

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/327160491>

# Software libre para reconocimiento automático de las nuevas patentes del Mercosur

Conference Paper · August 2016

CITATIONS

0

READS

602

5 authors, including:



**Javier Alejandro Jorge**

UNC Universidad nacional de corodoba

9 PUBLICATIONS 1 CITATION

SEE PROFILE



**José Amado**

Instituto Nacional de Tecnología Industrial

27 PUBLICATIONS 40 CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:



Weigh in motion WIM [View project](#)



COMPUTER VISION [View project](#)

# **Software libre para reconocimiento automático de las nuevas patentes del Mercosur**

Ignacio Moretti, Javier Jorge, José Amado, Cristian Caniglia, Daniel Puntillo, Marcos Blasco.

INSTITUTO NACIONAL DE TECNOLOGÍA INDUSTRIAL  
Avenida Vélez Sársfield 1561 X5000JKC - Córdoba – Argentina.  
(54) (0351) 460 3974/468 4835/4698304 internos 112/169/170.

labdei@inti.gob.ar

## Resumen

ANPR (Automatic number plate recognition) o Automatic license plate recognition (ALPR) son las siglas en ingles del reconocimiento automático de patentes, que básicamente consiste en la extracción de información de la patente del vehículo a partir de una imagen o una secuencia de imágenes. Históricamente, se ha aplicado en sistemas de seguridad, para el control de accesos de vehículos y estacionamientos. Hoy en día también se utiliza en el telepeaje, y el multado automático. Hay muchas herramientas disponibles en el mercado que tienen esta facultad, la gran mayoría tienen que ser entrenadas para reconocer las matrículas de diferentes países, provincias o estados. Estas placas suelen contener diferentes colores, están escritos en diferentes idiomas, y el uso de diferentes fuentes; algunas placas pueden tener un solo color de fondo y otros tienen imágenes de fondo. Por lo que la calidad de las imágenes adquiridas es un factor importante en el éxito de la ALPR. En este trabajo, se presenta una implementación de un sistema ALPR basado en software libre, capaz de reconocer el nuevo formato de placa de matrícula Mercosur. El documento describe las herramientas utilizadas, su principio de funcionamiento, y los cambios necesarios para su correcto funcionamiento. El sistema ha sido probado con fotografías reales de Uruguay y Argentina y además con fotografías generadas digitalmente de Argentina y Brasil. La prueba realizada muestra que el sistema implementado es capaz de ofrecer tasas de reconocimiento entre el 95 y el 98%. EL Reconocimiento automático de Patentes se considera como una aplicación específica de varias ramas de estudio e investigación como lo son el procesamiento de imágenes, la extracción de información a partir de imágenes, la inteligencia artificial, el reconocimiento de patrones, el reconocimiento de caracteres entre otras. Existen muchos trabajos e investigaciones realizadas donde abordan esta problemática de distintos puntos de vistas aplicando diferentes metodologías y herramientas para tratar de resolver este problema específico [1][2][3][4][5].

## Palabras claves

ANPR, ALPR, Procesamiento de imágenes, Inteligencia artificial, OpenCV, OpenALPR, OCR, TesseractOCR, Placas, Patentes, Vehículos.

# Introducción

El proceso general del reconocimiento automático de patentes podría resumirse en varias etapas bien definidas como muestra la Figura 1.



**Figura 1:** Etapas del proceso general de reconocimiento automático de patentes.

Las distintas aplicaciones, desarrollos e investigaciones realizadas difieren en las técnicas utilizadas para cubrir cada etapa y cumplir con los objetivos específicos de cada una de ellas. La problemática central es el reconocimiento de las nuevas patentes del Mercosur para vehículos particulares cuyas características mas relevantes para este desarrollo son:

1. Dispositivo dotado de un arreglo de siete caracteres que consta de letras y números y conforma un serial, embozado en alto relieve.
2. Color de fondo Blanco. Bandera del país, emblema del Mercosur, Marca de agua, tipo ensure, estampado en caliente con lamina de seguridad con efecto difractivo y onda sinusoidal.

3. Caracteres en color Negro.
4. Dimensiones: Largo 400mm +-2mm, Alto 130mm +- 2mm.
5. Tipo de letra: FEEngschrift, Alto del carácter 65mm.

