

Detección de placas vehiculares mediante modelo de clasificador en cascada basado en lenguaje Python

Detection of license plates by means of a cascade classifier model based on Python language

Carlos Vicente Niño-Rondón^a, Diego Andrés Castellano-Carvajal^b, Sergio Alexander Castro-Casadiegos^c, Byron Medina-Delgado^d, Dinael Guevara-Ibarra^e

^aIngeniero Electrónico, carlosvicentenr@ufps.edu.co, <https://orcid.org/0000-0002-3781-4564>, Servicio nacional de aprendizaje, Cúcuta, Colombia

^bEstudiante de Ingeniería Electrónica, diegoandrescc@ufps.edu, <https://orcid.org/0000-0002-4530-1136>, Universidad Francisco de Paula Santander, Cúcuta, Colombia

^cMagister en Ingeniería Electrónica, sergio.castroc@ufps.edu.co, <https://orcid.org/0000-0003-0962-9916>, Universidad Francisco de Paula Santander, Cúcuta, Colombia

^dDoctor en Ciencias, byronmedina@ufps.edu.co, <https://orcid.org/0000-0003-0754-8629>, Universidad Francisco de Paula Santander, Cúcuta, Colombia

^eDoctor en Ingeniería Electrónica, dinaelgi@ufps.edu.co, <https://orcid.org/0000-0003-3007-8354>, Universidad Francisco de Paula Santander, Cúcuta, Colombia

Forma de citar: Niño-Rondón, C. V., Castellano-Carvajal, D. A., Castro-Casadiegos, S. A., Medina-Delgado, B., Guevara-Ibarra, D. (2021), Detección de placas vehiculares mediante modelo de clasificador en cascada basado en lenguaje Python. *Eco Matemático*, 12 (1), 41-53

Recibido: 2 de septiembre de 2020

Aceptado: 10 de noviembre de 2020

Palabras clave

Detector en cascada,
placas,
transformación
geométrica,
tratamiento de imágenes,
vehículos.

Resumen: La detección de placas vehiculares empleando técnicas de aprendizaje automático mejora los procesos de rastreo, seguimiento y seguridad. Se presenta el desarrollo de un modelo de clasificador en cascada para la detección de placas vehiculares, utilizando las herramientas de Python, OpenCV y Cascade Trainer GUI, basadas en código abierto. Las imágenes utilizadas para el procesamiento fueron capturadas mediante una cámara para Raspberry Pi conectada a la placa embebida, en diversos puntos de la zona céntrica de la ciudad fronteriza de Cúcuta, Colombia; posteriormente enviadas a una computadora personal y redireccionadas mediante transformaciones geométricas; y para garantizar el alto rendimiento del sistema de clasificación, se aplican procesos de aumentado de datos, pasando de 245 a 1867 imágenes para el entrenamiento del detector en cascada. El modelo de clasificación tardó 17.4 minutos en crearse, y se probó con imágenes y videos en ambientes reales de la ciudad de Cúcuta, logrando la detección de placas vehiculares de matrículas colombianas y venezolanas con una efectividad del 90.26%.

*Autor para correspondencia: sergio.castroc@ufps.edu.co

DOI 10.22463/17948231.3068

2462-8794© 2021 Universidad Francisco de Paula Santander. Este es un artículo bajo la licencia CC BY 4.0

Keywords

Cascade detector,
plates,
geometric
transformation,
image processing,
vehicles.

Abstract: The detection of vehicle plates using machine learning techniques improves the processes of tracking, tracing and security. The development of a cascade classifier model for license plate detection is presented, using Python, OpenCV and Cascade Trainer GUI tools, based on open source. The images used for the processing were captured by a Raspberry Pi camera connected to the embedded plate, in several points of the central area of the frontier city of Cúcuta, Colombia, then sent to a personal computer and redirected through geometric transformations; and to guarantee the high performance of the classification system, data enhancement processes are applied, going from 245 to 1867 images for the training of the cascade detector. The classification model took 17.4 minutes to create, and was tested with images and videos in real environments in the city of Cúcuta, achieving the detection of Colombian and Venezuelan license plates with an effectiveness of 90.26 %.

Introducción

La visión artificial hace referencia al conjunto de técnicas por las que es posible procesar, analizar y obtener información de imágenes y videos del mundo real [1]. Las técnicas de aprendizaje permiten la distinción de patrones de forma automática facilitando los procesos de clasificación y toma de decisiones [2] [3].

Entre las técnicas de aprendizaje para la detección y clasificación de objetos destaca la técnica de clasificadores en cascada [4] [5], donde mediante estructuras de múltiples niveles se acoplan diversos clasificadores y se optimiza el proceso de detección [6]. En este proceso, un conjunto de imágenes correspondiente a imágenes positivas y negativas se entrena de modo que se logre la distinción entre los objetos en cuestión para la clasificación [7] [8] [9], atenuando los efectos de falsos positivos y falsos negativos en los procesos de detección [10] [11].

