own knowledge base.
- Every camera is fixed, that is, vertical or horizontal movements and zoom imply a new generation of the surveillance knowledge base.
- Video data can be imprecise.

Each video camera has its own knowledge base, which is composed of three set of fuzzy rules, and each set is generated by an independent training set:

1. Set of rules to determine the object's class. This information is critical when classifying events as normal or abnormal because it defines how objects should normally behave.
2. A second set of rules to determine the object's speed taking into account the object's class and its motion vectors obtained in the segmentation process.
3. Finally, a set of rules to infer whether a situation is normal or anomalous according to the object's class, speed, and the areas in which the object could be located.

To generate the first set of fuzzy rules, we employ the size and the position of the objects in the environment from 2D images (frames), and we take into account the camera view point. Thus, the system may learn, for instance, that vehicles are large or very large size objects when they are close to the camera, and small or medium size objects when they are far from the camera. The segmentation algorithm [16] used in this work determines the set of macroblocks in each frame for every object that appears in the scene. A macroblock is the basic unit in a MPEG stream and it is an area of 16 by 16 pixels in which the motion vectors are stored. The displacement between two macroblocks in different frames gives the motion vector and it specifies a distance and a direction. Each object is represented by means of an ellipse that involves the set of macroblocks. As it will be seen further (Section 4), the parameters of the ellipse are used to determine the location and the size of an object.

On the other hand, in order to generate the rules used to determine the object's speed, the position (centre of the ellipse), the displacement of the object between consecutive frames (distance between the centres of the ellipses), and the object class are used as input variables, i.e, each sample of the training set takes a value for every variable. Movements done by people and vehicles may imply different speed. A medium displacement for one person may involve high speed and for a car may involve low or medium speed. The main goal of this stage is to learn to interpret the object speed depending on their sizes and displacements by taking into account the camera point of view. Finally, the last set of fuzzy rules is generated from a training set where each sample represents the situation of an object in a concrete time instant, i.e, its class, speed, and the spatial relations with the zones or areas of the environment.

Once the rules have been generated and the knowledge base has been built, it can be used to classify simple events. The system analyses frames of a MPEG video stream, evaluating where are the objects located in the environment (2D position) and what are their size. From this information, the system obtains the class of each moving object. In this point, the surveillance system possibly knows where are the objects in each moment and what are their classes. Next, the system uses this information and the second set of rules to determine the speed of each object. Finally, the system determines the areas in which each object could be located and the information previously obtained (class and speed) to classify the simple events as normal or anomalous. In other words, the classification process can be defined as the chaining of three sets of fuzzy if-then rules in which the knowledge generated by a set is employed by the next one.

3 Suggested Machine Learning Algorithm

The machine learning algorithm described in this section is based on the work proposed by Castro *et al.* [17]. This method learns a set of maximal structure fuzzy if-then rules from a set of training data instances. This choice has been motivated by the following reasons: (a) The use of fuzzy logic provides a well-defined mathematical framework to deal with the uncertainty and vagueness inherently contained in the visual information used as input data. (b) This algorithm represents the inferred knowledge by means of a set of fuzzy if-then rules expressed in terms of linguistic variables. As a result, we obtain rules that are easily comprehensible by security guards, who are watching the monitors. Thus, an user can easily analyse why an alarm was activated. (c) As a supervised learning approach, once the set of rules has been inferred from the training data set, the algorithm does not require a significant amount of time to classify new data instances.

Basically, Castro's algorithm works in two phases. In the first one, each example of a training set is converted into a

ISBN: 978-989-95079-6-8

specific rule which describes how to act in a concrete situation, that is, how this example must be classified. Therefore, there will be so many particular rules as examples. In a second phase, the algorithm generalises these particular rules in order to act in a wide range of possibilities, that is to say, many examples of a class can be correctly classified by using one generalised rule. To do that, an amplification process extends the particular rules covering situations of the space of possible situations, which have not been covered before.

