

# CONSTRUCCION DE CONJUNTOS DE REGLAS DIFUSAS PARA LA CLASIFICACION DE OBJETOS

C.J. Solana-Ciprés<sup>1</sup> J. Albusac<sup>1</sup> J.J. Castro-Schez<sup>1</sup> J. Moreno-García<sup>2</sup> L. Rodríguez-Benítez<sup>1</sup>

<sup>1</sup> Escuela Superior de Informática, Paseo de la Universidad, 4, 13071, Ciudad Real, cayetanoj.solana@uclm.es

<sup>2</sup> E.U.I.T. Industrial, Avda. Carlos III s/n, 45071, Toledo, juan.moreno@uclm.es

## Resumen

En este artículo se presenta un método para la obtención de un conjunto de reglas que puedan ser usadas en la clasificación de objetos en entornos monitorizados. La propuesta está basada en un algoritmo de aprendizaje automático supervisado y toma como información de entrada los resultados obtenidos en un algoritmo de segmentación previamente desarrollado. El método propuesto genera un conjunto de reglas interpretables que pueden ser usadas durante la monitorización de entornos vídeo-vigilados.

**Palabras Clave:** Vídeo-vigilancia, algoritmo de aprendizaje automático, clasificación de objetos.

## 1. INTRODUCCION

Hoy en día, existe una preocupación generalizada por la seguridad de las personas y los edificios frente a actos vandálicos o terroristas. En este contexto, se origina un notable aumento de interés en la investigación en el área de la vídeo-vigilancia inteligente. Generalmente, un sistema de vídeo-vigilancia avanzado se divide en los módulos mostrados en la Figura 1.

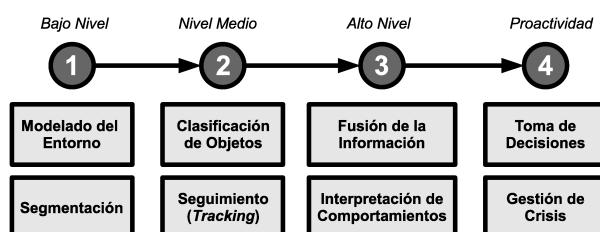


Figura 1: Arquitectura de un sistema de vigilancia.

Dentro de esta división, existe un amplio abanico de

líneas de investigación abiertas. En este artículo se propone un método para poder construir el módulo de clasificación de objetos (*nivel medio*). Este tipo de sistemas suele ejecutar la clasificación después de la segmentación, por lo que la calidad de los resultados de la etapa de segmentación será fundamental para obtener resultados óptimos en el proceso de clasificación.

Habitualmente, los métodos de etiquetado de objetos extraen las características relevantes a partir de la información de los frames capturados. Dichas características suelen basarse en la apariencia, forma, color, movimiento y textura de los objetos detectados. Hu et al. [5] dividen los métodos de clasificación en dos categorías principales: basada en la forma y basada en el movimiento. Sin embargo, Brown [3] distingue entre métodos dependientes o independientes de la perspectiva de la cámara. Por último, Valera y Velastin [9] dividen las propuestas en función del conocimiento geométrico de los objetos a seguir: 2D o 3D.

En trabajos recientes, se ha propuesto una alternativa para resolver la etapa de segmentación a través de un algoritmo de tiempo real sobre el dominio comprimido H.264/AVC [8] y un marco conceptual basado en el análisis de la normalidad para detectar comportamientos anómalos en entornos monitorizados [2]. Dicho esquema se compone de un conjunto de componentes de normalidad que definen el comportamiento de cada objeto desde el punto de vista de la vigilancia. Entre otros, los aspectos a considerar son la trayectoria que sigue un objeto, la velocidad, la normalidad en un paso de peatones, etc. Un objeto se comporta de forma apropiada en un entorno monitorizado si satisface la normalidad definida para cada uno de los componentes de normalidad, los cuales [2] necesitan conocer la clase a la que pertenece cada objeto para poder realizar la interpretación apropiada del comportamiento. Por los motivos expuestos, se puede considerar la clasificación o etiquetado de objetos como una etapa relevante en cualquier sistema de vigilancia avanzado porque permite detectar situaciones anormales.

