

# Public And Private Service Vehicle Classification In Bogotá using SVM and AdaBoost

Francisco Calderon\*, Carlos Alberto Parra\*\*

## Abstract

This paper presents the implementation and comparison of algorithms for support vector machines "SVM" and AdaBoost in the classification of public and private vehicles using segmented images of video sequences taken in Bogotá city. Using as tools the OpenCV libraries implemented in C. The algorithms performances are remarkable and therefore its use could have a positive impact in the reduction of traffic problems.

**Keywords:** AdaBoost, Binary Trees, OpenCV, Pattern recognition, SVM, Traffic engineering, Vehicles classification.

## Resumen

Este artículo presenta el diseño e implementación y comparación de algoritmos para la clasificación de vehículos privados y públicos en Bogotá. El desempeño de estos algoritmos de clasificación es notable, y vale la pena anotar el impacto potencial que tendrían en la reducción de problemas de tráfico. Los datos experimentales fueron imágenes segmentadas de videos tomados sobre el tráfico en la ciudad de Bogotá. Por otro lado, los algoritmos que se utilizaron son máquinas de aprendizaje como Support Vector Machines "SVM" y Adaboost. Vale la pena notar, que se hizo uso de las librerías OpenCV implementadas en C.

**Palabras clave:** AdaBoost, árboles binarios, OpenCV, reconocimiento de patrones, SVM, Ingeniería de tráfico, clasificación de vehículos.

Fecha Recibido: 2012-09-25

Fecha Aprobado: 2012-11-15

\* Maestría ingeniería electrónica, Candidato a Phd. Grupo de Investigación en Sistemas Inteligentes, Robótica y Percepción. Facultad de Ingeniería Electrónica. Pontificia Universidad Javeriana 2013. Email: calderonf@javeriana.edu.co

\*\* PhD, Maestría Automatica e Informática Industrial, Maestría en Ingeniería eléctrica. Email: carlos.parra@javeriana.edu.co



## I. Introducción

En ingeniería de tráfico es de gran importancia contar con mediciones hechas sobre la vía, estas mediciones son realizadas de manera manual lo que implica errores humanos. El uso de técnicas de visión artificial y en este caso en especial de reconocimiento de patrones es usado para clasificar entre automóviles públicos y particulares teniendo en cuenta las restricciones normativas impuestas en la ciudad de Bogotá. En esta ciudad los automóviles de servicio público como taxis, camionetas empresariales, minivans de transporte, etc, cuentan con una placa de color de fondo blanco con letras negras y una reproducción de esta debe estar a los costados y en la parte superior del vehículo debe ser idéntica en cuanto a fuente y colores con la placa que entrega el distrito y debe tener un tamaño de 65 X 30 cm[4]. Los vehículos particulares cuentan con una placa de color de fondo amarillo con letras negras, y esta debe estar ubicada solamente en la parte central tanto delante como detrás del automóvil.

Para realizar la clasificación se tomaron los algoritmos de clasificación de máquinas de soporte vectorial y Boosting mediante el algoritmo de AdaBoost. Se emplean estas técnicas dado que el problema cuenta con solo dos clases “público y particular” y a la alta dimensionalidad en las características a trabajar, elementos que son favorables para las técnicas de clasificación seleccionadas.

El presente artículo está distribuido de la siguiente forma: primero se mostrarán las características usadas para llevar a cabo la clasificación, seguido de una breve introducción a los algoritmos usados con la implementación y configuración de cada uno, pasando luego a compararlos para finalizar con las conclusiones.

### 2. Características usadas

Las características son tomadas sobre una imagen segmentada en la que se encuentra solamente el automóvil, esta imagen es tomada de un vídeo de la vía con una ubicación de cámara elevada, de tal manera que se presenten las menores obstrucciones posibles, en la Figura 1 se muestra un ejemplo de las imágenes usadas, estas se segmentaron de

manera automática usando el algoritmo propuesto en[3]. Tomando estas imágenes como base para la extracción de características.