Las matrículas o placas son la documentación oficial de los vehículos y se han convertido en un pilar para el desarrollo de la motorización, con efectos comerciales y de seguridad [12]. En herramientas basadas en código abierto como Python y OpenCV [13] [14] [15], se han desarrollado estructuras de clasificadores en cascada de tipo Haar, que entre otros, permiten principalmente la detección de

cuerpos, rostros y características faciales específicas [16] [17], pero se limitan a dichos modelos previamente entrenados.

En este documento se presenta el desarrollo de un modelo de clasificador en cascada tipo Haar basado en código abierto, para la detección de placas vehiculares en la ciudad de San José de Cúcuta, Colombia, con imágenes capturadas en su zona céntrica mediante una cámara y una placa embebida Raspberry Pi, enviadas a una computadora personal para la creación del modelo de clasificación en la aplicación Cascade Trainer GUI disponible para sistema operativo Windows, mediante lenguaje de programación Python y la librería OpenCV. El modelo se prueba en un ambiente real en la zona céntrica de la ciudad de Cúcuta y se mide su rendimiento según la tasa de falsos positivos y falsos negativos en las detecciones. Además, el rendimiento de las herramientas de hardware se obtiene midiendo el tiempo de respuesta y el espacio de memoria requerido.

Estado del arte

Aldás et al. [12], desarrollaron un sistema de procesamiento de imágenes para la detección de placas vehiculares. La programación se realiza en LabVIEW donde se corrigen las imágenes, además

de etapas de reconocimiento óptico de caracteres con procesos de binarización. Asimismo, Rodríguez et al. [18] implementaron un sistema de detección y extracción de placas de vehículos en imágenes de video, utilizando sistemas de iluminación para la cámara de video, digitalizador de imagen y computadora para el procesamiento. Utilizaron la librería OpenCV definiendo márgenes de distancia y altura para la ubicación de la cámara, cuyas imágenes se procesan mediante binarización y transformaciones morfológicas y por segmentación de regiones se determina la región de imagen referente a la placa. De igual forma, Tasiguano propuso un método para el desarrollo de algoritmos enfocados al reconocimiento de placas [19], donde destaca la localización de la placa dentro de la imagen teniendo en cuenta factores como la posición y orientación del objeto, y detección de partes faltantes en las placas. La programación se desarrolló en LabVIEW y para la adquisición de la imagen se utilizó una cámara IP de alta resolución. Por otra parte, Padilla [20], estudió la detección de objetos usando algoritmos de detectores en cascada y la librería OpenCV, además de métodos basados en procesamiento de imagen con segmentación, umbralización y transformaciones morfológicas y geométricas para la distinción de objetos en movimiento por el cuadro principal de la imagen. En esa misma línea, Cepeda aplicó fundamentos de visión artificial aplicado a la detección de matrículas de vehículos [21], utilizando el algoritmo de Adaboost, convirtiendo a escala de color HSV y binarizando la imagen posteriormente. La detección de placas vehiculares se realiza mediante el sistema de clasificación cascada de placas rusas disponible en el paquete de OpenCV, con una efectividad aproximada en las detecciones de 93 %. Igualmente, Navacerrada desarrolló un sistema para la detección de placas utilizando las herramientas de OpenCV [22]. Validó los algoritmos SURF, Haar Cascade y OCR, acompañado de etapas de binarización, segmentación, adelgazamiento de componentes y comparación con patrones. Hernández et al. [23] segmentaron imágenes de placas vehiculares

utilizando Haar-AdaBoost y clustering, así como la máquina de vectores de soporte SVM. Los autores hacen énfasis en la corrección de inclinación de la placa, y obtuvieron aciertos en las detecciones fluctuantes entre 85.71 % y 95.97 %. Asimismo, Menon y Omman [24] desarrollaron un sistema para la detección de placas en España basado en tratamiento y procesamiento de imagen, con etapas de detección de bordes y umbralización y operaciones morfológicas, búsqueda de contornos y mejorado mediante la máquina de vectores de soporte. De igual forma, Ramazankhani y Yazdian desarrollaron un sistema de clasificador en cascada para detección de placas de Irán [25]. Se enfocaron en un entrenamiento con recolección amplia de información de imágenes negativas para mejorar de esta forma el entrenamiento del clasificador. El método es aplicable para imágenes en color y escala de grises. Además, Davis et al. [26] desarrollaron un método para la detección de placas basándose en la mayor cantidad de bordes verticales de la imagen mediante el algoritmo VEDA. El método presentó mejor rendimiento en comparación con el método de Sobel, usado con frecuencia para detección de bordes o contornos.

Metodología

Se desarrolla una metodología que se apoya en el modelo ADDIE [27], (análisis, diseño, desarrollo, implementación y evaluación) de un modelo de clasificador en cascada para la detección de placas vehiculares en la ciudad de Cúcuta, Colombia. En primer lugar, se seleccionan las herramientas de hardware y software para el procesamiento. La segunda etapa hace referencia a la adquisición de la imagen. En la tercera etapa, se aplican procesos de transformación geométrica de la imagen. En la cuarta etapa, se aplican técnicas de aumento de datos, y en la quinta etapa se obtiene el modelo de clasificación en cascada para la detección de placas. La metodología se ilustra en la Figura. 1.

