

APLICACIÓN DEL MODELO DE INTERSECCIÓN CORTICAL PARA SEGMENTAR CARACTERES EN UNA PLACA VEHICULAR

*USING INTERSECTING CORTICAL MODEL FOR CHARACTER
SEGMENTATION ON A LICENSE PLATE*

Elizabeth Xicotencatl Flores

Benemérita Universidad Autónoma de Puebla, México
elizabeth.xicotencatlf@alumno.buap.mx

Aldrin Barreto Flores

Benemérita Universidad Autónoma de Puebla, México
aldrin.barreto@correo.buap.mx

Salvador Eugenio Ayala Raggi

Benemérita Universidad Autónoma de Puebla, México
salvador.raggi@correo.buap.mx

Verónica Edith Bautista López

Benemérita Universidad Autónoma de Puebla, México
veronica.bautistalo@correo.buap.mx

José Francisco Portillo Robledo

Benemérita Universidad Autónoma de Puebla, México
francisco.portillo@correo.buap.mx

Recepción: 12/mayo/2021

Aceptación: 9/noviembre/2021

Resumen

Los sistemas de reconocimiento automático de placas vehiculares han sido mundialmente estudiados utilizando diversas técnicas de visión por computadora e inteligencia artificial. En este trabajo se presenta la aplicación de un modelo simplificado de una red neuronal pulso acoplada para separar los caracteres en una placa. Los pulsos óptimos para seleccionar la imagen binaria se determinan a partir del uso del valor de la entropía para descartar aquellos donde los caracteres no son visibles. A la imagen resultante se le aplica un análisis de regiones individuales para obtener sólo los objetos de interés y a partir de ello separar cada carácter alfanumérico. El algoritmo se evaluó en imágenes de placas vehiculares obtenidas

en condiciones no controladas permitiendo tener resultados favorables en la mayoría de los casos.

Palabras Clave: Redes neuronales artificiales, segmentación de caracteres, reconocimiento de imágenes.

Abstract

Automatic license plate recognition systems have been studied worldwide using several computer vision and artificial intelligence techniques. In this work we present the application of a simplified model of a pulse coupled neural network to separate characters on a plate. The optimal pulses to select the binary image are determined using an entropy value, to discard those where the characters are not visible. Individual regions analysis is applied to the resulting image to obtain only the objects of interest and then separate each alphanumeric character. The algorithm was evaluated on images of license plates obtained under uncontrolled conditions, allowing favorable results in most of the cases.

Keywords: Artificial neural networks, character segmentation, image recognition.

1. Introducción

Los sistemas de reconocimiento vehicular han tomado gran relevancia en los últimos años debido al crecimiento poblacional y sus implicaciones en cuestión de seguridad. Como el nombre lo sugiere, tienen como finalidad identificar vehículos a través de los caracteres de las matrículas. Generalmente constan de cuatro algoritmos principales definidos como [Kulkarni, 2012]: preprocesamiento, detección de la placa, segmentación y reconocimiento de caracteres.

Parte fundamental de estos sistemas recae en el preprocesamiento y la detección de la placa debido a que su desempeño es variable en entornos no controlados. Trabajos previos han propuesto alternativas que incluyen el uso de operadores morfológicos [Hongping, 2011], conexiones en cascada con K-means [Zheng, 2011], coincidencia de plantillas [Jalil, 2015], proyecciones horizontales y verticales [Kakani, 2017], máquinas de vectores de soporte [Lin, 2018], redes neuronales convolucionales [Wang, 2019] y recientemente el uso de You Only Look Once

(YOLO) [Henry, 2020] por ser un detector y clasificador de objetos de video en tiempo real. No obstante, todos tienen sus propias limitaciones y áreas de oportunidad para aumentar el porcentaje final de reconocimiento por las variantes presentes en las condiciones de captura y el formato entre países. En este trabajo de investigación se presenta una alternativa de preprocesamiento dirigido a la segmentación de la secuencia de caracteres alfanuméricos de una matrícula vehicular que incluye como elemento principal el modelo de intersección cortical. En la sección 2 se muestra la metodología del algoritmo propuesto, así como conceptos utilizados para el análisis de la imagen. En la sección 3 los resultados experimentales y en la sección 4 la discusión de resultados tras valorar la efectividad del algoritmo.

