



## RECONOCIMIENTO DE OBJETIVOS EN IMÁGENES DE VIDEO UTILIZANDO VISIÓN ARTIFICIAL

**Reynaldo Rafael Mendoza Iglesias<sup>a</sup>**  
rmendoza@cotecmar.com

**Mario de la Fuente Martín<sup>b</sup>**  
mariodlf@et.mde.es

*Fecha de recepción: agosto de 2014. Fecha de aceptación: octubre de 2014*

**Resumen:** Se presenta el procedimiento llevado a cabo para el reconocimiento de objetivos en imágenes de video en tiempo real. El trabajo se basa en el reconocimiento de patrones mediante el uso de tecnologías de visión por computador por medio de la librería de código abierto OpenCV (Open source Computer Vision) para el tratamiento de imágenes y entrenamiento de clasificadores de objetos. A lo largo del trabajo se profundizará en los aspectos técnicos de entrenamiento del clasificador de objetos describiendo el procedimiento realizado para su entrenamiento como los procesos de adquisición de imágenes y otras actividades relacionadas con todo el proceso de aprendizaje y clasificación. Para finalizar, se presentarán los resultados y conclusiones obtenidas a partir del resultado de varios clasificadores con distintos parámetros de entrenamiento. El funcionamiento de los clasificadores entrenados fue verificado con una aplicación desarrollada en el lenguaje de programación Java.

**Palabras clave:** reconocimiento de patrones, visión artificial, visión por computador, clasificador de objetos, OpenCV.

**Abstract:** This article describes the procedure carried out for the target recognition in video images in real time. The work is based on pattern recognition by using computer vision technologies through the open source library OpenCV (Open Source Computer Vision) for image processing and training of object classifiers. Throughout the paper we will delve into the technical aspects for the training of the object classifiers, describing the procedure done to its training as the process of image acquisition and related activities about the learning process and classification. Conclusions are presented from the result of several trained classifiers with different training parameters. The performance of the trained classifiers was verified with a software application developed in Java programming language.

**Keywords:** pattern recognition, artificial vision, computer vision, object classifier, OpenCV.



<sup>a</sup>Ingeniero Electrónico. Máster en Tecnologías para la Defensa, Universidad Politécnica de Madrid. Diseñador de la División de Electricidad y Electrónica – Gerencia de Diseño e Ingeniería de Cotecmar, Cartagena de Indias, Colombia. Ingeniero.

<sup>b</sup>Técnico en Telecomunicaciones. Especialidad equipos electrónicos, Universidad Politécnica de Madrid. Capitán del Cuerpo de Ingenieros Politécnicos del Ejército de Tierra del Reino de España, Madrid, España.

## 1. INTRODUCCIÓN

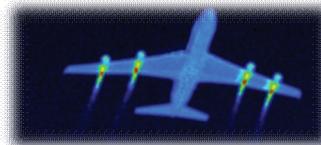
La visión artificial o visión por computador es considerada un campo de la inteligencia artificial cuyo objetivo final es utilizar los computadores para emular la visión humana, incluyendo el aprendizaje y la capacidad de hacer inferencias y tomar acciones basadas en entradas visuales (González, R. y Woods, R., 2002).

El reconocimiento de patrones utilizado en el presente trabajo es el punto de partida para el reconocimiento de objetivos que constituye un aspecto fundamental para el desarrollo de actividades de vigilancia en el entorno civil y militar. Está siendo utilizado en diversas aplicaciones como control de accesos (login al computador, ingreso a instalaciones), identificación (atentados, disturbios, personas desaparecidas, reconocimiento de amenazas), entre otras. Por otro lado, el seguimiento de objetos por imágenes también se ha extendido para la realización de vigilancia remota y rastreo de objetos. Las Tecnologías de procesamiento y análisis de la señal de video son complemento al trabajo que pueda realizar un operador.

Existen desarrollos tecnológicos de reconocimiento de objetivos cuyos costos y capacidades están relacionados con la característica utilizada para llevar a cabo dicho reconocimiento, debido a que determina los sensores y/o requerimientos de procesamiento y cálculo. Algunas características utilizadas para el reconocimiento de objetivos son: la firma espectral, la sección transversal de radar, los patrones acústicos, patrones de imágenes, entre otros (ver ). El presente trabajo muestra los resultados del uso de patrones en imágenes como característica para reconocer objetivos empleando software de código abierto de visión por computador y entrenamiento de clasificadores.



Fuente: (Sinodefenceforum, 2011)



Fuente: (Overton, G, 2011)

**Figura 1.** Reconocimiento de objetivos  
Izquierda: a partir de su patrón de imagen.  
Derecha: a partir de su firma infrarroja



El trabajo contempla el entrenamiento de clasificadores para el reconocimiento de cualquier objeto deseado, el trabajo ha sido desarrollado en este caso de uso para el reconocimiento de un robot móvil Summit XL de la empresa Robotnik, basado en las herramientas de software gráfico y de visión por computador, en concreto se ha trabajado con la librería OpenCV.