Las demás características de las patentes del Mercosur pueden ser consultadas en la resolución correspondiente [8]. Además de las características de la patente es importante reconocer este objeto en distintas condiciones (iluminación, posición, etc), para que el software desarrollado sea robusto a los cambios en el hardware e independiente del medio de adquisición de las imágenes.

## Objetivos

Estudiar investigaciones y desarrollos con el fin de definir las mejores herramientas a utilizar para cumplir con el objetivo de realizar una implementación de software libre para el reconocimiento de las nuevas patentes del Mercosur. Construir una base de datos de imágenes de vehículos particulares con patentes del Mercosur, desarrollar una herramienta de software libre capaz de reconocer automáticamente las patentes del Mercosur de vehículos particulares, registrar y publicar los resultados obtenidos aplicando este software a la base de datos de imágenes generada.

## Propuesta de Solución

Según la investigación previa del estado del arte en este área específica y basándonos en los resultados publicado en el trabajo realizado por Chirag Patel, Dipti Shah y Atul Patel [6] donde se puede apreciar una comparación de las diferentes estrategias y métodos utilizados para resolver la etapa más crítica del proceso que es el reconocimiento de caracteres (OCR), como muestra la Tabla 1, iniciamos una búsqueda de herramientas y librerías de software libres disponibles.

Ref.	Método	Porcentaje Éxito(%)	Tipo de Categoría
[9]	Two layer PNN	89,1	Letras
[10]	Statistical feature extraction.	85	No reportado
[11]	Feature salient	95,7	No reportado
[12]	SVM Integration with feature extraction	93,7	Caracteres Ingleses, Chinos, Numeros, Kana
[13]	Template matching	No reportado	letras y dígitos
[14]	multi layered perceptron (MLP) ANN	98,17	letras y dígitos
[15]	multi layered perceptron (MLP) ANN	No reportado	letras y dígitos
[16]	Open source OCR Tesseract	98,7	letras y dígitos
[17]	BP neural network	No reportado	letras y dígitos Koreanos
[18]	BP neural network	No reportado	letras y dígitos Ingles, letras Chinas
[19]	MRF	No reportado	letras y dígitos
[20]	character categorization, topological sorting, and self-organizing (SO) recognition	95,6	letras y dígitos
[21]	Hierarchical Neural Network(HNN)	95,2	letras y dígitos
[22]	PNN	96,5	letras y dígitos
[23]	BP neural network	93,5	letras y dígitos

**Tabla 1:** Comparativa de métodos para reconocer caracteres y los resultados obtenidos.

La herramienta elegida fue OpenALPR[24]. OpenALPR funciona como una tubería. La entrada es una imagen, luego esta imagen es procesada en diferentes etapas en cascada, y la salida son posibles números de placa que se encuentren en la imagen. Estas etapas de procesamiento siguen un orden preestablecido y con un objetivo particular en cada una de ellas como muestra la Tabla 2.

Etapas	clase C++ código fuente	Descripción
Detección	regiondetector.cpp	Encuentra las regiones potenciales donde pueda existir una placa.
Binarización	binarizewolf.cpp	Convierte la región potencial anterior en blanco y negro.
Análisis de Caracteres	characteranalysis.cpp	Encuentra "manchas" o "blobs" de tamaño de los caracteres en la región blanco y negro anterior.
Búsqueda de bordes de patente	platelines.cpp and platecorners.cpp	Busca los bordes de la placa.
Enderezamiento	licenseplatecandidate.cpp	Transforma la perspectiva a una vista directa basado en el tamaño ideal de la placa.
Segmentación de caracteres	charactersegmenter.cpp	Ubica cada carácter, aumenta el brillo y elimina el ruido para que puedan ser procesados de forma individual.
OCR Reconocimiento de caracteres	ocr.cpp	Analiza cada imagen de carácter y proporciona posibles letras/números.
Posprocesamiento	postprocess.cpp	Crea una lista de n posibles placas ordenadas de mayor a menor confianza según el OCR. También realiza comprobación contra una secuencia preestablecida.

**Tabla 2:** Etapas código fuente y descripción de OpenALPR.

## **Detección**

La fase de detección comienza una vez que la imagen es capturada o enviada para su procesamiento. Se utiliza el algoritmo de LBP[25][26] (generalmente utilizado para la detección de rostros) para encontrar posibles regiones de placa (x , y, ancho , altura). Cada una de estas regiones se envía a las posteriores fases del proceso para su posterior procesamiento. La fase de detección es por lo general la fase mayor coste computacional. Puede ser acelerado con el uso de la GPU para mejorar el rendimiento.