Figura 1. Ejemplos de Imágenes de automóviles

Las características a ser extraídas están relacionadas con la descripción del color de la imagen y de la forma en la que este color se distribuye en la imagen, no se hace una segmentación detallada del vehículo mas allá de la del rectángulo contenedor. Sobre la imagen en el espacio HSV se realiza el conteo de valores “histograma” sobre el plano de matiz “Hue” y sobre el plano de saturación que va de 0 a 255, el matiz va de 0° a 360°, sin embargo en la librería OpenCV, Se representa en 8 bits truncados de 0 a 180, se dividen los intervalos en  $si$  y  $sj$ , donde estos representan la cantidad de contenedores equiespaciados en el histograma bidimensional para matiz,  $si = 10$  y saturación  $sj = 10$  respectivamente. Y se crea una matriz de representación del histograma 2D sin normalizar  $h$  tal que:

$$h_{si,sj} = \begin{matrix} \# \text{elementos que cumplen los intervalos} \\ si \text{ y } sj \end{matrix}$$

Ésta matriz  $h_{si,sj}$  se normaliza para obtener la matriz  $H_{si,sj}$  de tal manera que la sumatoria de todos los elementos sea uno. Un ejemplo del resultado puede verse en la Figura 2, a partir de esta matriz se obtiene una representación del color en la imagen.

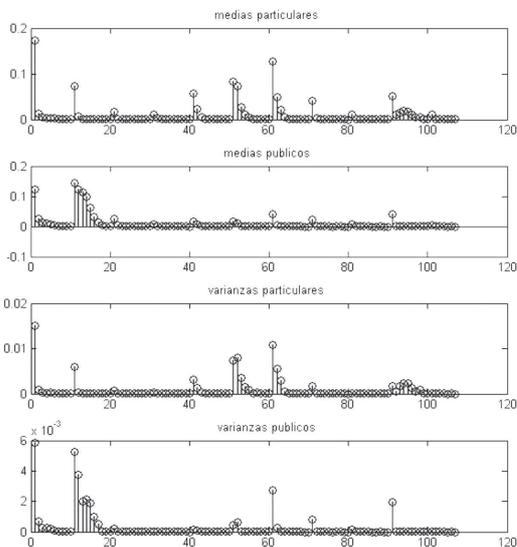
También se tienen como características sobre el espacio HSV la representación de forma, en la Figura 4 se aprecia como en el plano de saturación la forma del automóvil en el caso de los vehículos públicos presenta espacios de baja intensidad “negro” sobre la placa, cambiando de esta manera la forma del automóvil comparado con uno particular.

Sobre el plano de saturación se hallaron los 7 momentos de  $Hu$  [1], y de esta manera caracterizar la forma del automóvil.



**Figura 2.** Ejemplo de Histograma filas matriz, columnas saturación

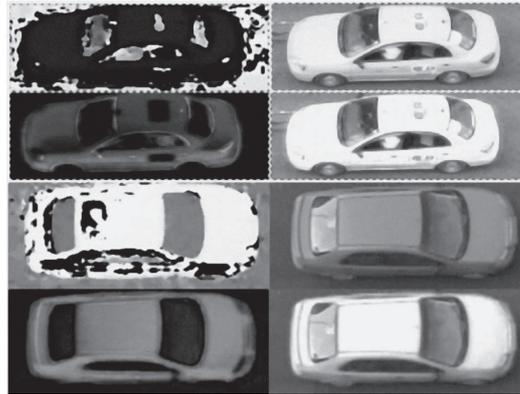
En base a las anteriores dos técnicas se encontraron las características a ser usadas sin embargo, en particular para representación en color, toda la matriz  $H_{st,sj}$  no aporta la misma información por lo que de esta se toman solo las coordenadas que más información aportan, estas se escogieron teniendo un conjunto de entrenamiento y hallando sobre este 100 valores representativos de color “matriz de 10X 10”, más 7 características correspondientes a los momentos de  $Hu$ , la media y la varianza por características se aprecia en la figura 3:



**Figura 3.** Media y varianza del conjunto de características, notando  $H$  de manera lineal por filas y concatenando con  $Hu$

Con base a los valores de media y varianza, se escogen 3 conjuntos de características, de las muestras  $A = [H_{(s,...,6)}, Hu]$ , donde  $Hu$  es el vector de los 7 momentos de  $Hu$ , y  $H^{(x,y)}$  representa una

notación lineal por filas sobre la matriz, en este caso de una matriz de 10 X 10 de 100 muestras, de la muestra  $x$  a la muestra  $y$ .  $B = [H_{(10,...,16)}, H_{(50,...,53)}, H_{(60,...,63)}, H_{(90,...,96)}, Hu]$  sobre la matriz  $H$  y  $Hu$ , el primer momento de  $Hu$ , y por último  $C = [H_{10-16}, H_{50-53}, H_{60-63}, H_{90-96}, Hu]$ .