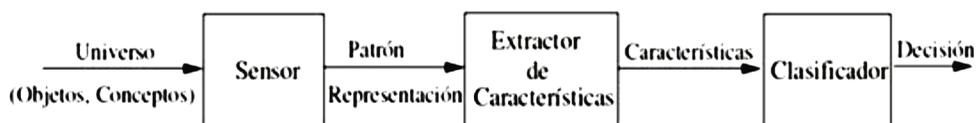
OpenCV es un conjunto de librerías que proporcionan funciones y programas completos relacionados con la visión por computador, siglas de las cuales deriva su nombre (Open source Computer Vision). OpenCV suministra una demostración de clasificadores para detección de rostros, ojos, sonrisas, entre otros. Adicionalmente suministra programas o funciones que nos permiten entrenar clasificadores para generar nuestro propio clasificador para la detección de un objeto deseado. También cuenta con una documentación detallada donde realizar consultas.

### 1.1. Clasificador de objetos

Los clasificadores de objetos hacen parte de una de las etapas que conforman un sistema de reconocimiento de patrones (ver ) y su objetivo es realizar la toma de decisiones del sistema.

*Figura 2. Etapas de un sistema de reconocimiento de patrones.*

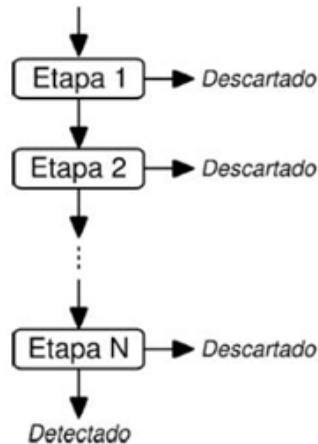
*Fuente: (Kittler, J, 2002).*



La clasificación puede ser supervisada si se tiene un conjunto de muestras para entrenar al clasificador o no supervisada si no existe entrenamiento previo. En el desarrollo del trabajo se hizo uso de un algoritmo de aprendizaje supervisado conocido como boosting que funciona entrenando clasificadores débiles en forma consecutiva y los combina en lo que se llama un clasificador fuerte, a este proceso se le conoce como "aprendizaje de un clasificador". Durante la etapa de entrenamiento de un detector de objetos que usa el algoritmo boosting se crean varios clasificadores fuertes que permiten decidir cuales regiones de la imagen contienen el objeto. Luego se organizan los clasificadores fuertes en una estructura en cascada (ver ) para mejorar la eficiencia del detector.



**Figura 3.** Cascada de clasificadores fuertes por Viola y Jones.  
Cada etapa está compuesta por un clasificador fuerte.  
Fuente: (Morelli, A. y Padonavi S, 2011).



## 1.2. Características HAAR y LBP

Una manera de clasificar imágenes es compararla pixel a pixel con un conjunto de imágenes ya conocido y clasificarla como la clase de la imagen más parecida, esta opción no es eficiente, por lo que se utilizan otras técnicas mucho más eficientes como la extracción de características o “features” del objeto.

La característica o “feature” se refiere al resultado de aplicar una operación en un área determinada de la imagen y su vecindad o a un patrón dentro de la imagen.

Para formar clasificadores, Viola y Jones (2004) evalúan las características o “features” de la imagen para decidir la presencia de un determinado objeto. Los clasificadores son construidos a partir de una variante de features que se conocen como “Haar-like features”, el nombre es debido a que el resultado es similar a aplicar filtros de Haar sobre la imagen. Un ejemplo de los filtros Haar se muestran a continuación y su cálculo consiste en hallar la diferencial entre la integral de la imagen en las zonas blancas y la integral de la imagen en las zonas negras.

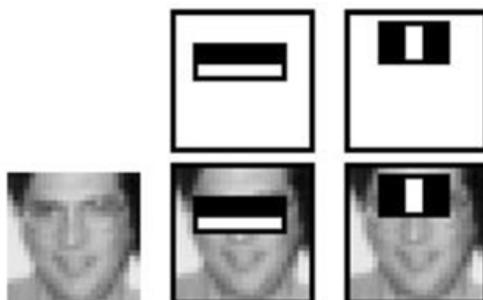
**Figura 4.** Tipos de features usados por Viola y Jones.  
Fuente: (Morelli, A. y Padonavi S, 2011).

$$\mathbb{F} = \left\{ F_1 : \begin{array}{|c|} \hline \blacksquare \\ \hline \end{array}, F_2 : \begin{array}{|c|} \hline \square \\ \hline \end{array}, F_3 : \begin{array}{|c|} \hline \blacksquare \square \\ \hline \end{array}, F_4 : \begin{array}{|c|} \hline \square \blacksquare \\ \hline \end{array}, F_5 : \begin{array}{|c|} \hline \blacksquare \square \\ \hline \end{array} \right\}$$



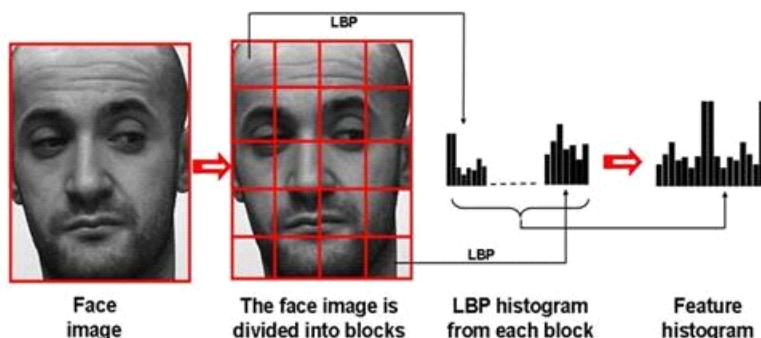
Una aplicación de los features Haar es en la captura de las características presentes en los rostros: oscurecimiento de la imagen en la zona de los ojos respecto a la zona de las mejillas, aclaramiento en la zona del puente nasal, etc. La siguiente figura muestra el concepto:

**Figura 5.** Características que puede capturar un feature.  
Fuente: (Viola, P. y Jones, M, 2004).



Otro tipo de feature utilizada es la Local Binary Patterns - LBP (patrones binarios locales) en la que se obtiene el vector de características de la imagen a través del histograma de las celdas que contienen los pixeles, que a su vez se obtienen dividiendo la imagen examinada en celdas que contienen los pixeles, cada pixel es comparado con sus ocho vecinos y se le asigna un valor de uno o cero por lo que se obtiene un numero binario de ocho dígitos. El vector de características obtenido puede ser procesado usando Support Vector Machine – SVM (máquinas de vectores de estados) o un algoritmo de aprendizaje de clasificación de imágenes (Pietikäinen, M. y Zhao, G., 2009). La muestra la descripción de un rostro usando LBP.

**Figura 6.** Descripción de rostro con patrones locales binarios.  
Fuente: (Pietikäinen, M. y Zhao, G., 2009).



## 2. METODOLOGÍA

### 2.1. Entrenamiento del clasificador

Para el entrenamiento de los clasificadores se hizo uso de:

- Un robot móvil tipo Summit XL de la empresa Robotnik como el objetivo a reconocer (ver).
- Un dron aéreo tipo AR Drone 2.0 de la empresa Parrot para adquirir las imágenes positivas y negativas requeridas para el entrenamiento del clasificador (ver ). Imágenes positivas (que contienen el robot móvil) y negativas (no contienen el robot móvil) en entornos abiertos con distintos fondos (grava, asfalto, arena, vegetación) y en recintos cerrados (hangar) (ver ).

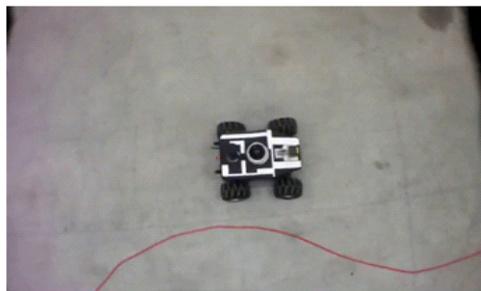
**Figura 7.** Robot móvil tipo Summit XL.  
Fuente: (Robotnik, 2014)



**Figura 8.** Dron AR 2.0.  
Fuente: (Parrot, 2014)



**Figura 9.** Imagen positiva del robot tomada desde el dron en fondo de asfalto.



Para el desarrollo del clasificador se utilizaron imágenes en formato Jpg de 1280\*720 pixeles.

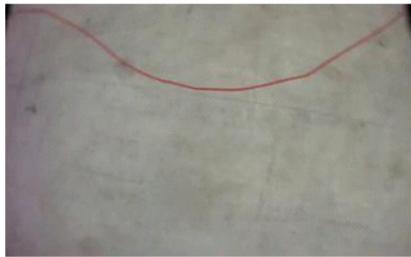
Se utilizaron distintos fondos:

- Grava: FnG
- Vegetación: FnV
- Camino de arena. FnC
- Asfalto en interior: FnI
- Asfalto en exterior: FnA



La figura 10 y la figura 14 muestran fotografías de los distintos fondos utilizados.

A partir de las imágenes positivas y negativas y con las herramientas que brinda OpenCV se realizó el entrenamiento del clasificador. OpenCV cuenta con dos aplicaciones para el entrenamiento de clasificadores en cascada: `opencv_haartraining` y `opencv_traincascade`. `Opencv_traincascade` es una versión más reciente y soporta características Haar y LBP.



**Figura 10.** Ambiente interior: asfalto



**Figura 11.** Ambiente exterior: asfalto



**Figura 12.** Ambiente exterior: grava



**Figura 13.** Ambiente exterior: vegetación.



**Figura 14.** Ambiente exterior: camino de arena.

La información utilizada y generada para la obtención del clasificador se describe a continuación:



### Imágenes positivas

La siguiente tabla describe las imágenes positivas utilizadas

**Tabla 1.** Imágenes positivas. El patrón A corresponde al robot móvil.

Nomenclatura				Fondo (Fn)	Patrón (Pt)	Luz (Lz)	Horario (Hr)	Cantidad
FnI	PtA	Lz0	Hr0	Interior	A	N/A	N/A	127
FnA	PtA	LzS	HrM	Asfalto	A	Sol	Mañana	40
FnC	PtA	LzS	HrM	Camino	A	Sol	Mañana	297
FnG	PtA	LzS	HrM	Grava	A	Sol	Mañana	230
FnV	PtA	LzS	HrM	Vegetación	A	Sol	Mañana	517
<b>TOTAL</b>								1211