Formally, the original algorithm starts out the training phase from a set of data instances $\Theta = \{e_1, \ldots, e_m\}$. Each $e_i = ((x_{i1}, \ldots, x_{in}), y_j)$ is a data example which is conveniently described by means of the set of input variables $V = \{v_1, \ldots, v_n\}$ and one output variable y_j, which is called the *class* of e_i. The elements $(x_{i1}, \ldots, x_{in})$ are the concrete values that the example e_i takes for each variable $v_k \in V$. Besides, there is a domain definition DDV_i for each variable in V, being $DDV = \{DDV_1, \ldots, DDV_n\}$ the set of all domains. Each $DDV_i = \{L_1, \ldots, L_p\}$ is composed of a set of linguistic labels which correspond to the fuzzy sets represented by means of trapezoidal functions as specified in expression (1).

$$\prod(u; a, b, c, d) = \begin{cases} 0 & u < a \\ \frac{(u-a)}{(b-a)} & a \leq u < b \\ 1 & b \leq u \leq c \\ \frac{(d-u)}{(d-c)} & c < u \leq d \\ 0 & u > d \end{cases} \quad (1)$$

In this way, each DDV_i verifies the following properties:

1. $\forall L_x \in DDV_i, height(L_x) = 1$
2. $\forall L_x, L_y \in DDV_i, nucleus(L_x) \cap nucleus(L_y) = \emptyset$
3. $\forall x \in X_i, \sum_{j=1}^{|DDV_i|} \mu_{L_j}(x) = 1$, being X_i the domain where v_i is defined.

Besides, the inferred knowledge is represented as a set of fuzzy if-then rules with the following structure:

$$\textbf{if } v_0 \textit{ is } ZD_0 \wedge \ldots \wedge v_n \textit{ is } ZD_n \textbf{ then } y_j \quad (2)$$

where $v_i \in V$ and $ZD_i \subseteq DDV_i$ is a subset of the linguistic labels defined in DDV_i for the variable v_i.

The original algorithm converts each $e_i \in \Theta$ into the fuzzy domain, according to the values $(x_{i1}, \ldots, x_{in})$ that e_i takes for each $v_i \in V$ and their corresponding $DDV_i \in DDV$. For this purpose, each $L_k \in DDV_i$ has associated a function $\mu_{L_k} : X_j \rightarrow [0, 1]$, being X_j the domain where the variable v_j takes its values (i.e $\Re, \aleph$, an interval $[a, b]$, a finite set A, etc). Thus, every $e_i \in \Theta$ is converted into $e'_i = ((L_{1x}, \ldots, L_{nz}), y_j)$ such that L_{jk} is the label which matches best, according to $L_{jk} = max\{\mu_{L_{jk}}(x_j)\}$ of the DDV_j. Each e'_i is an initial rule which belongs to the set of *initial rules*. Second, if some rule of the initial set does not subsume in any rule of the final set, the algorithm proceeds to amplify the rule. A rule R_i can be amplified to $R_{i'}$: **if** v_0 *is* $ZD_{i'0} \wedge \ldots \wedge v_n$ *is* $ZD_{i'n}$ **then** y_p if there is no rule R_j : **if** v_0 *is* $ZD_{j0} \wedge \ldots \wedge v_{jn}$ *is* ZD_n **then** y_q in the set of initial rules that verify $ZD_{jk} \subseteq ZD_{i'k}$ and $y_p \neq y_q$. In other words, an amplification is possible whenever there is no counterexample that conflicts with the amplified rule. This makes possible for this method to generate a set of rules that are as general as possible. A description in detail of the original algorithm by Castro *et al.* can be found in [17].