2. Metodología

En el diagrama de la figura 1 se presentan los pasos para implementar el algoritmo de segmentación de caracteres.

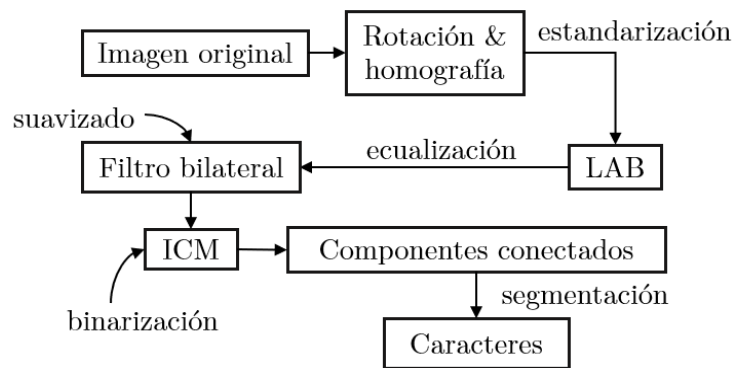
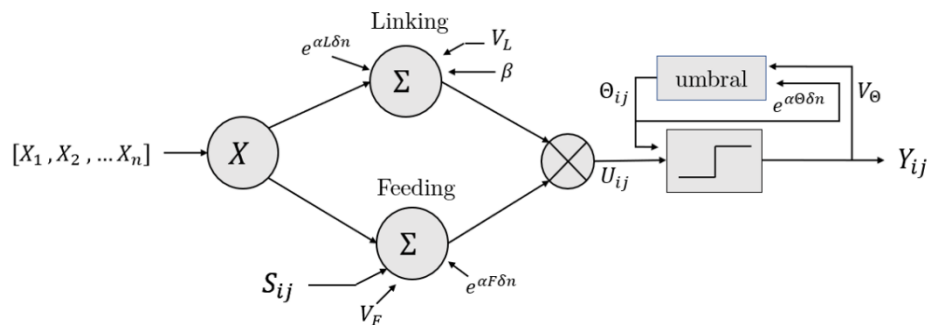


Figura 1 Etapas de segmentación.

En vista de no encontrar un conjunto de datos disponible al público con matrículas de México, se creó un banco de imágenes conformado por 730 placas capturado a través de un smartphone con una cámara digital de 13 megapíxeles considerando tanto la vista frontal como la trasera de automóviles particulares. Todas fueron adquiridas bajo diferentes condiciones (distorsiones, niveles de exposición y ruido) entre las 9:00 y las 18:00 horas.

Redes neuronales de tercera generación

Las Redes Neuronales Pulso Acopladas (PCNN) también conocidas como redes neuronales artificiales de tercera generación [Ortiz, 2017], son un sistema que permite emular las neuronas biológicas de la corteza visual de los mamíferos y se distinguen por no incluir una etapa de entrenamiento. Han sido aplicadas en segmentación de distintos tipos de imágenes [Cheng, 2018] incluyendo las de tipo SAR (Synthetic Aperture Radar) [Li, 2008], clasificación y reconocimiento de texturas [Ma, 2010]. El modelo original representado por la figura 2, expone que cada neurona recibe las señales *Linking* y *Feeding* ponderadas por un factor β para crear la respuesta U . Ambas señales son afectadas por los factores de atenuación V_L, V_F y amplificación $e^{\alpha L \delta n}, e^{\alpha F \delta n}$, al igual que el umbral por $e^{\alpha \theta \delta n}$.