### Imágenes negativas

La siguiente tabla describe las imágenes negativas utilizadas

**Tabla 2.** Imágenes negativas.

Nomenclatura				Fondo (Fn)	Patrón (Pt)	Luz (Lz)	Horario (Hr)	Cantidad
FnI	Pt0	Lz0	Hr0	Interior	N/A	N/A	N/A	152
FnA	Pt0	LzS	HrM	Asfalto	N/A	Sol	Mañana	20
FnC	Pt0	LzS	HrM	Camino	N/A	Sol	Mañana	442
FnG	Pt0	LzS	HrM	Grava	N/A	Sol	Mañana	108
FnV	Pt0	LzS	HrM	Vegetación	N/A	Sol	Mañana	416
<b>TOTAL</b>								1138

### Archivo de regiones positivas

A partir de las imágenes positivas se generó un archivo txt con lista de regiones positivas conteniendo para cada imagen: ubicación (ruta), nombre, número de objetos de interés en la imagen, posición del objeto en la imagen (x, y), ancho y alto del objeto (w, h). Para esto se utiliza object marker de opencv, que permite seleccionar el objeto de interés dentro de la imagen.



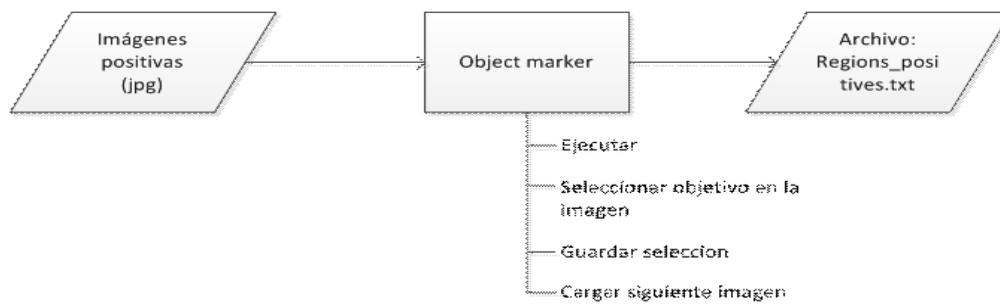


Figura 15. Generación de archivo de regiones positivas.

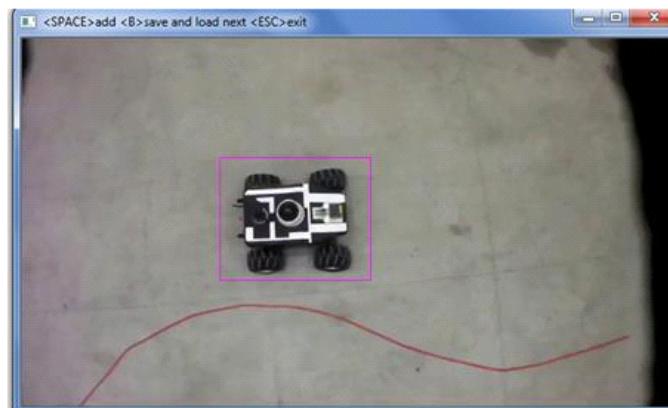


Figura 16 . Object marker: imagen cargada con objetivo seleccionado.

La siguiente figura muestra un ejemplo del archivo de regiones positivas generado:

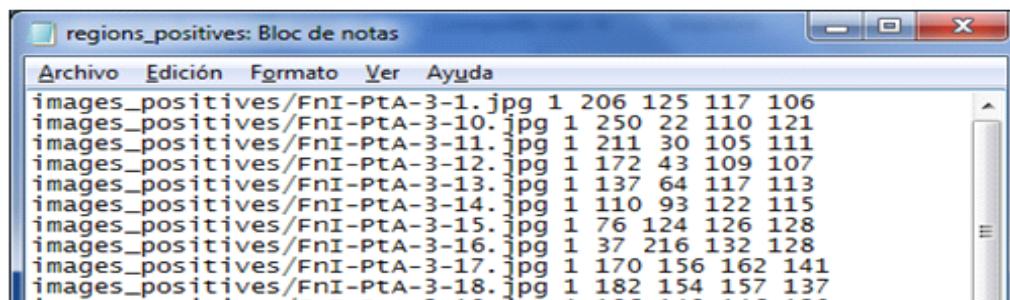
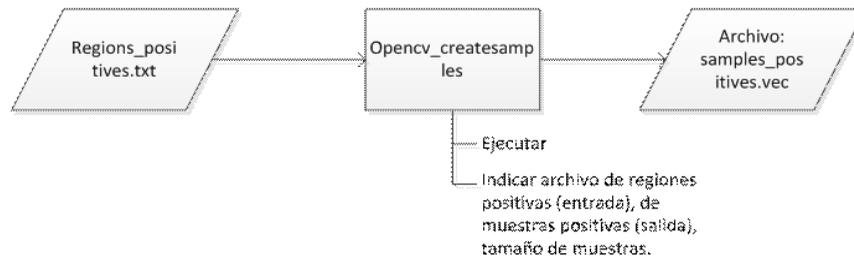


Figura 17. Archivo de regiones positivas.