However, this algorithm may cause troubles when applied in the surveillance context due to the production of rules that are over-generalised. With over-generalised we refer to the tendency of the final rules to cover spaces for which there are no counterexamples. For example, let us suppose three initial rules for classifying a moving object according to its size by taking into account the distance from the camera:

- R_i: **if** $size$ is {SMALL} $\wedge$ $distance$ is {MEDIUM} **then** y_1 is people
- R_j: **if** $size$ is {MEDSMALL} $\wedge$ $distance$ is {MEDIUM} then y_1 is motorbike
- R_k: **if** $size$ is {MEDBIG} $\wedge$ $distance$ is {MEDIUM} then y_1 is car

all of them contained in the set of initial rules. Let DDV_{size} = {VSMALL, SMALL, MEDSMALL, MEDBIG, BIG, VBIG}, if we try to amplify the rule R_i to achieve $R_{i'}$, according to the variable **size**, it could result in:

- $R_{i'}$: **if** $size$ is {VSMALL, SMALL, BIG, VBIG} $\wedge$ $distance$ is {MEDIUM} **then** y_1 is people

Note that $R_{i'}$ means that if an object is located at a *medium* distance from the camera and its size is *very small*, *small*, *big*, or *very big*, it certainly belongs to the class *people*. However, R_j and R_k mean that if an object is located at a *medium* distance from the camera and its size is *medium small* or *medium big*, the object is a *motorbike* or a *car*, respectively. Obviously, $R_{i'}$ lacks of any sense and it can cause problems because of the misclassification of objects. This happens because when amplifying R_i to $R_{i'}$, the labels $VSMALL$, BIG, and $VBIG$ are added to $R_{i'}$ as there are no counterexamples in the set of rules that conflict with it. Moreover, this type of counterexamples might never exist in the training set because it is possible that cars and motorbikes have never a big or very big size, when they are located at a medium distance from the camera.

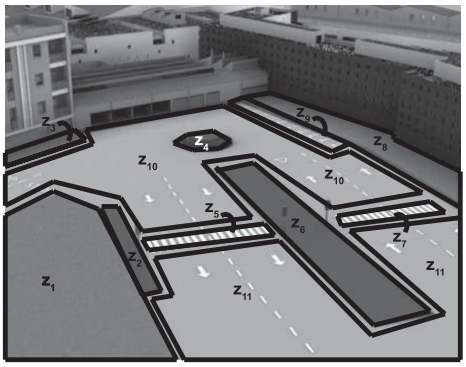
In order to solve this issue, we have modified the original algorithm to restrict the amplification of rules. We are interested in amplifying a rule only if the last label added to the rule in the last amplification step and the label being considered to be added are not too far from each other. Thus, we need to use a measure of separability s to estimate the dissimilarity between linguistic labels L_i and L_j (with $L_i < L_j$), which are expressed in terms of fuzzy sets:

$$s(L_i, L_j) = \frac{(b_{Lj} - c_{Li}) + (a_{Lj} - d_{Li})}{2} \quad (3)$$

being $(a_{Li}, b_{Li}, c_{Li}, d_{Li})$ and $(a_{Lj}, b_{Lj}, c_{Lj}, d_{Lj})$ the trapezoids that define the fuzzy sets L_i and L_j respectively, and verifying that $d_{Li} < a_{Lj}$. If $d_{Li} = a_{Lj}$ or $d_{Li} > a_{Lj}$, then $s(L_i, L_j) = 0$, as the area of the fuzzy set $s(L_i, L_j)$ is not significant enough.

The separability threshold is empirically calculated by observing how the algorithm selects the different values of percentage of the maximum separability between the fuzzy sets

ISBN: 978-989-95079-6-8



Zone Classification	
z_1	Garden
z_2	Sidewalk
z_3	Garden
z_4	Traffic Circle
z_5	Pedestrian Crossing
z_6	Sidewalk
z_7	Pedestrian Crossing
z_8	Garden
z_9	Parking
z_{10}	Road
z_{11}	Road

Figure 1: Definition and classification of areas for the analysed scene.

of a given linguistic variable. If we apply the expression (3) with L_i being the first label and L_j the last label of a linguistic variable, we obtain the maximum possible distance value for that variable. Then, it is possible to choose the 25% or the 10% of such separability as the threshold for amplifying a rule. Note that when the maximum allowed separability is 100%, the new version of the algorithm works as the original one.