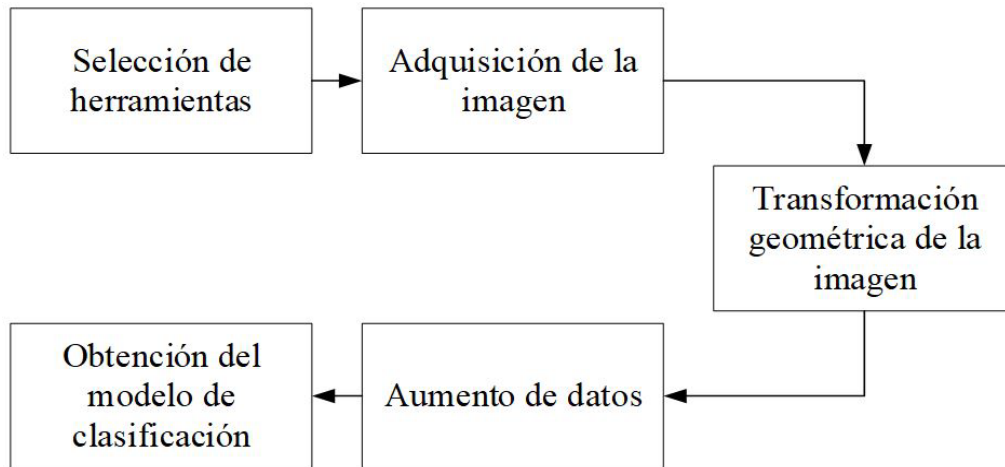


Figura 1. Metodología desarrollada. Fuente: Autores.

A. Selección de herramientas

Las herramientas de hardware y software se definen según dos etapas del procesamiento. En la primera de ellas se tienen en cuenta factores como adaptabilidad y portabilidad al momento de la captura y almacenamiento de las imágenes, por lo que se utilizan ordenadores de placa reducida. Para la segunda etapa, se debe tener en cuenta que para obtener el archivo .XML con el modelo de clasificación, se requiere de la aplicación Cascade Trainer GUI, disponible en exclusividad para versiones iguales y superiores a Windows 7.

B. Adquisición de la imagen

En los procesos de visión artificial, la adquisición de la imagen es una etapa fundamental para el procesamiento, ya que, de esta, depende la robustez con las que el sistema detectará y clasificará a los objetos presentes en el cuadro principal de la imagen [28]. En primer lugar, se selecciona el área en la cual se realizará la captura de la imagen, correspondiente a la zona céntrica de la ciudad de Cúcuta, Colombia, como se muestra en la figura 2.



Figura 2. Zonas de captura de la imagen. Fuente: Autores

Asimismo, se ubican la cámara y la placa embebida Raspberry Pi modelo 3B+ en puntos donde se logre la visualización de placas en las partes delantera y trasera de los vehículos [18]. En la ciudad de Cúcuta por su condición de frontera, se encuentran vehículos matriculados tanto en Colombia como en Venezuela, por lo que, en las imágenes obtenidas para el modelo de clasificación, se deben incluir fotogramas de placas de ambas nacionalidades, además de las de los vehículos de transporte público e intermunicipal, que circulan por la zona céntrica de la ciudad de Cúcuta. En la tabla I se muestran las zonas en las que se realizó la

adquisición de la imagen, la altura de localización del dispositivo y el tiempo de duración de los videos de los que se extraen las imágenes para el procesamiento.

Tabla I. Características de la adquisición de la imagen por zona

Zona	Altura de localización de la cámara (m)	Duración del video (s)
Cl 10, Av 2da y 3ra	6.2	210
Cl 9, Av 6ta y 7ma	5.7	217
Av 5ta, Cl 8 y 9	4	124
Cl 8, Av 5ta y 6ta	2.2	228
Cl 11, Av 4ta y 5ta	3.6	196

Fuente: Autores

Adicionalmente, en la Figura 3 se muestra la variación promedio de los niveles de luminosidad al momento de la captura de las imágenes de video en horas del día, entre las 8:00 AM y 3:00 PM, y horas de la noche entre las 5:50 PM y 8:00 PM respectivamente. El nivel de luxes promedio en las tomas diurnas fue de 7886.5 lx, mientras que para las tomas nocturnas fue de 306 lx. Los niveles de luminosidad se obtuvieron mediante un luxómetro digital.