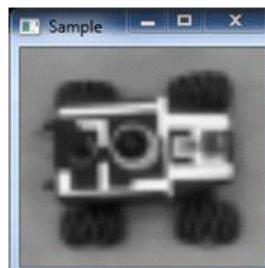
### Archivo de muestras positivas

A partir del archivo de regiones positivas se genera un archivo vec que contiene las muestras positivas para el entrenamiento. Para esto se utiliza opencv\_createsamples



**Figura 18 .** Generación de archivo de muestras positivas.

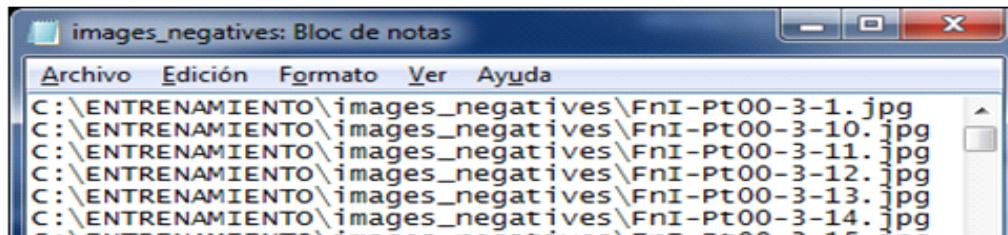
En la siguiente figura se puede observar una muestra generada:



**Figura 19.** Muestra positiva.

### Archivo de imágenes negativas

Se debe crear un archivo txt con lista de imágenes negativas conteniendo para cada imagen: ubicación (ruta), nombre. La siguiente figura muestra un ejemplo del archivo de imágenes negativas generado:



**Figura 20.** Archivo de imágenes negativas.



### Generación del clasificador

El clasificador es entrenado a partir del archivo de muestras positivas vec y el de imágenes negativas. El resultado son archivos XML.

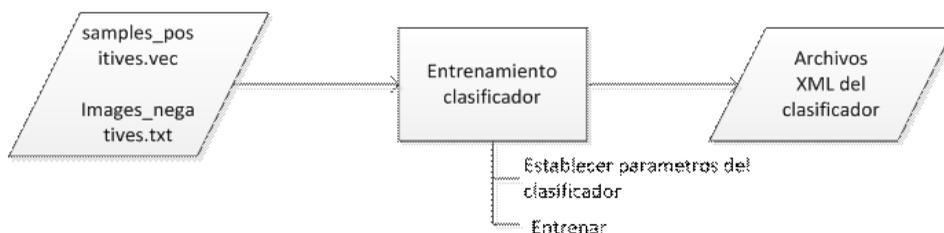


Figura 21 . Generación de clasificador.

A continuación se describen los parámetros más relevantes utilizados en el entrenamiento:

- numPos*: Entero con el número de muestras positivas usadas para una etapa del clasificador. Debe ser menor al que contiene el archivo VEC, se utilizó aproximadamente el 90% de muestras que contiene el archivo VEC.
- numNeg*: Entero con el número de muestras negativas usadas para una etapa del clasificador. Puede ser igual o mayor a la cantidad de imágenes negativas disponibles.
- numStages*: Numero de etapas en cascada con las que debe ser entrenado.
- featureType*: tipo de feature utilizado durante el entrenamiento: HAAR, LBP.
- minHitRate*: Tasa mínima de aciertos deseada para cada etapa del clasificador. Si se selecciona un valor de 1 se impone que no se aceptan falsos negativos durante el entrenamiento.
- maxFalseAlarmRate*: Tasa de falsos positivos máxima deseada para cada etapa del clasificador. Si se selecciona un valor de cero se impone que el objeto es detectado correctamente en todas las imágenes negativas.

La tabla 3 muestra los parámetros usados para el entrenamiento de los clasificadores.

Tabla 3. Parámetros para entrenamiento de clasificadores

Clasificador No.	Num Pos	Num Neg	Num Stages	Feature Type	minHit Rate	MaxFalse AlarmRate	Fondo us: Pos/Ne:
1	127	152	12	LBP	0,995	0,1	Fn/Fn
2	127	152	9	HAAR	0,995	0,1	Fn/Fn
3	127	152	16	LBP	0,995	0,1	Fn/Fn
4	800	1138	9	LBP	0,995	0,1	Todos/To
5	700	1138	9	HAAR	0,995	0,1	Todos/To
6	780	1138	11	LBP	0,995	0,2	Todos/To
7*	1000	5000	13	LBP	0,997	0,3	Todos/To
8*	1000	1138	11	LBP	0,995	0,2	Todos/To



Los clasificadores 7 y 8 fueron entrenados considerando 200 imágenes de vegetación adicionales respecto a los anteriores.

## 2.2. Pruebas del clasificador

Una vez se dispone del fichero XML resultado del entrenamiento del clasificador, se verificó su rendimiento para cada fondo de la siguiente manera:

### Procedimiento

1. Se utilizó aproximadamente un 10% de imágenes positivas y negativas de las utilizadas en el entrenamiento, estas imágenes son diferentes a las utilizadas en el entrenamiento (imágenes de prueba).
2. A cada imagen de prueba se le aplicó el clasificador entrenado y se verificó para cada una:
  - a. **Para las imágenes positivas:**
    - o Detección correcta - verdadero positivo: objetivo presente y detectado.
    - o Detección incorrecta - Falsos negativos: objetivo presente no detectado.
    - o Detección incorrecta - Falsos positivos: objetivo ausente y detectado.
  - b. **Para las imágenes negativas:**
    - o Detección correcta - verdadero negativo: objetivo ausente no detectado.
    - o Detección incorrecta - Falsos positivos: objetivo ausente y detectado.