En el presente trabajo se propone un método para construir un conjunto de reglas útiles para la clasificación de objetos en escenas de vídeo que utiliza un algoritmo de aprendizaje automático supervisado [1]. Dicho algoritmo comienza el proceso de aprendizaje a partir de un conjunto de entrenamiento en el que cada clase de objeto es descrita por un conjunto de ejemplos definidos por medio de un conjunto de características. Es decir, cada ejemplo del conjunto de entrenamiento estará asociado a la clase a la que pertenece. Esta asociación se puede hacer de dos formas, la primera es manual y requiere que sea realizada por un experto en el dominio. La segunda requiere de un proceso automático que realice la asociación. El hacerlo de manera manual tiene como ventaja principal la eliminación de ruido en el conjunto de entrenamiento, lo cual puede tener como efecto la obtención de mejores resultados por el algoritmo de aprendizaje automático; como desventaja hay que destacar que es una tarea lenta y tediosa. En este trabajo se opta por la segunda aproximación para reducir la actividad humana en el proceso, uno de los objetivos del presente trabajo.

A continuación, en la Sección 2 se analiza detalladamente la solución propuesta. En la Sección 3 se muestran los resultados experimentales obtenidos y finalmente en la Sección 4 se resumen las conclusiones y las líneas de investigación abiertas.

## 2. METODO PROPUESTO

En este trabajo se presenta un método que permite aprender un conjunto de reglas que puedan ser usadas en el diseño y desarrollo de un sistema para la clasificación de objetos móviles en entornos monitorizados. El método propuesto está basado en un algoritmo de aprendizaje automático supervisado. Dicho algoritmo tiene como objetivo aprender un conjunto de reglas lingüísticas que modelen la función  $S$  definida como  $S : V \rightarrow O$ . Donde  $O$  es el conjunto de clases de objetos que se mueven por la escena vigilada y  $V$  es información obtenida en la etapa de segmentación: posición vertical ( $v_1$ ) y horizontal ( $v_2$ ) del objeto, tamaño del objeto en número de macrobloques ( $v_3$ ) y en número de vectores de movimiento ( $v_4$ ), relación entre anchura y altura del objeto ( $v_5$ ) y una medida de dispersión del campo de vectores de movimiento pertenecientes al objeto ( $v_6$ ), es decir,  $V = \{v_1, v_2, v_3, v_4, v_5, v_6\}$ . Por su parte, el algoritmo de aprendizaje toma como entrada un conjunto de entrenamiento  $E = \{e_1, e_2, \dots, e_n\}$  donde cada ejemplo es  $e_i = (v_{i1}, v_{i2}, \dots, v_{i6} : o_i)$ , siendo  $v_{ij}$  el valor del ejemplo  $i$  para la variable  $v_j$  y  $o_i$  el tipo del objeto detectado en la escena ( $o_i \in O$ ).

Las características que se miden son representativas de los diferentes tipos de objetos porque definen su

posición ( $v_1$  y  $v_2$ ), tamaño ( $v_3$  y  $v_4$ ) y forma ( $v_5$ ). La posición y el tamaño son características estrechamente relacionadas porque un objeto será más grande o más pequeño en función de su posición respecto a la cámara. Por su parte, la forma supone un factor de distinción clave entre personas y vehículos, como se observa en la Tabla 1, donde se muestra el porcentaje de cada tipo de objeto en función de su ratio anchura/altura. Se ha llevado a cabo un estudio para identificar la relación entre la altura y la anchura de los objetos en una secuencia de vídeo específica; se concluye que las personas y las motocicletas son más altas que anchas, mientras que los coches y camiones son más anchos que altos. Finalmente, el atributo de la dispersión ( $v_6$ ) permite conocer la rigidez de un objeto dado que una persona o un grupo de personas serán codificados con diferentes vectores de movimiento debido a sus articulaciones, mientras que un vehículo será codificado con vectores de movimiento similares en dirección y tamaño.

Cuadro 1: Ratio Anchura (W) / Altura (H).