## **Binarización**

Esta fase se repite varias veces, una vez para cada posible región de matrícula detectada en la fase anterior. La etapa de binarización crea múltiples imágenes binarias (Blanco y Negro) para cada posible región de placa. Se generan imágenes binarias con diferentes cotas de binarización para obtener la mejor oportunidad para encontrar todos los caracteres. Una sola imagen binarizada puede perder caracteres si la imagen es demasiado oscura o demasiado clara. En esta etapa se utiliza el método de Wolf-Jolion [27] , así como el método de Sauvola [28] con diversos parámetros. Cada una de estas imágenes binarias se procesa en las fases posteriores.

## **Análisis de Caracteres**

El análisis de caracteres intenta encontrar zonas de tamaño predeterminado según el tamaño original de los caracteres especificados en la patente (esto puede ser configurable) en la imagen binaria que se le pasa como entrada a esta etapa. Esto se hace mediante la búsqueda de “manchas” o “blobs” conectados en la región central en la imagen binarizada de la posible placa. A continuación, busca manchas que son aproximadamente la anchura y la altura de un carácter de placa, que además estén alineadas en línea recta entre si y sean de anchura y altura similar. Este análisis se realiza varias veces en la región. Se inicia mediante la búsqueda de caracteres pequeños, y a continuación, se analizan caracteres más grandes. Si no se encuentra nada en la región, entonces la región es descartada y no sigue con las siguientes etapas. Si encuentra algunos caracteres posibles, entonces la esa región se pasa como entrada al posterior procesamiento.

## **Búsqueda de bordes de patente**

La siguiente fase consiste en encontrar los bordes de la matrícula. La fase de detección sólo

es responsable de la identificación de una posible región donde puede existir una placa de matrícula. En general se proporciona una región que es un poco más grande o más pequeña que la placa real. Los bordes de la placa que se trata de encontrar en esta etapa son los mejores bordes superior, inferior, izquierdo y derecho de la placa. El primer paso es encontrar todas las líneas con el algoritmo de la transformada de Hough [29][30] para la región de la matrícula. En el archivo "platelines.cpp" se encuentra el código fuente donde se observa el procesamiento de la imagen de la placa y calcula una lista de líneas horizontales y verticales. Luego se utiliza esta lista, así como la altura de los caracteres (calculado en el análisis de caracteres) para encontrar los bordes más probables. Se utiliza un número de pesos configurables para determinar qué borde tiene más sentido. Utilizando un borde predeterminado (basado en el ideal de anchura / altura de la placa) para elegir el que más se asemeje.

## **Enderezamiento**

Dados los bordes de la placa, la etapa de corrección de inclinación vuelve a asignar la región de la placa a un tamaño y orientación estándar. Esto nos dará una imagen de la placa orientada correctamente (eliminando rotación e inclinación).

## **Segmentación de caracteres**

La fase de segmentación de caracteres trata de aislar todos los caracteres que componen la imagen de la placa. Se utiliza un histograma vertical para encontrar huecos en los caracteres de la placa. Esta fase también limpia las cajas de caracteres mediante la eliminación de manchas pequeñas desconectadas y descalifica regiones de caracteres que no son bastante altos. Asimismo, trata de eliminar regiones "borde" de modo que el borde de la placa no sea clasificado como un '1' o un 'l'.

## **OCR Reconocimiento de caracteres**

La fase OCR analiza cada carácter independientemente. Para cada imagen de carácter calcula todos los caracteres posibles y su nivel de confianza. Para esta etapa se utiliza un herramienta de software libre llamada TesseractOCR [31] previamente entrenada.