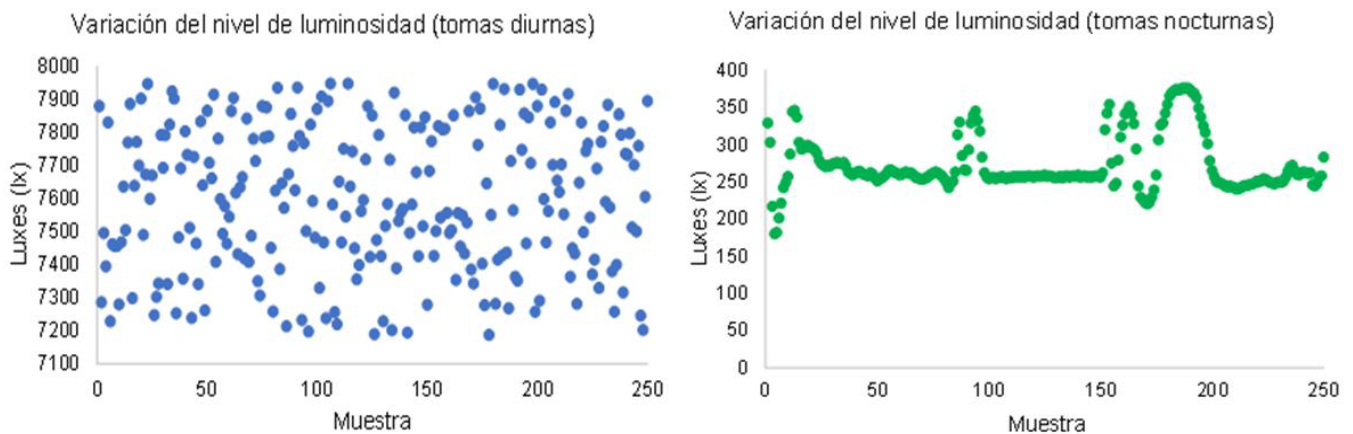


Figura 3. Variación del nivel de luminosidad al momento de la captura de las imágenes de video. Fuente: Autores

C. Transformación geométrica de la imagen

Con las transformaciones geométricas se realizan modificaciones en la distribución de los píxeles de la imagen respecto a un eje de coordenadas [29]. Usualmente, al producirse operaciones de giros, desplazamientos y aproximaciones, los nuevos píxeles de la imagen no coinciden las intersecciones iniciales y se ubican en puntos intermedios [30]. En la Fig. 4 se ilustra el efecto de una transformación

geométrica básica en la localización de los píxeles de la imagen.

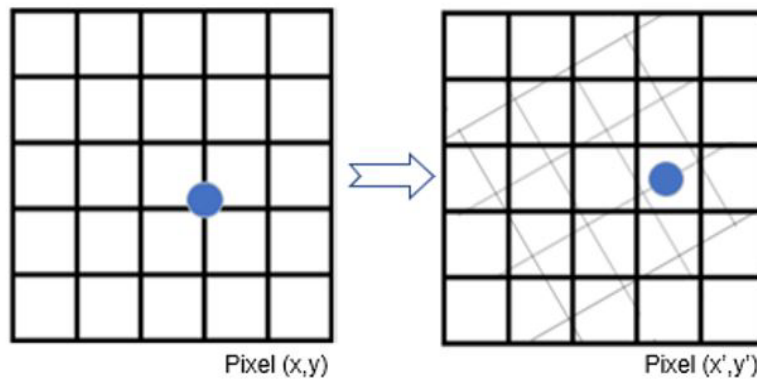


Figura 4. Modificación de la localización de píxeles al aplicar transformaciones geométricas. Fuente: Autores.

Por factores como superposición y peatones que circulan por las vías e impiden la distinción de la placa de los vehículos, ciertas imágenes se encuentran con variación respecto a los ejes horizontal y vertical, por lo que se aplica un proceso de redireccionamiento a las imágenes que presenten dicha condición. El redireccionamiento se aplica mediante eventos del mouse [31], donde manualmente se definen los cuatro vértices del rectángulo correspondiente a la placa del vehículo, y se logra la visualización frontal del área en cuestión.

D. Aumento de datos

El aumento de datos hace referencia a los procesos computacionales con los que se realizan variaciones leves a las imágenes capturadas en la adquisición [32] [33] [34]. Para la creación de un modelo de detectores en cascada tipo Haar, se requieren dos grupos de imágenes representativas referentes a las imágenes positivas, que contienen diversas tomas de placas vehiculares, e imágenes negativas, referentes a todos aquellos objetos que se encuentren en las zonas de espacio público y que no son categorizados como placas [35] [36].

Entre las modificaciones a las imágenes se realizan giros de 180° tanto horizontal como verticalmente. Así mismo, se realizan rotaciones a la imagen en un rango de 0° a 90° , escalas de zoom, desplazamiento horizontal, vertical y de corte de

0.15, 0.2, 0.2 y 0.15 respectivamente. El aumento de datos se realiza en Python mediante el paquete de Keras utilizando procesos de aprendizaje profundo [37].