**Figura 4.** Ejemplo de transformación a espacio HSV, izq - der, arriba - abajo para los dos vehículos: Matiz, Original, Saturación, Tinte “Value”

### 3. Máquinas de soporte vectorial “svm”

Las máquinas de soporte vectorial [1], [2], [6], [5] o SVM “por sus siglas en Inglés”, son un conjunto de métodos de aprendizaje supervisado para la clasificación y regresión, tomando los datos de entrada como conjuntos de vectores en un espacio  $n$ -dimensional. La SVM construye un hiperplano de separación en el espacio, de los datos de entrada que maximiza el margen entre los conjuntos de datos. En los casos que no son linealmente separables SVM emplea un kernel con el cual opera los datos de entrada que permite subir la dimensionalidad del problema a una en la que sea posible separar las muestras mediante un hiperplano.

Se emplean varios de los kernel y se elije el kernel de funciones de base radial “RBF” dado que presenta el menor tiempo de ejecución en comparación a los otros kernels, además de presentar buenos resultados en las primeras pruebas de clasificación. Este kernel está descrito por la ecuación 1.

$$K(X, X') = e^{(-\gamma \|X - X'\|^2)} \text{ para } \gamma > 0 \quad (1)$$

donde  $\gamma$  es el parámetro conocido como Gamma que describe el kernel.

En la implementación de SVM, Existe un parámetro de costo  $C$  que asigna un peso o penalidad a cada muestra mal clasificada mientras más grande es el valor, más grande es la penalidad que se le impone a cada muestra mal clasificada, adicionalmente al ser SVM un problema de optimización cuadrática, se deben fijar los parámetros de la iteración, que para este caso se toma un máximo de 1000 iteraciones ó que el error llegue a un punto en el que las iteraciones en punto flotante de 32 bits no sigan disminuyendo o sea  $1,0F + \epsilon = 1,0F$  en este caso:  $1.192092896e-07$  que es un define ubicado en la librería estándar de C/C++ "float.h".

Para la elección de estos parámetros, se realiza primero tomando los valores por defecto en la librería de OpenCV, y sobre estos se realiza un barrido de parámetros, hasta obtener el menor error sobre el conjunto de entrenamiento. Se toman valores desde  $C = 0,01$  a  $C = 2000$  y desde  $\gamma = 0,01$

a  $\gamma = 800$ , el algoritmo incluye la separación aleatoria en dos conjuntos de los datos originales, el primero usado para el entrenamiento del clasificador y el segundo que prueba el clasificador obtenido; se hace esto por cada valor de  $C$  y  $\gamma$  con 10 saltos, cada uno de estos saltos es dado multiplicando el valor anterior por una constante  $l$  tal que

$$l = \frac{V_f^{\frac{1}{10}}}{V_i^{\frac{1}{10}}}$$

los que garantiza 10 saltos exponenciales sobre el intervalo  $[V_i, V_f]$ , usando esta ecuación, se seleccionaron los intervalos para  $C$  y  $\gamma$ . Por último sobre estos valores se tomó los  $C$  y  $\gamma$  que menos errores presentaran en la clasificación sobre el conjunto de entrenamiento.

#### 4. Boosting - adaboost

El objetivo de Boosting[1], [5], [7] es mejorar la precisión de un determinado algoritmo de aprendizaje. La clasificación total es realizada con la ayuda de los pesos combinados de un grupo de clasificadores.

El entrenamiento se realiza con cada uno de estos clasificadores a la vez, ajustando el conjunto de entrenamiento y su voto para la suma total dependiendo de su precisión sobre la clasificación realizada en el conjunto de entrenamiento.