	$W > H$	$W = H$	$W < H$
Persona	10.53 %	1.28 %	88.19 %
Grupo	32.44 %	4.82 %	62.74 %
Motocicleta	0 %	0 %	100 %
Vehículo	96.32 %	0.85 %	2.83 %
Camión	93.94 %	0.4 %	5.66 %

En la Figura 2 se muestra un esquema general del método propuesto para la construcción de reglas que puedan emplearse en la construcción del sistema de clasificación de objetos de una escena. La información de entrada la constituye el flujo de vídeo comprimido en formato H.264/AVC [6] obtenido de cámaras de vigilancia; como salida se obtiene el conjunto de reglas difusas que permite clasificar los diferentes tipos de objetos de forma automática. En la primera parte de la propuesta se realiza la *adquisicion de informacion* a través de cámaras de vigilancia, se aplica un algoritmo de segmentación previamente diseñado [8] y se adapta la información resultante para su uso en las siguientes fases. En la segunda fase (*etapa de entrenamiento*) se aplica el algoritmo de clustering k-means y se construyen de manera automática los conjuntos de entrenamiento necesarios para la tercera etapa. En la *etapa de aprendizaje*, el algoritmo de aprendizaje automático aprende el conjunto de reglas difusas que caracteriza los diferentes tipos de objetos móviles a partir de los conjuntos de entrenamiento generados en la etapa anterior. De este modo, a partir de las reglas generales se puede deducir a qué tipo pertenecerá un objeto móvil en una secuencia de vídeo. Cada una de las etapas será analizada brevemente a continuación.

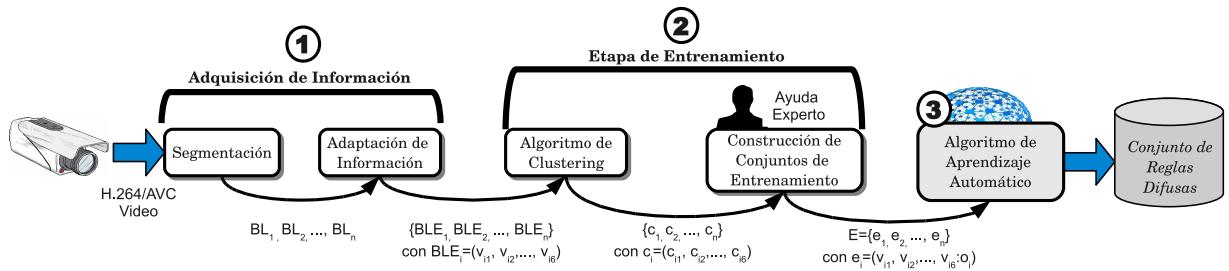


Figura 2: Esquema general del sistema propuesto.

## 2.1. ETAPA DE ADQUISICIÓN DE INFORMACIÓN

### 2.1.1. Algoritmo de segmentación

En esta etapa, se utiliza razonamiento aproximado para la segmentación de los objetos móviles en escenas de vídeo [8]. La propuesta de segmentación ha sido implementada utilizando el software H.264/AVC JM 15.1 [6]. El resultado del algoritmo es un conjunto de blobs lingüísticos o regiones [8]. Un **blob lingüístico** se define como una 7-tupla compuesta por uno o más vectores de movimiento conceptualmente similares que representan una región de manera lingüística:

$$BL = \langle NF, Tam, MB, I_{PH}, I_{PV}, I_{VH}, I_{VV} \rangle \quad (1)$$

donde  $NF$  es el número de frame en el que se localiza el blob,  $Tam$  es el número de vectores de movimiento lingüísticos [7] agrupados en el blob,  $MB$  es la lista de los macrobloques pertenecientes al blob y los cuatro últimos elementos son los intervalos lingüísticos que representan la posición y velocidad del blob: *Posicion Horizontal*, *Posicion Vertical*, *Velocidad Horizontal* and *Velocidad Vertical*. Dado que la lista de los macrobloques se almacena para cada blob ( $MB$ ), los objetos se pueden reconstruir mediante un proceso de defuzzificación para obtener su representación visual, como se observa en la Figura 3.

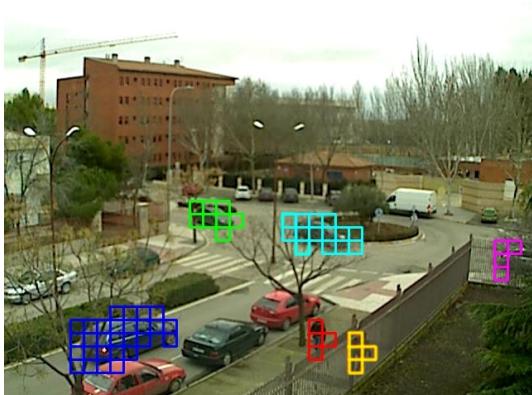


Figura 3: Ejemplo de frame segmentado.