E. Obtención del modelo de clasificación

Las imágenes positivas y negativas obtenidas en el aumento de datos se alojan en dos carpetas y se cargan a la aplicación Cascade Trainer GUI [11], utilizada para la creación del modelo de clasificador en cascada Haar que está disponible para sistema operativo Windows 7 o superiores. Las etapas requeridas para la obtención del modelo de clasificación están basadas en aprendizaje automático [38].

El clasificador consta de 20 etapas en cascada, 24×24 píxeles de tamaño en las imágenes de entrada, tasa de exactitud en el entrenamiento y validación del modelo de 0.995, y tasa máxima de error de 0.4. Los parámetros referidos se configuran en las pestañas de la aplicación Cascade Trainer GUI, que genera un archivo de extensión XML con el modelo de clasificación obtenido.

El modelo propuesto se presenta como una alternativa al modelo de clasificador en cascada para la detección de placas vehiculares rusas que se encuentra disponible entre los paquetes de descarga de OpenCV, ya que, dicho modelo se limita a la

detección de placas que presenten dimensiones de alrededor de 520x112 milímetros, siendo este el tamaño de las placas en las divisiones administrativas de Rusia, mientras que, en países como Colombia y Venezuela, las dimensiones de las placas vehiculares se encuentran alrededor de 330x150 milímetros, como lo indica la normatividad para estos países.

Resultados y discusión

Respecto a las herramientas utilizadas para el procesamiento, en la Figura 5 se presenta la arquitectura del sistema. Para la adquisición de las imágenes y videos se emplea el ordenador de placa

reducida Raspberry Pi 3B+, conectado a una cámara de 5 MPx de la familia Raspberry por estándar CSI. La placa Raspberry Pi funciona bajo el sistema operativo Raspbian y la codificación se realiza en Python, soportado por dicho sistema operativo. Las imágenes y videos se transfieren mediante una unidad USB extraíble a una computadora personal con procesador Core i7, 4GB de memoria RAM instalada y sistema operativo Windows 10, y mediante la aplicación Cascade Trainer GUI se obtiene el modelo de clasificación. Asimismo, el archivo .XML con el modelo de clasificación se transfiere mediante la unidad USB extraíble a la placa Raspberry Pi 3B+ donde se ejecuta y se valida.

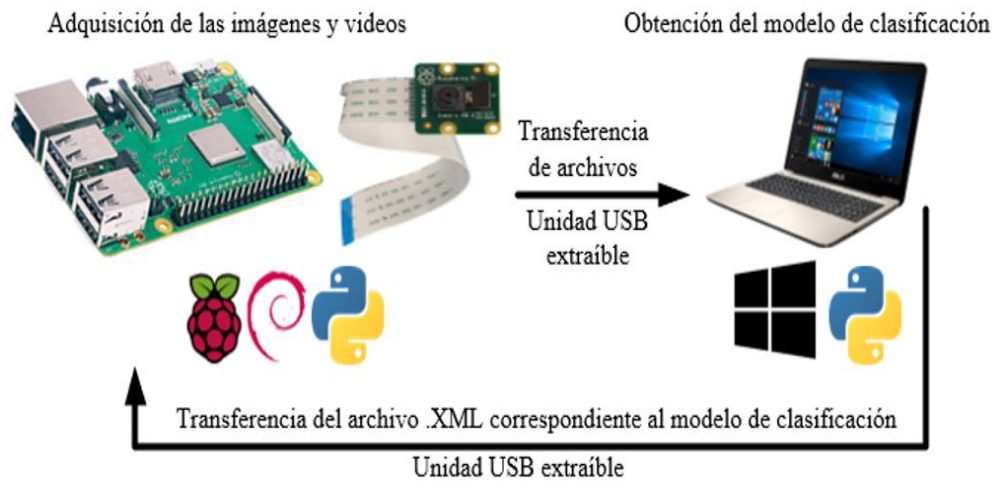


Figura 5. Arquitectura del sistema de procesamiento. Fuente: Autores

Así mismo, en la Figura 6, de arriba a abajo se muestran las imágenes obtenidas en la zona céntrica de la ciudad de Cúcuta. Seguido, se presenta la región de interés para el procesamiento referente al área en donde se localizan las placas de los vehículos. Además, se muestran los vértices ubicados mediante eventos del mouse, para el redireccionamiento a las placas mediante la transformación geométrica.



Figura 6. Adquisición y modificaciones iniciales a la imagen. Fuente: Autores.

En la tabla II se muestra la relación respecto al número de imágenes capturadas y modificadas inicialmente, y las obtenidas al aplicar el proceso de aumento de datos. Las 125 imágenes positivas resultan en 895 imágenes, mientras que de las 120 imágenes negativas resultan 972 imágenes. El aumento de las imágenes positivas y negativas se desarrolló con una exactitud de 87.63 % y 88.42 % respectivamente.