Existen variaciones del Boosting básico, el más popular es el conocido como AdaBoost o "Adaptative Boosting". Este permite al diseñador continuar añadiendo clasificadores débiles, hasta que se haya llegado a un bajo nivel de error sobre las muestras de entrenamiento. Cada muestra de entrenamiento recibe un peso que determina su probabilidad de ser seleccionada en la siguiente iteración para un conjunto de entrenamiento de cada clasificador componente, si una muestra de entrenamiento es correctamente clasificada, entonces se reduce la posibilidad de ser usada de nuevo en el siguiente entrenamiento. AdaBoost se centra en los patrones mal clasificados por los clasificadores componentes anteriores y los toma como muestras de los clasificadores componentes futuros.

Para la implementación de AdaBoost, se usa como clasificador componente árboles de decisión binarios, de estos, se escoge el nivel de profundidad  $L$  y se toma como de igual peso el error en la clasificación, es decir, es igual un error de particular a público que al contrario, sobre los parámetros de AdaBoost, se puede escoger el número de clasificadores componentes  $k$  y un umbral realizado sobre la distribución de pesos, de tal manera que solo toma para entrenar el siguiente clasificador las muestras cuyos peso en la distribución contribuyan en más de este umbral  $Wk$ .

Para escoger estos parámetros se toma primero un valor pequeño en profundidad de los árboles de decisión binarios de 1 hasta 10, y se varia el número de clasificadores componente en 50, 100 y 200.

#### 5. Resultados

Sobre los 3 conjuntos de características, se prueban los clasificadores para distintos valores de configuración de  $\gamma$  y  $C$  en las SVM.  $Y_k$  y  $W_k$  para AdaBoost con distintos valores de profundidad de los árboles de decisión  $L$ .

En las tablas presentadas a continuación,  $E_{puT}$  y  $E_{paT}$  es el error total sobre el conjunto de entrenamiento,  $E_{puP}$  y  $E_{paP}$  es el error sobre el conjunto de prueba.

Se cuenta con un total de 1521 muestras las cuales se dividen en 70 % para entrenamiento y 30 % para pruebas, en la Tabla 1, se tiene el número

de muestras correspondiente a la división mencionada.

**Cuadro 1.** Muestras

	Entrenamiento	Pruebas
particulares	460	196
públicos	606	259

### A. Resultados de máquinas de soporte vectorial “SVM”

En las tablas 2, 3 y 4. Se presenta el resumen de los resultados obtenidos aplicando el algoritmo de SVM.

**Cuadro 2.** Mejores resultados svm muestras A

1 Automático		2		3	
$\gamma$	0.338	$\gamma$	120	$\gamma$	300
$C$	10	$C$	10	$C$	30
VS	105	VS	594	VS	864
$E_{puT}$	25 - 2.890 %	$E_{puT}$	7 - 0.809 %	$E_{puT}$	18 - 2.080 %
$E_{paT}$	9 - 1.371 %	$E_{paT}$	2 - 0.304 %	$E_{paT}$	2 - 0.304 %
$E_{puP}$	5 - 0.825 %	$E_{puP}$	7 - 1.155 %	$E_{puP}$	18 - 2.970 %
$E_{paP}$	4 - 0.869 %	$E_{paP}$	2 - 1.521 %	$E_{paP}$	2 - 0.434 %

**Cuadro 3.** Mejores resultados svm muestras B

1 Automático		2		3	
$\gamma$	5.767	$\gamma$	120	$\gamma$	300
$C$	10	$C$	10	$C$	30
VS	89	VS	388	VS	643
$E_{puT}$	25 - 2.890 %	$E_{puT}$	9 - 1.040 %	$E_{puT}$	11 - 1.271 %
$E_{paT}$	8 - 1.219 %	$E_{paT}$	7 - 1.067 %	$E_{paT}$	8 - 1.219 %
$E_{puP}$	5 - 0.825 %	$E_{puP}$	8 - 1.320 %	$E_{puP}$	11 - 1.815 %
$E_{paP}$	4 - 0.869 %	$E_{paP}$	7 - 1.521 %	$E_{paP}$	8 - 1.739 %