El algoritmo de segmentación basado en razonamiento aproximado permite obtener la descripción de regiones (*blobs*) mediante características como su forma, tamaño, posición y velocidad. Estas representaciones lingüísticas caracterizan a los objetos y facilitan la clasificación de los mismos en diferentes grupos predefinidos como personas, grupos de personas, motocicletas, coches o camiones.

### 2.1.2. Adaptación de información

El algoritmo k-means es una de las técnicas más sencillas y extendidas para la agrupación de objetos basándose en particiones, donde cada partición representa un grupo o *cluster*. Sin embargo, para ser aplicado, la información de entrada debe representarse mediante un conjunto de características numéricas, por lo que el resultado del algoritmo de segmentación se debe transformar. Dicha adaptación se lleva a cabo mediante la aplicación de una función que transforma cada elemento  $BL_i$  en un nuevo elemento  $BLE_i$  definido sobre el conjunto de variables utilizadas en el algoritmo de aprendizaje. Así, un **blob lingüístico extendido**  $BLE$  es una 6-tupla que define numéricamente una región móvil:

$$BLE = \langle v_1, v_2, v_3, v_4, v_5, v_6 \rangle \quad (2)$$

donde cada elemento se obtiene del siguiente modo:

- $v_1 = \mu_{PV}^1(I_{PV})$ , es decir,  $v_1$  se obtiene mediante la defuzzificación de la variable *Posicion Vertical*.
- $v_2 = \mu_{PH}^1(I_{PH})$ , es decir,  $v_2$  se obtiene mediante la defuzzificación de la variable *Posicion Horizontal*.
- $v_3 = size(MB)$ ,  $v_3$  es el número de elementos de la lista  $MB$ .
- $v_4 = Tam$ , luego  $v_4$  es el segundo elemento del objeto  $BL$ .
- $v_5 = width(MB)/height(MB)$ , es decir,  $v_5$  se calcula como la relación entre la anchura y la altura en macrobloques del objeto si su límite se definiese dentro de un rectángulo.

- $v_6 = \text{disp}(BL)$ , donde  $\text{disp}(BL)$  es una medida de dispersión del campo de vectores de movimiento del objeto.

Para la obtención de  $\text{disp}(BL)$ , es necesario obtener previamente el vector que define el desplazamiento global del blob, denominado **vector de velocidad general**, el cual se calcula mediante la defuzzificación de las variables *Velocidad Horizontal* y *Velocidad Vertical*:

$$VVG(BL) = < \mu_{VH}^1(I_{VH}), \mu_{VV}^1(I_{VV}) > \quad (3)$$

Así, a partir de este vector, se obtiene  $\text{disp}(BL)$  calculando la suma de las diferencias entre los vectores de movimiento lingüísticos pertenecientes al blob y el correspondiente *vector de velocidad general*. Finalmente, el resultado se divide entre el número de vectores de movimiento del blob o región.

## 2.2. ETAPA DE ENTRENAMIENTO

### 2.2.1. Algoritmo de clustering

El objetivo de la primera parte de la etapa de entrenamiento consiste en generar los grupos (*clusters*) que se utilizarán para la construcción del conjunto de datos de entrenamiento. Para ello, se sugiere la utilización del algoritmo k-means. Como se ha mencionado previamente, este algoritmo será aplicado para obtener los *clusters* a los que pertenece cada tipo de objeto. Así, en esta etapa se genera un conjunto de *clusters*  $C = \{C_1, C_2, \dots, C_n\}$ , donde cada *cluster*  $C_j$  se identifica con su centroide  $c_j$ , donde  $c_j = (c_{j1}, c_{j2}, \dots, c_{j6})$  siendo cada  $c_{ji}$  el valor que toma el centroide  $j$  en la variable  $v_i$ , y es asociado a una clase de objeto  $o_j$  (tal que  $o_j \in O$ ), es decir  $C_j = (c_j : o_j)$ .