Tabla II. Relación respecto al número de imágenes

Imágenes	Cantidad inicial	Cantidad final	Exactitud
Positivas	125	895	0.8763
Negativas	120	972	0.8842
Total	245	1867	

Fuente: Autores

Respecto al modelo de clasificación, en la tabla III se presenta el índice de ruptura de aceptación, referente a la intensidad con la que se debe entrenar el modelo para garantizar robustez en la clasificación, y el tiempo de entrenamiento empleado en cada una de las etapas del clasificador en cascada. El entrenamiento del modelo demoró 1044 segundos y tras las 20 etapas, el índice de ruptura fue de 0.0001. El tiempo de entrenamiento es dependiente de la herramienta de hardware utilizada para el procesamiento, que para el presente caso corresponde

a una computadora personal con procesador Core-i7 y 4 GB de memoria RAM instalada.

Tabla III. Características del entrenamiento del modelo de clasificador en cascada

Etapa	Índice de ruptura de aceptación	Tiempo de entrenamiento (s)
1	1	60
2	0.2612	80
3	0.1061	56
4	0.0597	64
5	0.0311	72
6	0.0251	64
7	0.0145	56
8	0.0462	52
9	0.0305	52
10	0.0175	40
11	0.0125	36
12	0.0084	42
13	0.0035	56
14	0.0028	24
15	0.0015	28
16	0.0013	40
17	0.0012	48
18	0.0004	48
19	0.0003	64
20	0.0001	72
Total		1044

Fuente: Autores

Asimismo, en la figura 6 se muestra el funcionamiento del modelo de clasificación. En la primera y en la tercera fila se muestran las imágenes de entrada, y en la segunda y cuarta fila, se ilustra la detección de las placas vehiculares realizadas por el modelo.



Figura 6. Detección de las placas vehiculares. Fuente: Autores.

Además, en la tabla IV se presenta la exactitud del modelo de clasificación tipo Haar creado para la detección de placas vehiculares, según el error cuadrático medio al conocer la cifra de falsos positivos y falsos negativos durante la detección. De 127 placas, el modelo arroja 133 detecciones, existiendo 9 falsos positivos y 3 falsos negativos, obteniendo una exactitud de 90.26 %. De las 127 placas que circularon por el cuadro principal de video, 65 correspondieron a vehículos de servicio público y particulares colombianos, y 62 correspondieron a vehículos con matrículas venezolanas.

Tabla IV. Características del entrenamiento del modelo de clasificador en cascada

Parámetro	Valor
Número real de placas	127
Número de Detecciones	136
Falsos positivos	12
Falsos negativos	3
Exactitud del modelo	90.26 %

Fuente: Autores

Conclusiones

La captura de imágenes y videos en la zona céntrica de la ciudad de Cúcuta permitió tener a consideración del conjunto de imágenes los diversos tipos de vehículos que circulan por la ciudad. Asimismo, al capturar las imágenes desde alturas variadas se logra que, con el posterior entrenamiento del modelo de clasificación, se detecten placas vehiculares desde diversos puntos, y que el rendimiento del sistema de procesamiento no se limite a una determinada altura de localización de la cámara de video. Además, la modificación de perspectiva de las imágenes mediante transformaciones geométricas mejoró el conjunto de datos, ya que por factores como cercanía entre vehículos y peatones que se movilizan por las vías, en el 23.7 % de las imágenes obtenidas no se lograba la distinción de las placas de los vehículos.

En cuanto al aumento de datos para el entrenamiento del sistema de clasificación, las imágenes positivas se aumentaron aproximadamente 7 veces, mientras que las imágenes negativas se aumentaron aproximadamente 8 veces. Con esto, se garantiza una mejoría en el entrenamiento del modelo de clasificación, ya que, en los procesos de visión artificial basado en aprendizaje automático,

la robustez del sistema depende en gran medida de la calidad y cantidad de imágenes utilizadas para el entrenamiento.

El modelo creado presentó un error en las detecciones de 9.74 %, en su mayoría por falsos positivos, ya que se categorizaban como placas algunos sectores de los vehículos aledaños a las mismas. Además, se logró la detección tanto de placas colombianas y venezolanas, que son las que circulan con frecuencia por las vías de la ciudad de Cúcuta.

El modelo de clasificador cascada obtenido se presenta como alternativa a métodos de aprendizaje profundo como redes neuronales, que requieren de mayor tiempo de entrenamiento y recursos de herramienta de hardware. Igualmente, las herramientas basadas en código abierto y software libre como Python, OpenCV y Cascade Trainer GUI, permiten el desarrollo de aplicaciones de visión por computador con altas eficiencias computacionales y de aciertos en las detecciones.