Next, the steps of the modified algorithm are described in detail.

1. Convert each training example into an initial fuzzy rule (translation into the fuzzy domain). Each element $e_i \in \Theta$ is translated into a fuzzy rule in which the value of each input variable is represented by means of a linguistic label. This step creates the set of *initial* rules.
2. Take a rule from the set of initial rules.
3. Try to subsume the taken rule in some rule of the set of definitive rules. If that happens so, ignore the taken rule and go back to step 2.
4. If the taken rule does not subsume in any rule of the definitive set, try to amplify it. For each variable:
   (a) For each unconsidered label:
      i. Try to amplify the rule. If it is not possible, go to step 4.a; otherwise, proceed to step 4.a.ii. One rule can be amplified only if:
         A. There is no R_j : **if** v_0 *is* ZD_{j0} $\wedge$ ... $\wedge$ v_n *is* ZD_{jn} **then** y_q in the set of initial rules that $ZD_{jk} \subseteq ZD_{i'k}$ and $y_p \neq y_q$ (we maintain the constraint of the original algorithm).
         B. The separability between the last label added to the rule and the label being considered for amplifying the rule does not exceed a separability threshold.
      ii. Amplify the rule and include it in the set of definitive rules.
5. If there are still unconsidered rules in the initial set of rules, go to step 2. Otherwise, **END**.

4 Applying the Suggested Algorithm to an Example

The proposed machine learning algorithm has been tested on the virtual environment shown in Figure 2 due to the following reasons: (a) the difficulty of obtaining a wide range of real video scenes with anomalous situations, (b) the possibility of generating all abnormal situations as needed without attempting on people security, and (c) the freedom of changing camera position with no configuration cost.

Figure 1 shows the scenario used in this work, which represents a typical urban environment composed of buildings, roads, gardens, traffic signals, and so on. There are also pedestrian areas in which vehicles are not allowed to drive. In the same way, there are only vehicle areas in which pedestrians should not walk.

In our particular application, there are three training sets: Θ_1, Θ_2, and Θ_3, one for each phase of the learning process. Each training set is composed of a set of examples where every one of them e_i is made up from a set of features extracted from the 2D images captured by a video camera, such as the position of a moving object in a frame or the size of the ellipse that contains it. The algorithm described in Section 3 is performed for every training set Θ_i, in order to obtain a set of fuzzy rules. On the other hand, there are three output variables: the class of the moving object (y_1), its speed (y_2), and whether its behaviour is normal or anomalous (y_3).

4.1 Learning Process

As described before, the first step consists in learning the set of rules to determine the object class. Determining the object class is crucial to know if its behaviour in a monitored environment is normal. Each one of these classes has a set of norms or rules. If an object meets the norms associated to its class, then its behaviour will be considered as normal. To do that, we first need to build a training set in which each sample is defined in the following way: $e_i = ((x_{i1}, \ldots, x_{in}), y_j)$, being $(x_{i1}, \ldots, x_{in})$ the variable values used to learn and y_j the output variable. The variables employed in this first phase are $((X_{pos}, Y_{pos}, R_h, R_v)\, y_1)$. X_{pos} represents the moving object horizontal position and Y_{pos} represents the moving object vertical position. The values of X_{pos} and Y_{pos} comes from the coordinate x, y of the ellipse central point that encloses the object. On the other hand, R_h determines the horizontal size measured in pixels of the ellipse that encloses the object and R_v the vertical size measured also in pixels. Finally, y_1 is the output variable, which can take the following values: $\{pedestrian, vehicle\}$.