**Cuadro 4.** Mejores resultados svm muestras C

1 Automático		2		3	
$\gamma$	5.767	$\gamma$	120	$\gamma$	300
$C$	10	$C$	10	$C$	30
VS	93	VS	388	VS	643
$E_{puT}$	25 - 2.890 %	$E_{puT}$	9 - 1.040 %	$E_{puT}$	11 - 1.271 %
$E_{paT}$	8 - 1.219 %	$E_{paT}$	7 - 1.067 %	$E_{paT}$	8 - 1.219 %
$E_{puP}$	5 - 0.825 %	$E_{puP}$	8 - 1.320 %	$E_{puP}$	11 - 1.815 %
$E_{paP}$	3 - 0.652 %	$E_{paP}$	7 - 1.521 %	$E_{paP}$	8 - 1.739 %

El algoritmo de SVM presentó desde un comienzo errores en la clasificación menores a el 3 %, sobre el conjunto de pruebas, y en algunos casos 0 % sobre el conjunto de entrenamiento, sin embargo las muestras que presentaron error en el caso de ser públicos y que fueron catalogadas como particulares fueron los vehículos que no son taxis. La penalización de error del algoritmo hizo que al haber pocos de estos casos no fueran tenidos en cuenta en la clase público.

Por otro lado los errores en los vehículos particulares son bajos, seguramente por el hecho de que la clase público es una clase con una varianza pequeña entre sus muestras como se aprecia en la Figura 3, los mejores resultados sobre el conjunto de pruebas se alcanzaron en todos los casos con la selección de parámetros automática en estos es más notorio que sobre el conjunto de entrenamiento se presentaron mayores errores algunos de estos se muestran en la Figura 5.

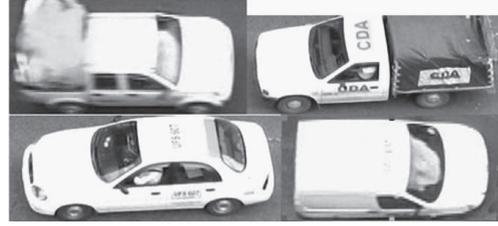


Figura 5. Ejemplo de públicos mal clasificados por SVM

Algunos particulares que presentaron error, se encuentran en la Figura 6, se presentaron más casos de error de clasificación, al eliminar las características de más en la matriz  $H_{si,sj}$



Figura 6. Ejemplo de Particulares mal clasificados por SVM

### B. Resultados AdaBoost

En las tablas 5, 6 y 7. Se presentan los mejores resultados de los clasificadores aplicando la técnica de Adaboost.

Cuadro 5. mejores resultados adaboost muestras A

1		2		3	
$k$	100	$k$	200	$k$	100
$W_k$	0.95	$W_k$	0.95	$W_k$	0.95
$L$	5	$L$	1	$L$	10
$E_{puT}$	7 - 0.809%	$E_{puT}$	6 - 0.693%	$E_{puT}$	5 - 0.578%
$E_{paT}$	7 - 1.067%	$E_{paT}$	11 - 1.676%	$E_{paT}$	10 - 1.524%
$E_{puP}$	5 - 0.825%	$E_{puP}$	6 - 0.990%	$E_{puP}$	5 - 0.825%
$E_{paP}$	14 - 3.043%	$E_{paP}$	11 - 2.391%	$E_{paP}$	10 - 2.173%

Cuadro 6. Mejores resultados adaboost muestras B

1		2		3	
$k$	100	$k$	200	$k$	100
$W_k$	0.95	$W_k$	0.95	$W_k$	0.95
$L$	5	$L$	1	$L$	10
$E_{puT}$	5 - 0.578 %	$E_{puT}$	7 - 0.809 %	$E_{puT}$	5 - 0.578 %
$E_{paT}$	6 - 0.914 %	$E_{paT}$	6 - 0.914 %	$E_{paT}$	3 - 0.457 %
$E_{puP}$	5 - 0.825 %	$E_{puP}$	7 - 1.155 %	$E_{puP}$	5 - 0.825 %
$E_{paP}$	6 - 1.304 %	$E_{paP}$	6 - 1.304 %	$E_{paP}$	3 - 0.652 %