Referencias

A. Menon and B. Omman, "Detection and Recognition of Multiple License Plate from

- Still Images,” in *2018 International Conference on Circuits and Systems in Digital Enterprise Technology*, ICCSDET 2018, 2018, pp. 1–5. DOI: 10.1109/ICCSDET.2018.8821138
- A. Mikołajczyk and M. Grochowski, “Data augmentation for improving deep learning in image classification problem,” in *2019 International Interdisciplinary PhD Workshop, IIPhDW 2019*, 2019, pp. 117–122. DOI: 10.1109/IIPhDW.2018.8388338
- A. M. Davis, C. Arunvinodh, and N. P. Arathy Menon, “Automatic license plate detection using vertical edge detection method,” in *ICIIECS 2015 - 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems*, 2015. DOI: 10.1109/ICIIECS.2015.7193073
- C. A. Castillo Benavides, L. F. García, N. D. Duque Méndez, and J. H. Estrada Estrada, “Mouse for People With Physical Disabilities in Virtual Educational Contexts,” *Ing. e Innovación*, vol. 3, no. 1, pp. 82–89, 2015. DOI: 10.21897/23460466.796
- C. H. Setjo, B. Achmad, and Faridah, “Thermal image human detection using Haar-cascade classifier,” in *Proceedings - 2017 7th International Annual Engineering Seminar*, InAES 2017, 2017. DOI: 10.1109/INAES.2017.8068554
- Cristian Andrés Tasiguano, “Desarrollo de algoritmos de reconocimiento de placas de vehículos,” *Quito Epn*, no. May, pp. 5–10, 2011
- C. M. Budoya, M. Kissaka, and J. Mtebe, “Instructional Design Enabled Agile Method Using ADDIE Model and Feature Driven Development Process,” *Int. J. Educ. Dev. Using Inf. Commun. Technol.*, vol. 15, no. 1, pp. 35–54, 2019
- C. Padilla, “Objects Detection using haarscascade algorithms and OpenCV,” Universidad de Camagüey Ignacio Agramonte y Loynaz, 2016
- C. Shorten and T. M. Khoshgoftaar, “A survey on Image Data Augmentation for Deep Learning,” *J. Big Data*, vol. 6, no. 1, pp. 1–48, 2019. DOI: 10.1186/s40537-019-0197-0
- C. V. Niño-Rondón, S. A. Castro-Casadiago, B. Medina-Delgado, D. Guevara-Ibarra, J. J. Ramírez-Mateus, and K. C. Puerto-López, “Comparación multiplataforma de técnicas basadas en visión artificial para detección de personas en espacios abiertos,” *Investig. e Innovación en Ing.*, vol. 9, no. 1, pp. 22–33, 2021. DOI: 10.17081/invinno.9.1.3965
- C. V. Niño-Rondón, S. A. Castro-Casadiago, B. Medina-Delgado, D. Guevara, and J. Gómez-Rojas, “Procesamiento a imágenes de video para verificación de distanciamiento social durante la pandemia de la COVID-19,” *Logos Cienc. Tecnol.*, vol. 13, no. 1, pp. 116–127, 2021. DOI: 10.22335/rict.v13i1.1305
- D. I. Gómez-Bedoya and R. Castrillón, “Reconocimiento automático de ganado bovino a partir de imágenes aéreas,” *III Congr. Int. en Intel. Ambient. Ing. Softw. y Salud Electrónica y Móvil*, pp. 32–39, 2019
- D. L. Siqueira and A. Manso Correa MacHado, “People Detection and Tracking in Low Frame-rate Dynamic Scenes,” *IEEE Lat. Am. Trans.*, vol. 14, no. 4, pp. 1966–1971, 2016. DOI: 10.1109/TLA.2016.7483541
- D. S. Aldás Salazar, S. M. Collantes Vaca, and J. P. Reyes Vásquez, “Procesamiento de imágenes con visión artificial para el reconocimiento de placas vehiculares,” *Sathiri*, no. 11, p. 133, 2016. DOI: 10.32645/13906925.15
- D. Villamarin, “Estado del Arte, Herramientas y Aplicaciones para Transformaciones geométricas 3D,” in *X Congreso de Ciencia y Tecnología ESPE 2015*, 2015, no. June 2015, pp. 1–6. DOI: 10.24133/cctespe.v10i1.51
- E. Alegre, G. Pajares, and A. de la Escalera, *Conceptos y métodos en visión por computador*, vol. 1. Madrid, España, 2016. ISBN: 978-84-608-8933-5
- E. Plebani, L. Celona, D. Pau, P. Karimi, and M. Marcon, “Training an object detector using only positive samples,” in *2015 IEEE 1st International Workshop on Consumer Electronics - Novi Sad, CE WS 2015*, 2017, no. January 2018, pp. 1–4.