## **Posprocesamiento**

Dada una lista de todos los posibles caracteres y sus niveles de confianza, el procesamiento posterior determina las mejores combinaciones de letras y números en la posible placa. Se

organizada como una lista de las N superiores según su nivel de confianza. El procesamiento posterior descalifica a todos los caracteres por debajo de un determinado umbral. También tiene umbrales "blandas" ya que es posible que el carácter de bajo nivel de confianza no es realmente parte de la placa. En esta etapa es posible validar de la región geográfica a la que pertenece la placa, por ejemplo Argentina, Brasil, Uruguay, Paraguay, Venezuela, etc. Esto es configurable en OpenALPR a través del uso de "templates" de patentes, para que coincida con un formato dado ([Letra][Letra] [Número][Número][Número] [Letra][Letra]). Así, por ejemplo, si la lista fue:

1. AAZ86FQ
2. AA2B6FQ
3. AA286FQ

La tercera entrada coincide con la plantilla, pero los otros dos no lo hacen. Por lo tanto, el procesamiento posterior será la señal de que la tercera entrada es nuestro mejor candidato.

La Imagen 1 describe la entrada de la aplicación, la Imagen 2 resultado intermedio de la etapa de Binarización. La Imagen 4 y 5 muestran resultados intermedios de la etapa de búsqueda de bordes de patente. La Imagen 6, 7, 8 y 9 muestran resultados intermedios de la etapa de Segmentación de caracteres. Por último la Imagen 10 el resultado final con OCR.



**Imagen 1:** Imagen de entrada original.



Imagen 2: Detección.



Imagen 3: Binarización.

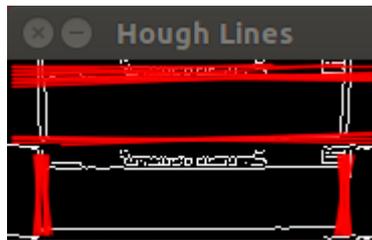


Imagen 4: Búsqueda de bordes de patente.



Imagen 5: Búsqueda de bordes de patente.

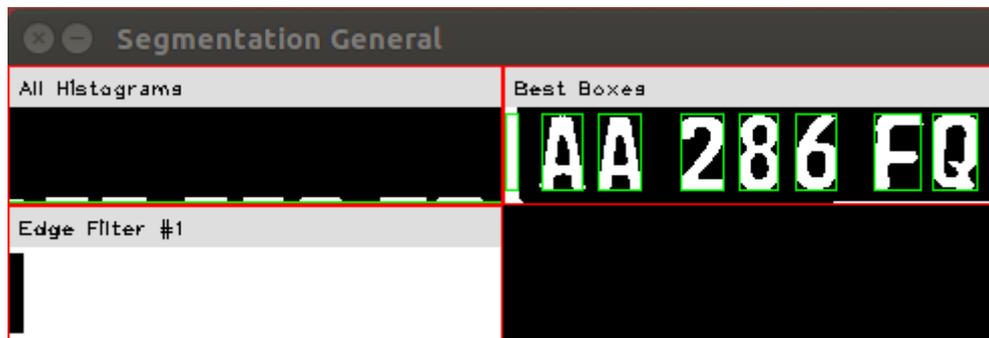


Imagen 6: Segmentación de Caracteres



Imagen 7: Segmentación de Caracteres.



Imagen 8: Segmentación de Caracteres.



Imagen 9: Segmentación de caracteres.



Imagen 10: Resultado Final con resultado de OCR más confiable y con formato buscado.

## Desarrollo

A continuación describiremos el trabajo realizado en la construcción de esta herramienta.

### Base de Datos de Imágenes

El primer paso fue la búsqueda de imágenes reales de autos con patentes del Mercosur. Cuando fue iniciado este proyecto solo se encontraban disponibles imágenes de automóviles Uruguayos ya que en los otros países no se había comenzado con la utilización de estas nuevas patentes. Decidimos Crear algunas imágenes digitalizadas no reales con un

modelo de patente según el prototipo publicado. El primer prototipo de este software tomo como base de pruebas un banco de datos de 20 imágenes reales y generadas digitalmente. A medida que fueron apareciendo vehículos con estas nuevas patentes en Argentina, se han incorporado nuevas imágenes al banco de datos. Actualmente contamos con un banco de 200 imágenes anotadas (134 imágenes positivas que contienen una patente del Mercosur y 66 imágenes negativas que no contienen ninguna patente del Mercosur). Las imágenes que conforman este banco de datos son de diferentes características ya que se planteo poder construir una aplicación que no dependa de un tipo específico de resolución iluminación ángulo de captura, etc. En la Tabla 3 podemos observar diferentes características de las imágenes que forman parte de esta base de datos generada.