Dado que el algoritmo k-means trata de agrupar la información de características similares, una de ellas el tamaño de los objetos, aparece el problema de la perspectiva: diferentes tipos de objetos pueden tener el mismo tamaño en función de su posición con respecto a la cámara y, por tanto, ser agrupados en un mismo *cluster*. Para evitar este problema, se propone la división de la información de entrada en función de la distancia a la cámara. La Figura 4 muestra un ejemplo de escena particionada en la que se utilizan sólo divisiones horizontales; sin embargo, la escena se puede dividir de diferentes maneras en función de las recomendaciones de un experto. En este ejemplo, la escena se divide en cuatro regiones: *Fuera*, *Lejos*, *Medio* y *Cerca*. De este modo, se pueden discriminar las áreas de la imagen en las que no debería haber objetos móviles (*Fuera*) y aquellas con menor o mayor distancia a la cámara. Así, el algoritmo k-means se aplica sobre los objetos móviles de cada área y obtiene un conjunto de

*clusters* diferenciados para cada una de ellas. Después de un número determinado de iteraciones, los objetos son agrupados en torno a los centroides. El algoritmo termina cuando los centroides no se modifican en dos iteraciones consecutivas o cuando se alcanza una iteración prefijada (*poda*).



Figura 4: Ejemplo escenario particionado.

La calidad de los resultados depende de la selección inicial de los centroides. Habitualmente, el conjunto inicial de centroides se selecciona de forma aleatoria porque el algoritmo k-means tiene la habilidad de distribuir los centroides de forma eficiente si el conjunto de entrenamiento es suficientemente amplio. Esta opción es la más rápida, pero no la más eficiente. Es preferible inicializar los centroides utilizando una heurística que facilite la distribución de los grupos. En este trabajo, se utiliza un algoritmo para la selección automática de centroides iniciales basado en la variable con mayor varianza.

Una vez que se obtiene el conjunto inicial de centroides para cada partición de perspectiva, el algoritmo k-means distribuye el conjunto de entrenamiento en  $k$  grupos modificando la representación de los grupos iniciales. Como resultado, se obtiene la caracterización de los  $k$  grupos. Por ejemplo, el conjunto de grupos para la partición *Cerca* podría ser el siguiente con  $k = 5$ :

$$\begin{aligned} c_0 &- (26.5, 16.9, 10.9, 45.3, 1.05, 3.7) \\ c_1 &- (24.7, 23.4, 4.1, 8.2, 0.54, 2.8) \\ c_2 &- (26.4, 18.1, 8.3, 26.9, 1.03, 3.2) \\ c_3 &- (25.4, 4.0, 23.6, 154.8, 2.12, 4.9) \\ c_4 &- (27.2, 16.2, 4.6, 13.8, 0.96, 3.0) \end{aligned}$$

En este punto, existe un conjunto de grupos característicos para cada partición del escenario. Sin embargo, se debe etiquetar el tipo de objeto correspondiente a cada grupo en función de las propiedades que lo definen. Este proceso lo lleva a cabo un experto de forma manual. Por ejemplo, los centroides anteriormente citados podrían ser etiquetados en los siguientes grupos:  $C = \{c_0 : \text{Grupo}; c_1 : \text{Persona}; c_2 : \text{Grupo}; c_3 : \text{Vehiculo}; c_4 : \text{Persona}\}$ .

### 2.2.2. Construcción de conjuntos de entrenamiento

Una vez que los clusters han sido generados, la siguiente fase de la etapa de entrenamiento consiste en la generación del conjunto de entrenamiento  $E$  donde cada ejemplo se define como  $e_i = (v_{i1}, v_{i2}, \dots, v_{i6} : o_i)$ . Este proceso se ejecuta automáticamente mediante la comparación de cada blob lingüístico extendido  $BLE_i = (v_{i1}, v_{i2}, \dots, v_{i6})$  con cada centroide  $c_j = (c_{j1}, c_{j2}, \dots, c_{j6})$  obtenido en la fase anterior. Así, el ejemplo de entrada  $e_i$  será etiquetado con el tipo de objeto  $o_j$  del cluster  $C_j$  más cercano. La similaridad entre un blob lingüístico extendido  $BLE_i$  y un cluster  $C_j$  se calcula mediante la distancia Euclídea de las seis variables:

$$\text{distancia}(BLE_i, C_j) = \sqrt{\sum_{n=1}^6 (v_{in} - c_{jn})^2} \quad (4)$$