- DOI: 10.1109/CEWS.2015.7867139
- F. Gao, Q. Xu, and B. Li, "Aircraft detection from VHR images based on circle-frequency filter and multilevel features," *Sci. World J.*, vol. 2013, no. March 2015, 2013. DOI: 10.1155/2013/917928
- F. Ramazankhani and M. Yazdian Dehkordi, "Iranian LicensePlateDetectionusingCascadeClassifier," in *ICEE 2019 - 27th Iranian Conference on Electrical Engineering, 2019*, pp. 1860–1863. DOI: 10.1109/IranianCEE.2019.8786468
- G. F. Contreras Contreras, B. Medina Delgado, D. G. Ibarra, C. Leite De Castro, and B. R. Acevedo Jaimes, "Cluster CV2: A Computer Vision Approach to Spatial Identification of Data Clusters," *2019 22nd Symp. Image, Signal Process. Artif. Vision, STSIVA 2019 - Conf. Proc.*, vol. 2, no. 1, pp. 1–5, 2019. DOI: 10.1109/STSIVA.2019.8730239
- J. A. Taquía-Gutiérrez, "El procesamiento de imágenes y su potencial aplicación en empresas con estrategia digital," *Interfases*, vol. 0, no. 010, p. 11, 2017. DOI: 10.26439/interfases2017.n10.1767
- J. Celis et al., "Voice processing with Internet of Things for a home automation system," in *Proceedings of the 2018 IEEE 25th International Conference on Electronics, Electrical Engineering and Computing, INTERCON 2018, 2018*, pp. 0–3. DOI: 10.1109/INTERCON.2018.8526420
- J. Hernández Santiago, J. S. Ruiz Castilla, C. H. Moreno Montiel, and B. Hernández Santiago, "Segmentación de placas vehiculares usando Haar-AdaBoost y Clustering," *Res. Comput. Sci.*, vol. 147, no. 5, pp. 269–279, 2018. DOI: 10.13053/rcs-147-5-20
- J. Navacerrada, "Sistema de detección de matrículas con Open CV," Universidad Politécnica de Madrid, 2017
- K. Selvaraj, A. A. Fathima, and V. Vaidehi, "Multi-class object detection by part based approach," in *International Conference on Recent Trends in Information Technology, ICRTIT 2012, 2012*, pp. 114–118. DOI: 10.1109/ICRTIT.2012.6206837
- K. Visakha and S. S. Prakash, "Detection and Tracking of Human Beings in a Video Using Haar Classifier," in *Proceedings of the International Conference on Inventive Research in Computing Applications, ICIRCA 2018, 2018*, no. Icirca, pp. 1–4. DOI: 10.1109/ICIRCA.2018.8597322
- L. Alejandro, S. Moncayo, and M. S. Barragán, "Dataset de contenidos musicales de video , basado en emociones Dataset of music video content , based on emotions," vol. 7, no. 1, pp. 37–46, 2016. DOI: 10.21500/20275846.2460
- L. Ramírez-Carvajal, K. Puerto-López, and J. Ramírez-Mateus, "Mathematical analysis of physical stability in the design of bipolar amplifiers through a computer tool," *J. Phys. Conf. Ser.*, vol. 1672, p. 012005, Oct. 2020. DOI: 10.1088/1742-6596/1672/1/012005
- Mohana and H. V. Ravish Aradhya, "Simulation of object detection algorithms for video surveillance applications," in *Proceedings of the International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2018, 2019*, pp. 651–655. DOI: 10.1109/I-SMAC.2018.8653665
- M. G. S. K. Sai Teja, M. R. Reddy, and R. Aishwarya, "Man-on-Man Brutality Identification on Video data using Haar Cascade Algorithm," in *Proceedings of the International Conference on Intelligent Computing and Control Systems, ICICCS 2020, 2020*, no. Iccics, pp. 274–278. DOI: 10.1109/ICICCS48265.2020.9120872
- O. D. Velasco-Delgado, M. F. Pérez-Sandoval, and J. F. Flórez- Marulanda, "Diseño y construcción de una estación de clasificación automática con visión de máquina," *TecnoLógicas*, vol. 17, no. 32, p. 97, 2014. DOI: 10.22430/22565337.200
- R. Cepeda Marín, "Fundamentos de visión artificial y su aplicación a la lectura de matrículas de coches," Universidad de Cantabria, 2018
- R. J. Moreno, O. Fernando, A. S. Dario, A. Hurtado, C. Gordillo, and C. F. Espinosa, "Detección vehicular mediante técnicas de visión de máquina," *ing@UAN*, vol. 2, no. 4, pp. 65–71, 2012
- R. Phase Tejas and S. Patil, "Building Custom

- HAAR-Cascade Classifier for face Detection,” *Int. J. Eng. Res.*, vol. V8, no. 12, pp. 881–886, 2020. DOI: 10.17577/IJERTV8IS120350
- R. Vera, B. Vintimilla, and H. Rodríguez, “Detección y Extracción de placas de vehículos en señales de video,” *Rev. Tecnológica ESPOL*, vol. 25, no. 1, p. 11, 2012
- V. K. Mishra, S. Kumar, and N. Shukla, “Image Acquisition and Techniques to Perform Image Acquisition,” *SAMRIDDHI A J. Phys. Sci. Eng. Technol.*, vol. 9, no. 01, pp. 21–24, 2017. DOI: 10.18090/samriddhi.v9i01.8333
- X. Zou, “A Review of object detection techniques,” in *Proceedings - 2019 International Conference on Smart Grid and Electrical Automation, ICSGEA 2019*, 2019, pp. 251–254. DOI: 10.1109/ICSGEA.2019.00065