### Rendimiento de los clasificadores

1. Se obtiene la tasa de acierto para el conjunto imágenes de prueba positiva y negativa de cada fondo. La tasa de acierto para imágenes positivas es calculada como:

$$TAP = DC / TIP \quad 1)$$

*TAP = Tasa de acierto de imágenes positivas.*

*DC = Detecciones Correctas*

*TIP = No. total imágenes positivas de prueba.*

La tasa de acierto para las imágenes negativas se calcula de forma similar.

1. Un acierto en una imagen positiva (objeto presente) permite validar 2 aspectos:
  - a. Que no hay falsos negativos porque el objeto es detectado con éxito.
  - b. Que no hay falsos positivos porque solo es detectado el objeto de interés.
- Un acierto en una imagen negativa (objeto ausente) permite validar 1 solo aspecto:
  - a. Que no hay falsos positivos porque no es detectado el objeto.



Los 3 aspectos permiten garantizar una precisión del 100%. De esta manera, el acierto de una imagen positiva (validación de 2 aspectos) permite garantizar un 67% de la precisión, mientras el acierto de una imagen negativa (validación de 1 aspecto) permite garantizar un 33%. Esta es la ponderación asignada a cada tasa de acierto.

El rendimiento R para cada fondo es calculado de la siguiente forma:

$$R = (0,67 \times TAP + 0,33 \times TAN) \times 100\% \quad (2)$$

TAP = Tasa de acierto de imágenes positivas.

TAN = Tasa de acierto de imágenes negativas.

### Criterios de aceptación

Se considera un valor aceptable de precisión del clasificador si el rendimiento es igual o mayor al 80%. Este valor establece que si los aciertos en imágenes negativas son del 100%, en las positivas debe ser del 70% y si los aciertos en imágenes positivas son del 100%, en las imágenes negativas debe ser del 40%.

## 3. RESULTADOS

### Evaluación de clasificadores

A continuación se muestra una parte del resultado de las pruebas realizadas a un clasificador entrenado para obtener su rendimiento cuando el fondo es asfalto en exterior (FnA). El título 'Clasificador No.' de la tabla coincide con el clasificador del mismo ítem en la tabla 3.

Se puede observar que el rendimiento del clasificador No. 1 es bajo: 14% para detectar correctamente el objeto en imágenes positivas y del 0% para no detectarlo en imágenes negativas.

Tabla 3. Reporte de pruebas de clasificador.

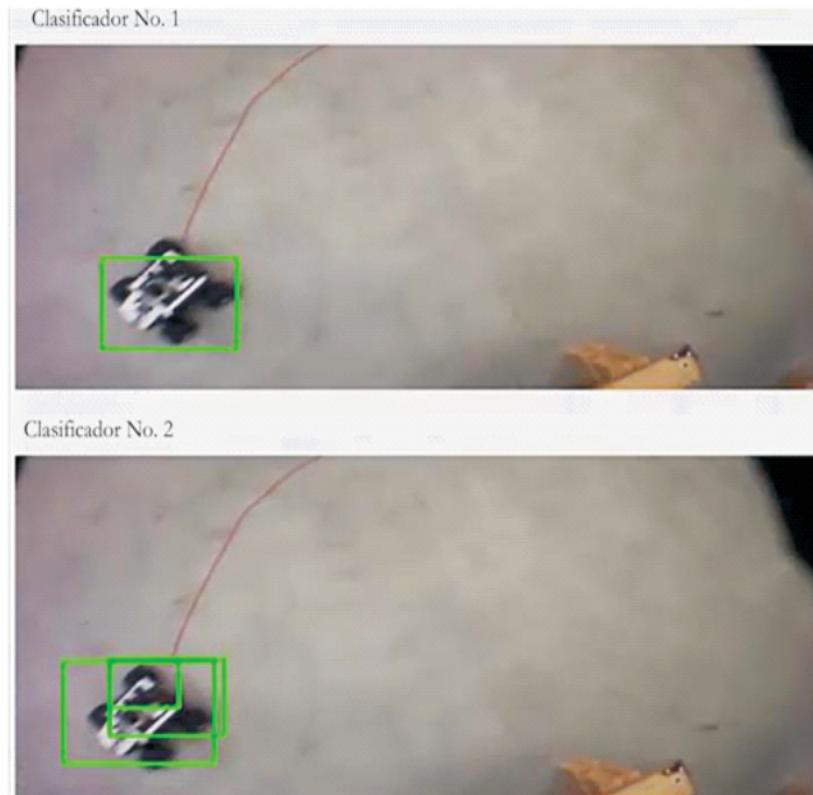
	Imágenes positivas				Imágenes negativas		
	Image n No.	Correcta	Incorrecta		Image n No.	Correcta	Incorrecta
			Falso Negativo	Falso positivo			Falso positivo
<b>Clasificador No.:1</b> <b>Fondo: FnA</b>	1			x	1		x
	2		x	x	2		x
	3			x	3		x
	4		x	x	4		x
	5		x	x	5		x
	6	x					
	7		x	x			



Resumen FnA	7	1	6	5	0	5
Tasa Aciertos	0,14			0		
Rendimiento FnA (%)	9					

Las siguientes figuras muestran los resultados de las pruebas realizadas a los clasificadores No. 1 y 2 funcionando en un entorno cerrado (FnI) y abierto (FnA) respectivamente, y en el mismo fondo (asfalto).