Nombre	Resolución	Origen Captura	Iluminación
auto85.jpg	2560x1536	Cámara digital	Natural día
auto127.jpg	2560x1536	Cámara digital	Natural día
auto103.jpg	2560x1536	Cámara digital	Artificial noche
auto4.jpg	280x160	Internet	Natural día
auto20.jpg	500x281	Internet	Natural día
auto57.jpg	1398x410	Internet	Artificial Interior
auto58.jpg	600x800	Internet	Natural día
auto51.jpg	672x420	Internet	Generada digitalmente
auto12.jpg	1024x767	Internet	Generada digitalmente

**Tabla 3:** Características de algunas imágenes de la base de datos generada.

## Configuración de OpenALPR

Para adaptar la herramienta OpenALPR fue necesario modificar las configuraciones por defecto, trabajamos en especificar las dimensiones de la placa a identificar, dimensiones de los caracteres tamaño de los espacios en blanco entre caracteres, tamaño máximo y mínimo de pixeles (ancho y alto) de la placa a ser analizada proporcional al tamaño real en milímetros, lenguaje del OCR a ejecutar, etc.[32]. Luego de configurar estas especificaciones se configuro el formato de los caracteres para el posprocesamiento, para esto se especifico tanto la patentes de cada país como se muestra en la Imagen 11.



**Imagen 11:** Formato de patentes del Mercosur.

## Programación del entorno de pruebas

Una vez configurada la aplicación OpenALPR para que reconozca las nuevas patentes del Mercosur se desarrolló un programa de shell scripts para la ejecución automática de la aplicación OpenALPR sobre la base de datos creada anteriormente. Este programa ejecuta el reconocedor de patentes por cada imagen de la base de datos y verifica el resultado obtenido con el resultado esperado que fue anotado anteriormente. Además registra la cantidad de casos exitosos y fallidos para poder calcular el nivel de confianza de la aplicación. Esto nos permite tener un factor de comparación con otras aplicaciones similares.

## Desarrollo de programa en c

En base a un ejemplo de uso de OpenALPR como librería C/C++ [33], desarrollamos una software como prototipo de ejecución. Esta aplicación ejecuta el reconocedor de patentes y como resultado escribe en la imagen de entrada el reconocimiento y la zona de la detección realizada como muestra la Imagen 12.



Imagen 12: Resultado final de aplicación en C del uso de la API C/C++.

## Resultados

Los resultados obtenidos sobre la base de datos de imágenes fueron los siguientes:

- De un total de **200** imágenes fueron reconocidas correctamente **195**, contando positivas y negativas.
- Esto nos da un nivel de confianza sobre esta base de datos de **97,50%**
- Los 5 errores detectados fueron solo 1 falso negativo y 4 errores en solo un carácter de los siete encontrados:
  - En la imagen auto26t.jpg el resultado esperado fue SBZ4799 y el resultado obtenido fue que no se encontró ninguna patente.(Posiblemente el error fue generado por un reflejo en la imagen que distorsiono la imagen de la patente

véase Imagen 13)

- En la imagen auto31t.jpg el resultado esperado fue AAO1535 y el resultado obtenido fue AAQ1535 error en el 3<sup>er</sup> caracter “O” por “Q”.
- En la imagen auto33.jpg el resultado esperado fue AA068GN y el resultado obtenido fue AA060GN error en el 5<sup>to</sup> caracter “8” por “0”.
- En la imagen auto34.jpg el resultado esperado fue AA068GN y el resultado obtenido fue AA060GN error en el 5<sup>to</sup> caracter “8” por “0”.
- auto67.jpg::AA017WV::AA017HV
- En la imagen auto67.jpg el resultado esperado fue AAO17WV y el resultado obtenido fue AA017HV error en el 3<sup>er</sup> caracter “W” por “H”.



**Imagen 13:** Única Imagen falso negativo.

- En cuanto a los tiempos de ejecución de esta herramienta podemos observar la Tabla 4 donde se especifica según imagen de entrada el tiempo en ms en reconocer o no la patente. Las características del hardware de la PC utilizada y su sistema operativo fue Intel® Core™ i5-2310 CPU @ 2.90GHz x 4, 4GB de RAM, SO Ubuntu 14.04 LTS. Analizando estos datos obtenemos un promedio de 138 ns (nanosegundos) por pixel.