Por ejemplo, siendo  $BLE_0 : (26, 8, 20, 144, 2, 3)$ , la distancia al grupo  $C_3$  será:  $\text{distancia}(BLE_0, C_3) = 12,23$ . El centroide  $c_3$  asociado al grupo  $C_3$  es el más cercano a  $BLE_0$ , por lo que se puede concluir que el ejemplo  $e_0$  tendrá las seis variables  $v_i$  del blob  $BLE_0$  y se le asignará como objeto  $o_0$  el valor *Veh culo* dado que  $C_3$  ha sido etiquetado como grupo *Veh culo*. De este modo, los objetos detectados en el algoritmo de segmentación podrían ser usados en la construcción de los conjuntos de entrenamiento.

### 2.3. ETAPA DE APRENDIZAJE

Para el aprendizaje de la función  $S$  a partir del conjunto de entrenamiento  $E$  se va a hacer uso del algoritmo de aprendizaje automático presentado por Castro et al. [4]. Dicho algoritmo trabaja en tres fases. La primera de ellas consiste en la división de los dominios de las variables de entrada (en este caso, son seis). Cada uno de los dominios de las variables de entrada se divide en varias regiones, cada una de las cuales será presentada por medio de una función trapezoidal. La elección de las regiones que conforman tal división se podrá hacer de forma manual en la que un experto hace la división del dominio, determinando cuantos valores puede tomar cada variable y la definición de cada valor, o de manera automática empleando algunas heurísticas para hacer la mejor división del dominio posible.

En la segunda fase, cada ejemplo del conjunto de entrenamiento se convierte en una regla inicial o básica, la cual carece de toda generalidad. De hecho, esta regla inicial únicamente especifica cómo clasificar un ejemplo muy concreto o un conjunto de ejemplos muy próximos, es decir, cómo clasificar el ejemplo o

ejemplos a partir del cual se ha generado la regla. Por tanto, en el peor de los casos, existirán tantas reglas particulares como ejemplos haya en el conjunto de entrenamiento. En cada una de estas reglas, cada valor de cada variable de entrada será presentado por medio de una región de su dominio y la salida del ejemplo será codificada.

Finalmente, la tercera fase tiene como objetivo reducir el conjunto de reglas iniciales para construir reglas más generales. Cada una de las reglas generales abarca varias reglas particulares o iniciales. Para ello se utiliza un proceso de ampliación en el que una regla inicial se extiende, con valores del dominio aún no considerados, cubriendo nuevas situaciones del espacio global de posibles situaciones, para las cuales no existen contraejemplos, es decir no están cubiertas por reglas iniciales que clasifiquen de distinta forma. Para una especificación detallada del algoritmo ver [4] y una modificación del mismo aplicado a la adquisición de conocimiento en entornos monitorizados en [1]. Esta modificación ha sido necesaria para evitar un problema de sobregeneralización que se produce con la propuesta inicial y que en este ámbito tiene efectos no deseados. A continuación se muestran ejemplos de reglas que han sido aprendidas con el método propuesto:

1. SI  $v_1$  es {CentroVertical,Arriba} y  $v_3$  es {Grande} y  $v_6$  es {Pequeño} ENTONCES  $o_i$  es *Veh culo*.
2. SI  $v_1$  es {Abajo} y  $v_5$  es {MuyPequeño, Pequeño} y  $v_6$  es {Grande, MuyGrande} ENTONCES  $o_i$  es *Persona*.

### 3. RESULTADOS

El sistema de clasificación ha sido evaluado en secuencias de vídeo comprimido con formato H.264/AVC utilizando el codificador J.M. 15.1. La configuración del codificador ha sido fijada con perfil *baseline* y modo de optimización RD *low complexity* [6]. La resolución de los vídeos es de 640x480 píxeles. En la Figura 5 se muestra una captura de una secuencia de vídeo en la que los blobs aparecen dibujados en el frame y etiquetados en la parte derecha de la imagen<sup>1</sup>.