El clasificador No. 1 funciona mejor que el No.2 en entorno cerrado, el No. 2 detecta más falsos positivos.



**Figura 22.** Funcionamiento de clasificadores en entorno cerrado.

En entorno abierto el clasificador No. 2 funciona mejor que el No.1, en este caso el No.1 detecta más falsos positivos.



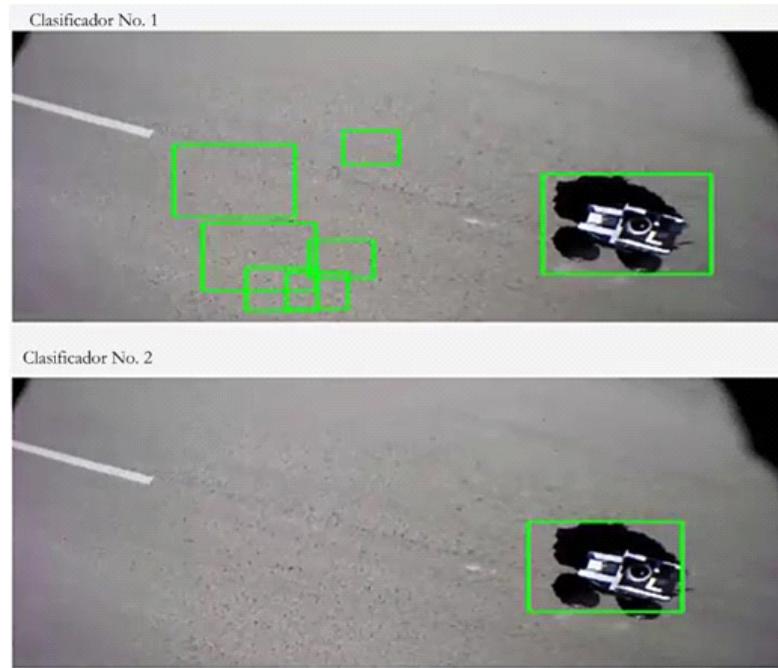


Figura 23. Funcionamiento de clasificadores en entorno abierto.

### Resumen de resultados

La tabla a continuación resume el rendimiento de los clasificadores entrenados. Se describe el rendimiento de cada fondo resaltando los que superan el criterio de aceptación establecido (mayor a 80%).

Tabla 5. Rendimiento de clasificadores entrenados.

Clasif No.	Rendimiento (%)					Stages-feature	nPos/nNeg	HitRate/falseAlarm Rate	Fondo
	Fnl	FnA	FnC	FnG	FnV				
1	100	9	31	0	6	12 - LBP	127/152	0,995/0,1	Fnl
2	95	91	81	33	32	9 - HAAR	127/152	0,995/0,1	Fnl
3	89	13	39	7	32	16 - LBP	127/152	0,995/0,1	Fnl
4	95	100	91	91	83	9 - LBP	800/1138	0,995/0,1	Todos
5	95	100	100	97	75	9 - HAAR	700/1138	0,995/0,1	Todos
6	100	100	87	97	71	11 - LBP	780/1138	0,995/0,2	Todos
7	89	100	93	94	88	13 - LBP	1000/5000	0,997/0,3	Todos
8	98	91	42	27	31	11 - LBP	1000/1138	0,995/0,2	Todos



### 3.1. Análisis de resultados

Los parámetros de entrenamiento deben ser establecidos de manera equilibrada para evitar obtener un clasificador que funcione en los extremos, es decir, muy riguroso o muy débil. Los valores dependen de cada aplicación específica principalmente del patrón que se desea detectar y del fondo o entorno donde será detectado.

#### Fondo de imágenes de entrenamiento

El entorno o fondo en los que se requiere detectar el objetivo debe ser considerado en el entrenamiento del clasificador: los clasificadores entrenados con imágenes solo de interior funcionan bien en este entorno, pero no en entorno exterior. El clasificador No. 5 (entrenado con imágenes interior y exterior) tiene mejor rendimiento en exterior que el No. 2 (entrenado con imágenes solo de interior).

#### Etapas de entrenamiento

En cada etapa de entrenamiento se obtiene más precisión del clasificador (debido a que en la siguiente etapa se re-evalúan imágenes), pero se debe tener en cuenta que establecer muchas etapas puede 'sobre entrenar' el clasificador y no detectar el objetivo. Las etapas del clasificador deben ser seleccionadas acuerdo a la cantidad de imágenes disponibles para el entrenamiento y considerando las tasas de acierto y falsa alarma que se requieran. El clasificador No.1 ha sido entrenado con 12 etapas y presenta un rendimiento levemente mayor al No.3 que ha sido entrenado con 16 etapas.