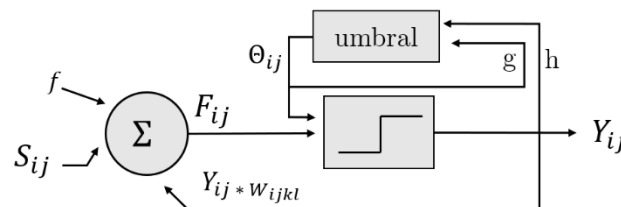


Fuente: Elaboración propia

Figura 2 Diagrama PCNN.

Modelo de Intersección Cortical (ICM)

El modelo ICM es una versión simplificada de una red neuronal pulso acoplada diseñada para minimizar el costo computacional implícito en el procesamiento de imágenes. La arquitectura de esta red se muestra en la figura 3.



Fuente: Elaboración propia

Figura 3 Arquitectura ICM.

La ecuación 1 representa a la señal *Feeding*, donde, cada neurona recibe como entrada un píxel de la imagen *S* en la posición *ij*, y la retroalimentación de la red con la salida *Y* ponderada por *W*, una matriz generalmente gaussiana asociada a la conexión entre los píxeles.

$$F_{ij}[n + 1] = fF_{ij}[n] + S_{ij} + W\{Y[n]\}_{ij} \quad (1)$$

El resultado de *F* se compara con un umbral dinámico θ para generar mediante las iteraciones los pulsos de salida representados por *n* imágenes binarias, las cuales se obtienen con la ecuación 2.

$$Y_{ij}[n + 1] = \begin{cases} 1 & \text{si } F_{ij}[n + 1] > \theta_{ij}[n] \\ 0 & \text{En otro caso} \end{cases} \quad (2)$$

El valor del umbral es dinámico cuando se activa la neurona y su valor se controla con la ecuación 3, siendo *f*, *g* y *h* las constantes que ajustan el modelo.

$$\theta_{ij}[n + 1] = g\theta_{ij}[n] + hY_{ij}[n + 1] \quad (3)$$

Descripción del algoritmo

El modelo de color LAB se añadió con el propósito de aumentar las características de cada imagen de muestra, permitiendo modificar el componente de luminosidad "L" con la ecualización de su histograma sin alterar los valores de tono y saturación de los dos canales de color restantes con las tonalidades rojo-verde y azul-amarillo.

Durante la conversión se utilizó el modelo CIEXYZ para obtener las nuevas componentes RGB relativas a un punto blanco definido como $C=(X,Y,Z)$. La matriz de transformación se muestra en la ecuación 4.

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} = M \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 0.412453 & 0.357580 & 0.180423 \\ 0.212671 & 0.715160 & 0.072169 \\ 0.019334 & 0.119193 & 0.950227 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (4)$$

La ecualización logró distribuir los niveles de gris resaltando la zona de la placa en automóviles blancos donde la distribución de los niveles de intensidad es similar, este proceso se muestra en la figura 4.

Posteriormente se agregó un filtro bilateral, que reemplaza la intensidad de cada píxel por una media ponderada de los valores de sus vecinos de tal forma que

suaviza la imagen manteniendo los bordes [Durand, 2002]. El resultado de aplicar este filtro no lineal a un píxel s se expresa en la ecuación 5.

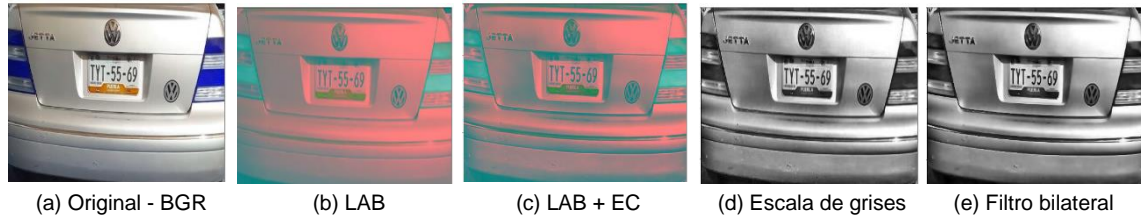


Figura 4 Transformación LAB.

$$J_s = \frac{\sum_{p \in \Omega} f(p-s)g(I_p - I_s)I_p}{\sum_{p \in \Omega} f(p-s)g(I_p - I_s)} \quad (5)$$

Donde f y g son funciones gaussianas, I representa la intensidad de los píxeles y $(p-s)$ la distancia entre el píxel y sus vecinos. En la figura 5 se muestran otras técnicas de suavizado.