Each sample refers to a moving object detected in the segmentation process, which has been marked with an ellipse. Precisely, the ellipse parameters (central point coordinates, horizontal radius, and vertical radius) are used to generate the rules in this step. The goal is to determine whether a certain object is a pedestrian or a vehicle depending on its size and position from the camera point of view. Distinguishing be-

ISBN: 978-989-95079-6-8

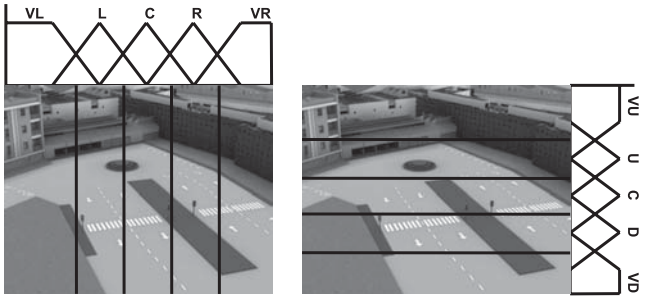


Figure 2: Vertical-horizontal scene division from an image captured by a surveillance camera.

tween people and vehicles depending on size and position is feasible because the difference between them is notable independently of the area in which they are located. On the other hand, to determine the zone in which an object is placed, a vertical-horizontal division as shown in Figure 2 is carried out. In this environment, the far zones from camera position, where objects may move, are located in $X_{pos} = \{\text{centre, right}\}$ and $Y_{pos} = \{\text{up}\}$. However, the close zones from camera position are located in $X_{pos} = \{\text{very left, left, centre}\}$ and $Y_{pos} = \{\text{very down, down}\}$.

An example of a fuzzy rule generated in this stage is as follows:

R : if $Xpos$ is {very right} and $Ypos$ is not {very up, up} and Rh is {very small, small} and Rv is {very small, small} then y_1 is pedestrian

Once applied the first stage, the next step consists in determining the object speed depending on its position and movement (measured in pixels) between the current and previous frame. We consider that two frames per second for the studied environment is enough for detecting anomalous situations. For every object class and zone there is a range of allowed speed values, which should be met by every object. Therefore, the speed study is interesting for the analysis of behaviours in monitored environments. In order to learn the second set of rules, we use input examples with the following form: $((X_{pos}, Y_{pos}, y_1, Mov)\ y_2)$, being Mov the movement of the object measured in pixels between two consecutive frames, and y_2 the output variable which can take the possible following values: $\{slow, normal, fast\}$.

An example of a rule generated in this step is the following:

R : if $Xpos$ is {very left, left} and $Ypos$ is {very down, down} and y_1 is {pedestrian} and Mov is not {medium, low, very low} then y_2 is slow

Finally, the last step in the general learning process consists in generating a set of rules that allows the surveillance system to detect if an object behaves normally or anomalously depending on the object class, its speed and the zones in which it could be located. As in previous stages, a training set is built for the rule generation. Every input sample is represented in the following way: $((y_1, y_2, Garden, Sidewalk, Trafficcircle, Parking, PedestrianCrossing, Road)\ Y_3)$, where every kind of zone represents an input variable whose value sets the intersection grade between the object and the type of zone. Finally, y_3 is the output variable whose values can be $\{normal\ situation, anomalous\ situation\}$.

An example of two rules generated in this final step is:

R : if y_1 is {pedestrian} and y_2 is {slow, normal} and $garden$ is {very high} and $road$ is {out, very low, low} then y_3 is normal situation

R : if y_1 is {pedestrian, vehicle} and y_2 is {fast} and $road$ is {low, medium, high} then y_3 is anomalous situation

4.2 Classification process

Once the learning process is done, the set of rules is available to determine, in each frame, the normality of an object behaviour. To carry out the classification process, first it is necessary to obtain a set of low-level objects data. Next, we expose the description of such data and how to obtain it.