**Cuadro 7.** Mejores resultados adaboost muestras C

	1		2		3
$k$	100	$k$	200	$k$	100
$W_k$	0.95	$W_k$	0.95	$W_k$	0.95
$L$	5	$L$	1	$L$	10
$E_{puT}$	5 - 0.578 %	$E_{puT}$	6 - 0.693 %	$E_{puT}$	4 - 0.462 %
$E_{paT}$	5 - 0.762 %	$E_{paT}$	10 - 0.524 %	$E_{paT}$	3 - 0.457 %
$E_{puP}$	5 - 0.825 %	$E_{puP}$	6 - 0.990 %	$E_{puP}$	4 - 0.660 %
$E_{paP}$	5 - 1.086 %	$E_{paP}$	10 - 2.173 %	$E_{paP}$	3 - 0.652 %

Para la técnica de Adaboost se presentan errores menores a los de SVM tanto en el conjunto de entrenamiento como en el conjunto de pruebas, los mejores resultados se presentaron con un L alto,  $L = [5, 10]$ , con un  $L = 1$ , se requirió de más clasificadores para obtener los mismos resultados. Al contrario de SVM, se presentaron los mejores resultados al eliminar las características de más en la matriz  $H_{si,j}$  y usar todos los momentos de Hu en la clasificación, esto al comparar la Tabla 5 con la Tabla 7.

Se pueden apreciar los errores del algoritmo usando el conjunto de características C y los parámetros presentados en la Tabla VII-3 en la Figura 7 y la Figura 8.



**Figura 7.** Errores de clasificación de públicos por AdaBoost



**Figura 8.** Errores de clasificación de particulares por AdaBoost

Este mismo algoritmo fue capaz de clasificar más minivans y vehículos públicos de color blanco que SVM algunas de estas se muestran en la Figura 9.



**Figura 9.** Aciertos en clasificación de públicos por AdaBoost

### Conclusiones

Como se presentó en los resultados AdaBoost presenta mejores resultados en la clasificación que SVM, tanto sobre el conjunto de entrenamiento como sobre el conjunto de prueba; los errores que se presentan en Adaboost son casos poco usuales, como son el caso de vehículos particulares en donde menos del 1 % de las muestras recopiladas de vehículos cuentan con una tonalidad amarilla y sobre estos el clasificador AdaBoost con las características de forma usando momentos de Hu presentó menos errores, sin embargo para vehículos públicos las minivans y camionetas de color blanco presentaron mayor error en la clasificación, todos los errores menos uno pertenecen a este grupo.

En comparación en el caso de SVM presentó más errores en la clasificación de vehículos públicos, al hacer un análisis de la varianza y la media sobre la clase público, se encontró que solo los taxis poseen una varianza aún menor que la mostrada por la Figura 3 y esta es aumentada por las minivans y vehículos de color blanco que

entre ellos poseen una varianza algo menor a la varianza de la clase particular, esto y el hecho de que se recopilaban pocas muestras de minivans en comparación con taxis, pudo ocasionar que las minivans fueran consideradas por SVM como posibles errores sobre el conjunto de prueba.

A pesar que el algoritmo de AdaBoost presentó el menor error entre las técnicas, el número de clasificadores componente y la profundidad de los árboles de decisión fue grande para lograr segmentar de manera correcta el problema.

Para mejorar la clasificación, puede incluirse características que se basen en la detección de la placa en los automóviles de servicio público. Para esto se requiere de que la ubicación de la cámara permita imágenes de mayor resolución y una mayor complejidad del algoritmo de extracción de características.

## Referencias

- [1] G. Bradski and A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library. O'Reilly Media, Inc., 2008.
- [2] C. J. Burges, "A tutorial on support vector machines for pattern recognition," *ell Laboratories*, Lucent Technologies, 1998.
- [3] Calderon.F and Urrego.G, *Conteo Automático De Vehículos*. PUJ, Nov. 2008.
- [4] C. Cordoba, "Sobre la reproducción de la placa nacional," *Transito Y Transporte* resolucion 538, 2001.
- [5] R. Duda, P. Hart, and D. Stork, *Pattern classification*. Wiley New York, 2001.
- [6] C.-W. Hsu, C.-C. Chang, and C.-J. Lin, "A practical guide to support vector classification," *Department of Computer Science*, Oct. 2008.
- [7] Yoav.F and Schapire. R.E, "A short introduction to boosting," *Japanese Society for Artificial Intelligence*, Sep. 1999.