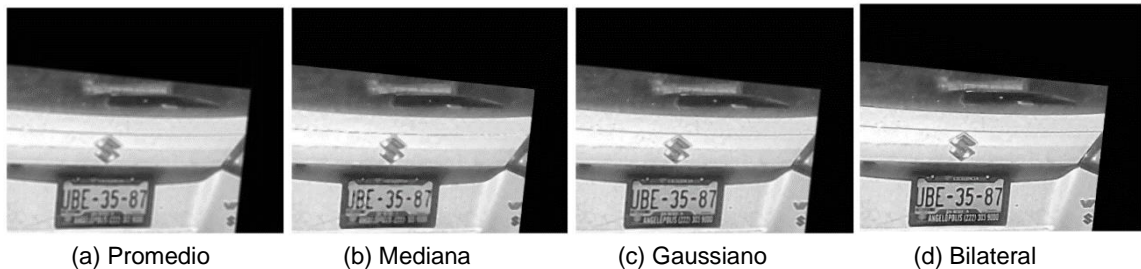


Figura 5 Comparación entre técnicas de filtrado.

Como se observa, el filtro bilateral resalta por difuminar áreas ajenas a los caracteres preservando los bordes. Después la imagen difuminada se ingresó a la red neuronal transformándola previamente a escala de grises para trabajar con un solo canal y poder normalizar los valores para que el rango de intensidad en lugar de ser de 0 a 255, se considere de 0 a 1.

- *Primera parte: Preprocesamiento*
- Obtener el conjunto de muestras
- Redimensionar a 430x350 píxeles
- Convertir espacio de color BGR a LAB

- Separar los tres canales
- Ecuilibrar el histograma del canal L (Intensidad)
- Unir canales y convertir formato a escala de grises
- Aplicar filtro bilateral
- Escalar valores, uint8 -> float64

Para elegir el pulso adecuado se utilizó como criterio la entropía de Shannon para seleccionar aquella donde la pérdida de información era menor:

1. *Segunda parte: modelo ICM*
2. Para todas las muestras:
 - i. Definir $z = []$ (lista vacía)
 - ii. Inicializar matriz de ceros para F, Y, T
 - iii. Definir f, g y h
 - iv. Iterar n veces:
 - v. Obtener salida de la red:
 - ✓ $F = (f * F) + S + (W * Y)$
 - ✓ if $(F > T)$, $Y = 1$
 - ✓ $T = (g * T) + (h * Y)$
 - ✓ $W(Y_{n-1}) = [(F_2 * (W * A)) + (F_1 * A)] < 0.5$
 - ✓ $F_1 = \text{valor } 0 \text{ en } Y_{n-1}; F_2 = \text{valor } 1 \text{ en } Y_{n-1}; A = (Y_{n-1}) + [F_1 * W] > 0.5;$
3. Añadir Y a z
4. Para todas las muestras en z:
 - i. $p_1 = \text{píxeles con valor } 1; p_0 = \text{píxeles con valor } 0;$
 - ii. Calcular entropía:
 - iii. $H(s) = -p_1 \ln p_1 - p_0 \ln p_0$
5. Ordenar H(s) de forma descendente
6. Elegir pulso con máxima entropía

La segmentación se llevó a cabo mediante el etiquetado de componentes conectados para poder agrupar los píxeles que pertenecían a cada región. Se

aislaron tomando como métrica el área y se descartaron aquellas que no eran de interés como los guiones medios para después recortarlas. A continuación, se muestra parte del pseudocódigo del algoritmo implementado y el resultado de aplicarlo en la figura 6.