#### Tasa de acierto (hitRate)

Determina el rechazo de falsos negativos en las imágenes positivas, un valor muy alto de este parámetro genera un clasificador estricto o riguroso, contrario al caso de establecer un valor muy bajo (genera un clasificador débil), en ambos casos se puede obtener un clasificador que no detecte el objetivo correctamente.

#### Tasa de falsa alarma (falseAlarmRate)

En cada etapa se agregan características (features) y los falsos positivos decrecen hasta alcanzar la tasa de falsa alarma. Un valor muy alto de este parámetro conlleva a tener un clasificador que detecte una gran cantidad de falsos positivos.

#### LBP Vs HAAR

El entrenamiento con HAAR requiere más tiempo de procesamiento que el LBP, pero es más preciso. Se puede observar que el clasificador No. 2 a pesar de tener menos etapas que el No. 1, tiene un mayor rendimiento que el No. 1 en asfalto y uno muy similar en interior.



#### 4. CONCLUSIONES

De acuerdo a los resultados obtenidos, se puede afirmar que las herramientas informáticas de código abierto que actualmente existen destinadas al procesamiento digital de imágenes y a la visión por computador permiten realizar aplicaciones con un muy buen rendimiento y calidad, lo que puede conllevar a que algunas de estas herramientas sean de uso dual (ámbito civil y militar) con las modificaciones pertinentes requeridas para cada campo de aplicación.

Tal como se comentó en el análisis de resultados de los clasificadores de objetos, para la obtención de un clasificador óptimo se debe partir del patrón de interés, el entorno donde será reconocido y las imágenes positivas y negativas (cantidad, fondo) que son usadas para el entrenamiento. Se pudo observar que no hay parámetros establecidos o formulaciones matemáticas que permitan obtener los mejores valores para el entrenamiento, y que estos dependen de la aplicación específica.

El presente trabajo puede ser utilizado como base de partida o guía para el desarrollo de sistemas que requieran el reconocimiento de objetivos de interés civil o militar haciendo uso de tecnología de visión por computador, el procedimiento descrito para el entrenamiento de clasificadores permite generar un clasificador para el reconocimiento de un nuevo objetivo de interés utilizando imágenes positivas de dicho objetivo y, como ya se ha mencionado, imágenes negativas preferiblemente del entorno de operación.

Como trabajo futuro resultaría interesante investigar un modelo o formulación matemática que permita establecer o brindar una aproximación de los parámetros de entrenamiento de los clasificadores de objetos considerando el patrón y el entorno o fondo de operación. Adicionalmente, OpenCV cuenta con una aplicación para determinar de manera rápida con imágenes de prueba el rendimiento de los clasificadores en cascada que han sido entrenados utilizando la versión antigua de entrenamiento (`opencv_haartraining`), por lo que se plantea trabajar o profundizar en la generación de una aplicación que permita determinar este rendimiento para clasificadores entrenados con la versión nueva (`opencv_traincascade`).



## REFERENCIAS

- González, R. y Woods, R. (2002). *Digital Image Processing*. New Jersey: Pearson Prentice Hall.
- Kittler, J. (2002). *Reconocimiento de patrones*. Obtenido de [http://iie.fing.edu.uy/ense/assign/recpat/material/sistemas\\_rec\\_patrones.pdf](http://iie.fing.edu.uy/ense/assign/recpat/material/sistemas_rec_patrones.pdf)
- Morelli, A. y Padonavi S. (2011). *Detección y reconocimiento de caras*. Tesis de Licenciatura en Ciencias de la Computación. Universidad de Buenos Aires, Buenos Aires, Argentina.
- OpenCV. (2014). *Open Source Computer Vision*. Obtenido de <http://opencv.org/>
- Overton, G. (2011). *Photonics applied: Defense: IR countermeasures aim for safer flights*. Obtenido de <http://www.laserfocusworld.com/articles/print/volume-47/issue-8/features/photonics-applied-defense-ir-countermeasures-aim-for-safer-flights.html>
- Parrot. (2014). *AR. Drone 2.0*. Obtenido de <http://ardrone2.parrot.com/>
- Pietikäinen, M. y Zhao, G. (2009). *Local Texture Descriptors in Computer Vision*. Obtenido de [http://www.ee.oulu.fi/mvg/files/File/ICCV2009\\_tutorial\\_Matti\\_guoying-Local%20Texture%20Descriptors%20in%20Computer%20Vision.pdf](http://www.ee.oulu.fi/mvg/files/File/ICCV2009_tutorial_Matti_guoying-Local%20Texture%20Descriptors%20in%20Computer%20Vision.pdf)
- Robotnik. (2014). *Summit XL*. Obtenido de <http://www.robotnik.es/robots-moviles/robot-movil-summit-xl/>
- Sinodefenceforum. (2011). *J-20, The new generation fighter III*. Obtenido de <http://www.sinodefenceforum.com/air-force/j-20-new-generation-fighter-iii-51-5600.html>
- Viola, P. y Jones, M. (2004). *Robust Real-Time Face Detection*. Obtenido de <http://www.vision.caltech.edu/html-files/EE148-2005-pring/pprs/viola04jcv.pdf>