Entrada	Resolución	Tiempo en ms
auto85.jpg	2560x1536	436
auto127.jpg	2560x1536	359
auto103.jpg	2560x1536	392
auto4.jpeg	280x160	49
auto20.jpg	500x281	60
auto57.jpg	1398x410	176
auto58.jpg	600x800	125
auto51.jpg	672x420	104
auto12.jpg	1024x767	257

**Tabla 4:** Tiempo de cálculo en milisegundos.

## Conclusiones

Se trabajó en la construcción de un banco de datos de imágenes reales y generadas digitalmente para probar las distintas herramientas disponibles, se configuraron las herramientas y se realizaron pruebas sobre el banco de datos generado. Este desarrollo fue planteado como un módulo de un proyecto realizado anteriormente (balanza de pesaje y clasificación dinámica de vehículos de alta velocidad)[7]. Claramente el objetivo planteado de construir una aplicación con herramientas de software libre que sea capaz de reconocer las nuevas patentes del Mercosur fue cumplido. Considerando que este trabajo es un prototipo y en base al tamaño de la base de datos generada podemos concluir que los resultados preliminares que se obtuvieron son más que satisfactorios, llegando a un nivel de confianza de la aplicación del 97,50%. Como trabajo futuro planteamos comparar esta aplicación con otras aplicaciones disponibles utilizando esta misma base de datos generada. Otro de los trabajos a futuro es integrar esta aplicación como un módulo a la balanza de pesaje dictaminada desarrollada en INTI [7], además poder portar esta aplicación a otros dispositivos.

## Bibliografía

- [1] Anuja P. Nagare, "License Plate Character Recognition System using Neural Network", *International Journal of Computer Applications* (0975 – 8887) Volume 25–No.10, 2011.
- [2] Amr Badr, Mohamed M. Abdelwahab, Ahmed M. Thabet, y Ahmed M. Abdelsadek, "Automatic Number Plate Recognition System", *Annals of the University of Craiova, Mathematics and Computer Science Series* Volume 38(1), Pages 62–71, ISSN: 1223-6934, , 2011.
- [3] Dhiraj Y. Gaikwad y Pramod B. Borole, "A Review Paper on Automatic Number Plate Recognition (ANPR) System", *International Journal of Innovative Research in Advanced Engineering (IJIRAE)*, Volume 1, Issue 1, 2014.
- [4] Muhammad Tahir Qadri y Muhammad Asif, "Automatic Number Plate Recognition System For Vehicle Identification Using Optical Character Recognition", *International Conference on Education Technology and Computer* , 978-0-7695-3609-5/09 © 2009 IEEE DOI 10.1109/ICETC.2009.54, 2009.
- [5] R. T. Lee, K. C. Hung, y H. S. Wang, "Real Time Vehicle License Plate RecognitionBased on 2D Haar Discrete Wavelet Transform", *International Journal of Scientific & Engineering Research*, Volume 3, Issue 4, ISSN 2229-5518, 2012.
- [6] Chirag Patel, Dipti Shah y Atul Patel, "Automatic Number Plate Recognition System (ANPR): A Survey", *International Journal of Computer Applications* (0975 –8887) Volume 69–No.9, 2013.
- [7] R. Muñoz, J. Amado, D. Puntillo, C. Caniglia, J. Jorge, L. Chalimond, C. Galanzino y G. Alessandrini, "Sistema De Clasificación Y Pesaje Dinámico De Vehículos", 9º encuentro INTI de presentación de trabajos, Paguinas 154-155, 2010.
- [8] Mercosur, Patente y sistema de consultas sobre vehículos del Mercosur, *Mercosur/GMC/RES. N° 33/14*, 2014.
- [9] Christos Nikolaos E. Anagnostopoulos, Ioannis E. Anagnostopoulos, Vassili Loumos, and Eleftherios Kayafas, "A License Plate-Recognition Algorithm for Intelligent Transportation System Applications," pp. 377-392, 2006.
- [10] Prathamesh Kulkarni, Ashish Khatri, Prateek Banga, and Kushal Shah, "Automatic Number Plate Recognition (ANPR)," in *RADIOELEKTRONIKA. 19th International Conference*, 2009.
- [11] Zhen-Xue Chen, Cheng-Yun Liu, Fa-Liang Chang, and Guo-You Wang, "Automatic License-Plate Location and Recognition Based on Feature Saliance," *IEEE Transactions on Vehicular Technology*, vol. 58, no. 7, pp. 3781-3785, 2009.
- [12] Ying Wen et al., "An Algorithm for License Plate recognition Applied to Intelligent

Transportation System," IEEE Transactions of Intelligent Transportation Systems, pp. 1-16, 2011.