En la Tabla 2 se muestra una comparativa de los resultados obtenidos en dos secuencias de vídeo para la clasificación de objetos a partir de los clústers generados en la etapa de entrenamiento. En este caso, cada objeto se clasifica del tipo perteneciente al clúster con menor distancia (Ecación 4). Se denomina *probabilidad de detección* a la probabilidad de que un objeto

<sup>1</sup>Se ha desarrollado un sistema web para la clasificación automática de objetos en una secuencia de vídeo; se encuentra disponible en: <http://oreto.esi.uclm.es/aplicaciones/labelling/>



Figura 5: Ejemplo de la aplicación web.

móvil sea detectado y *precision* a la probabilidad de que un objeto detectado esté bien etiquetado. Utilizando este sistema de clasificación, el experto sólo ha de seleccionar a qué tipo de objeto pertenece cada clúster, sin embargo, la interpretabilidad de los resultados es más deficiente que si se usa el conjunto de reglas difusas generado en la fase de aprendizaje.

Cuadro 2: Precisión y probabilidad de detección.

	Vídeo 1	Vídeo 2
Frames	1375	586
Objetos reales	5995	897
Objetos detectados	4958	822
Prob. detección	82.7 %	91.6 %
Precisión	87.3 %	93.1 %

## 4. CONCLUSION

Se ha presentado un método para la construcción de un conjunto de reglas que puedan ser usadas en la clasificación de objetos, basado en un algoritmo de aprendizaje supervisado. El conjunto de reglas construido determina el tipo al que pertenece cada objeto detectado en una escena. El conjunto de entrenamiento que toma como entrada el algoritmo de aprendizaje se genera de forma automática utilizando técnicas de clustering en la etapa de entrenamiento. Finalmente, hay que destacar que las reglas aprendidas podrían ser usadas en tiempo real.

Como trabajo futuro, se propone dividir la perspectiva del escenario de forma difusa para evitar las imprecisiones que puede acarrear la agrupación de los objetos en tres sub-áreas predefinidas. Por otro lado, se propone el estudio de las ventajas que conlleva la aplicación del algoritmo de aprendizaje supervisado con respecto a la completitud y validez del conjunto de reglas generado.

## Agradecimientos

Este trabajo ha sido financiado por la Consejería de Educación y Ciencia de Castilla-La Mancha a través de los proyectos FEDER PII2I09-0052-3440 y PII1C09-0137-6488.

## Referencias

- [1] J. Albusac, J.J. Castro-Schez, L. Lopez-Lopez, D. Vallejo, L. Jimenez-Linares. A supervised learning approach to automate the acquisition of knowledge in surveillance systems. *Journal of Signal Processing*, vol. 89, nº. 12, pp. 2400-2414, 2009.
- [2] J. Albusac, D. Vallejo, L. Jimenez, J.J. Castro-Schez, L. Rodriguez. Intelligent surveillance based on normality analysis to detect abnormal behaviours. *Int. J. of Pattern Recognition and AI*, vol. 23, nº. 7, pp. 1223-1244, 2009.
- [3] L.M. Brown. View independent vehicle-person classification. *Proc. ACM 2nd Int. Workshop on Video Surveillance*, pp. 114-123, 2004.
- [4] J.L. Castro, J.J. Castro-Schez, J.M. Zurita. Learning maximal structure rules in fuzzy logic for knowledge acquisition in expert systems. *Fuzzy Sets and Systems*, vol. 101, nº. 3, pp. 331-342, 1999.
- [5] 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*, vol. 34, nº. 3, pp. 334-352, 2004.
- [6] Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG. Reference Software to Committee Draft JVT-F100 JM15.1, 2008.
- [7] L. Rodriguez-Benitez, J. Moreno, J.J. Castro-Schez, J. Albusac, L. Jimenez. Automatic objects behaviour recognition from compressed video domain. *Image and Vision Computing*, vol. 27, nº. 6, pp. 648-657, 2009.
- [8] C.J. Solana-Cipres, G. Fernandez-Escribano, L. Rodriguez-Benitez, J. Moreno, L. Jimenez. Real-time moving object segmentation in H.264 compressed domain based on approximate reasoning. *International Journal of Approximate Reasoning*, vol. 51, nº. 1, pp. 99-114, 2009.
- [9] M. Valera, S.A. Velastin. Intelligent distributed surveillance systems: a review. *IEE Proc. Vision, Image and Signal Processing*, vol. 152, nº. 2, pp. 192-204, 2005.