For each frame, the system performs a segmentation process to detect every moving object, which is then bounded within a ellipse. The parameters of the ellipse are employed to determine the position and the size of the bounded object (X_{pos},Y_{pos},R_h,R_v). If an object is moving, the position of its bounding ellipse in the current frame changes respecting to the previous frame. Thus, the movement of an object is calculated through the distance between the origin of the previous and the current ellipses.

To complete the low level data acquisition of the objects, the system performs a proximity analysis of every tracked object to the areas of the environment which are defined according to the camera point of view. The goal of this analysis is to learn in which areas can be located a concrete object class. Thus, an expert is required to define such areas through the definition of their corresponding bounding polygons and the class which every area belongs to. Figure 1 shows the definition and classification of areas performed for the analysed scene.

The next step is to study the intersection between the objects bounding ellipses and the defined areas to learn in which extent an objects is on an area. For this purpose, the system calculates the amount of points in common in an object's bounding ellipse and the polygon defining an area. To do that, we use the following algorithm [18] represented in the programming language C:

```
int pnpoly(int nvert, float *vertx, float *verty,
           float testx, float testy)
{
  int i, j, c = 0;
  for (i = 0, j = nvert-1; i < nvert; j = i++) {
    if ( ((verty[i]>testy) != (verty[j]>testy)) &&
         (testx < (vertx[j]-vertx[i]) * (testy-verty[i])
         / (verty[j]-verty[i]) + vertx[i]) )

         c = !c;
  }
  return c;
}
```

where *nvert* is the number of vertices in the polygon, *vertx* and *verty* the arrays containing the x- and y-coordinates of the polygon vertices and, finally, *testx* and *testy* are the x- and y-coordinate of the test point (the algorithm is performed for each point of the ellipse). The amount of points in common is converted into the fuzzy domain and is classified as *out*, *very low*, *low*, *medium*, *high*, or *very high*.

The classification process proceeds in each frame and it begins once the system knows the low level data of every object in the scene. The system makes use of the three sets of fuzzy rules to determine the object's class and speed, and if its behaviour is normal or, on the contrary, anomalous.

ISBN: 978-989-95079-6-8

In order to evaluate the learning method proposed in this paper, we have run ten tests for every step of the general process. Each training set had a different number of examples, and the 80% of them were used for learning and the rest were used in the classification process. Table 1 resumes the results obtained in the tests, where the average of the generated rules, examples correctly classified, classification errors, and percentage of successful are shown for every step.

Table 1: Results obtained in the tests

Step	Gen. Rules	Correct	Wrong	% Correct
step 1	29	43	2	96.28
step 2	22	79	3	96.34
step 3	50	120	5	96

5 Conclusions

Surveillance systems are being used in a wide range of environments which require more and more sophisticated solutions. In fact, a notable investment has been made during the last years in order to provide surveillance services, both in public and private environments, which increase the efficiency of such systems and manage high-level information to allow users to adequately make decisions and manage crisis situations. In this paper, we have presented a possible solution to classify and describe simple events related to spatial properties in a monitored environment having into account the camera perspective. Moreover, we have also determined whether these events are normal or not.

We have proposed a machine learning algorithm to acquire the necessary knowledge from examples of situations and a method that uses this algorithm. The proposed method includes three calls to the algorithm as has been explained in Section 2.

Our work starts from the fuzzy machine learning algorithm proposed by Castro et al. [17]. A key innovation has been added to this algorithm with the aim of adapting it to the new needs of our problem. This was primarily motivated due to that several generated rules with different consequences could be applied to the same premises, generally scenarios that are not present in the training data. This is due to the generalisation of each particular rule into a definitive rule. The modified algorithm prunes the over-generalisation of a definitive rule during the amplification process. As has been described in Section 3, the over-generalisation is not desirable for us and it will be controlled by means of a separability measurement. This measurement allows one definitive rule to capture the premises that are not present in the training data but close to those evidences in the data that justify that rule. Thanks to this separability measurement the results obtained are improved.