[13] Ch.Jaya Lakshmi, Dr.A.Jhansi Rani, Dr.K.Sri Ramakrishna, and M. KantiKiran, "A Novel Approach for Indian License Recognition System," International Journal of Advanced Engineering Sciences and Technologies, vol. 6, no. 1, pp. 10-14, 2011.

[14] H. Erdinc Kocer and K. Kursat Cevik, "Artificial neural networks based vehicle license plate recognition," Procedia Computer Science, vol. 3, pp. 1033-1037, 2011.

[15] A Roy and D.P Ghoshal, "Number Plate Recognition for use in different countries using an improved segmentation," in 2nd National Conference on Emerging Trends and Applications in Computer Science(NCETACS), pp. 1-5, 2011.

[16] Lihong Zheng, Xiangjian He, Bijan Samali, and Laurence T. Yang, "An algorithm for accuracy enhancement of license recognition," Journal of Computer and System Sciences, 2012.

[17] Kaushik Deb, Ibrahim Kahn, Anik Saha, and Kang-Hyun Jo, "An Efficient Method of Vehicle License Plate Recognition Based on Sliding Concentric Windows and Artificial Neural Network," Procedia Technology, vol. 4, pp. 812-819, 2012.

[18] Zhigang Zhang and Cong Wang, "The Research of Vehicle Plate Recognition Technical Based on BP Neural Network," AASRI Procedia, vol. 1, pp. 74-81, 2012.

[19] Yuntao Cui and Qian Huang, "Extracting character of license plates from video sequences," Machine Vision and Applications, Springer Verlag, p. 308, 1998.

[20] Shyang-Lih Chang, Li-Shien Chen, Yun-Chung Chung, and Sei-Wan Chen, "Automatic license plate recognition," IEEE Transactions on Intelligent Transportation Systems, vol. 5, no. 1, pp. 42-53, 2004.

[21] Nicolas Thome, Antoine Vacavant, Lionel Robinault, and Serge Miguet, "A cognitive and video-based approach for multinational License Plate Recognition," Machine Vision and Applications, Springer-Verlag, pp. 389-407, 2011.

[22] Fikriye Öztürk and Figen Özen, "A New License Plate Recognition System Based on Probabilistic Neural Networks," Procedia Technology, vol. 1, pp. 124-128, 2012.

[23] Yifan Zhu, Han Huang, Zhenyu Xu, Yiyu He, and Shiqiu Liu, "Chinese-style Plate Recognition Based on Artificial Neural Network and Statistics," Procedia Engineering, vol. 15, pp. 3556-3561, 2011.

[24] OpenALPR, <http://www.openalpr.com/>.

[25] DC. He y L. Wang, "Texture Unit, Texture Spectrum, And Texture Analysis", Geoscience and Remote Sensing, IEEE Transactions on, vol. 28, pp. 509 – 512, 1990.

[26] L. Wang y DC. He, "Texture Classification Using Texture Spectrum", Pattern Recognition, Vol. 23, No. 8, pp. 905 – 910, 1990.

- [27] C. Wolf y J-M. Jolion, "Extraction and Recognition of Artificial Text in Multimedia Documents", Pattern Analysis and Applications, 6(4):309-326, 2003.
- [28] J.Sauvola, T,Seppanen, S.Haapakoski y M.Pietikainen, "Adaptive Document Binarization", 4th Int. Conf. On Document Analysis and Recognition, Ulm, Germany, pp.147-152, 1997.
- [29] HOUGH, P. V. C. Method and means for recognizing complex patterns. U. S. Patent 3, 069 654, 1962.
- [30] Duda, R. O. y P. E. Hart, "Use of the Hough Transformation to Detect Lines and Curves in Pictures," Comm. ACM, Vol. 15, pp. 11–15 , 1972.
- [31] TesseractOCR, <https://github.com/tesseract-ocr/tesseract>.
- [32] Training OpenALPR, <https://github.com/openalpr/train-ocr>