Finally, we want to remark that the separability threshold is empirically calculated by observing how the algorithm selects the different values of percentage of the maximum separability between the fuzzy sets of a given linguistic variable. For this reason, one of our lines of future research consists in designing a new algorithm to decide the best separability value to optimise the results.

Acknowledgment

This work has been founded by the Regional Goverment of Castilla-La Mancha under Research Projects PII2I09-0052-3440 and PII1C09-0137-6488.

References

[1] W. Wang and S. Maybank. A survey on visual surveillance of object motion and behaviors. *Systems, Man and Cybernetics, Part C, IEEE Transactions on*, 34(3):334–352, 2004.

[2] M. Valera and S.A. Velastin. Intelligent distributed surveillance systems: a review. In *Vision, Image and Signal Processing, IEE Proceedings-*, volume 152, pages 192–204, 2005.

[3] G.J.D. Smith. Behind the screens: Examining constructions of deviance and informal practices among cctv control room operators in the uk. *Surveillance and Society*, 2(2/3):376–95, 2004.

[4] J.F. Allen. An interval-based representation of temporal knowledge. In *Proc. 7th International Joint Conference on Artificial Intelligence, Vancouver, Canada*, pages 221–226, 1981.

[5] J. Allen and G. Ferguson. Actions and events in interval temporal logic. In *Journal of Logic and Computation*, volume 4(5), pages 531–579, 1994.

[6] F. Fusier, V. Valentin, F. Bremond, M. Thonnat, M. Borg, D. Thirde, and J. Ferryman. Video understanding for complex activity recognition. *Machine Vision and Applications*, 18(3):167–188, 2007.

[7] H.M. Dee and D.C. Hogg. Navigational strategies and surveillance. In *Proceedings of the IEEE International Workshop on Visual Surveillance*, pages 73–81, 2005.

[8] D. Makris and T. Ellis. Path detection in video surveillance. *Image and Vision Computing*, 20(12):895–903, 2002.

[9] G.L. Foresti, C. Micheloni, and L. Snidaro. Event classification for automatic visual-based surveillance of parking lots. *Proc. of the 17th International Confrence on Pattern Recognition*, 3:314–317, 2004.

[10] H. Buxton and S. Gong. Visual surveillance in a dynamic and uncertain world. *Artificial Intelligence*, 78(1-2):431–459, 1995.

[11] P. Remagnino and GA Jones. Classifying Surveillance Events from Attributes and Behaviour. In *the Proceeding of the British Machine Vision Conference*, pages 10–13.

[12] Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8:338–353, 1965.

[13] T. Takagi and M. Sugeno. Fuzzy identification of systems and its applications to modelling and control. *IEEE Transactions on Systems, Man, and Cybernetics*, 15(1):116–132, Jan-Feb 1985.

[14] P. Angelov and X. Zhou. Evolving Fuzzy-Rule-Based Classifiers From Data Streams. *IEEE Transactions on Fuzzy Systems*, 16(6):1462–1475, 2008.

[15] H. Stern, U. Kartoun, and A. Shmilovici. A prototype Fuzzy System for Surveillance Picture Understanding. *IASTED International Conference Visualization, Imaging, and Image Processing (VIIP 2001), Marbella, Spain*, pages 624–629, 2001.

[16] L. Rodriguez-Benitez, J. Moreno-Garcia, J.J. Castro-Schez, J. Albusac, and L. Jimenez-Linares. Automatic objects behaviour recognition from compressed video domain. *Image and Vision Computing*, 27(6):648 – 657, 2009.

[17] J.L. Castro, J.J. Castro-Schez, and J.M. Zurita. Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. *Fuzzy Sets and Systems*, 101(3):331–342, 1999.

[18] M. Shimrat. Algorithm 112: Position of point relative to polygon. *Communications of the ACM*, 5(8):434, 1962.

ISBN: 978-989-95079-6-8