- *Tercera parte: Componentes conexas (4-vecinos)*
- Recorrer la imagen $Y(x,y)$
 - Asignar etiquetas iniciales (e)
 - Si vecindad $Y(x,y) = 0$ entonces $e = 1$
Incrementar e
 - Si algún vecino $Y(x,y)$ está asignado, copiar valor a píxel actual
 - Si existen varias etiquetas, asignar solo una.
 - Las etiquetas no usadas son colisiones.
- Resolver colisiones
- Reetiquetar con los píxeles con los valores mínimos



Figura 6 Obtención de caracteres.

Transformaciones geométricas

Las transformaciones geométricas que a continuación se describen se aplicaron con el fin de estandarizar los datos en el sentido de que las letras tuvieran una misma orientación (un solo eje de referencia) para que la variación entre imágenes no disminuyera el número de aciertos durante la segmentación.

Para rotar la imagen, cada píxel $f(x,y)$ se giró θ grados considerando un punto de origen y calculando la nueva posición $g(x',y')$ con base en la ecuación 6:

- *Cuarta parte: Rotación*
- *Importar imagen I*
- *Definir ángulo θ y convertirlo a radianes*
- *Obtener el seno de θ*

- *Obtener el coseno de θ*
- *Inicializar l' con las dimensiones de l*
 - l filas h , columnas w , canales c
 - $h = |h \cdot \cos \theta| + |w \cdot \sin \theta|$
 - $w = |w \cdot \cos \theta| + |h \cdot \sin \theta|$
- *Obtener el centro de l y l' $\{cx, cy, cx2, cy2\}$*
- *Para $i=1:h$ & $j=1:w$*
 - Ubicar las coordenadas del píxel con respecto al centro de l
 - $x = w-1-j-cx$; $y = h-1-i-cy$
 - Ubicar las coordenadas del píxel con respecto a l'
 - $x2 = x \cdot \cos \theta + y \cdot \sin \theta$
 - $y2 = -x \cdot \sin \theta + y \cdot \cos \theta$
 - Ajustar el centro de l'
 - $x2 = cx2 - x2$; $y2 = cy2 - y2$
 - Mover los píxeles a su nueva posición
 - $l'[y2, x2, :] = l[i, j, :]$

$$\begin{aligned} x' &= x \cos \theta - y \sin \theta \\ y' &= x \sin \theta + y \cos \theta \end{aligned} \quad (6)$$

El pseudocódigo de la *cuarta parte* se añadió antes de equalizar el canal L donde l' representa la imagen resultante, l la imagen original e i y j las filas y columnas. En adición se buscó aplicar una sola matriz de correspondencia entre píxeles para todas las imágenes conocida como homografía para corregir la distorsión de la perspectiva. Para encontrar los parámetros de este tipo de transformación proyectiva se necesitan conocer las coordenadas de al menos cuatro puntos homólogos en cada imagen [Alegre, 2016]. La ecuación 7 describe este tipo de transformación aplicada a dos planos.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = M \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{00} & h_{01} & h_{02} \\ h_{10} & h_{11} & h_{12} \\ h_{20} & h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (7)$$

En la figura 7 se muestra el resultado de aplicar la transformación geométrica y proyectiva a cuatro imágenes.



Figura 7 Resultado de aplicar las transformaciones geométricas a diferentes imágenes.

3. Resultados

El método de segmentación se evaluó en las 730 imágenes del conjunto de datos creado donde la placa era completamente visible y la cadena de caracteres estaba conformada por tres letras y cuatro números.

La tabla 1 muestra la comparación entre los algoritmos que conforman la metodología propuesta, paso por paso para observar el aumento gradual en el número de aciertos (segmentación adecuada).

Tabla 1 Desempeño entre algoritmos con el mismo banco de imágenes.

Algoritmo	Segmentación adecuada	Tiempo de ejecución (min)
1. ICM	240	32
2. ICM + homografía	390	23
3. ICM + rotación + homografía	425	28
4. ICM + caracteres	92	18
5. ICM + rotación + homografía + caracteres	160	18

Adicionalmente se incluye el tiempo de cómputo de cada script ejecutado en lenguaje de programación de Python en el entorno de desarrollo Spyder en Windows 10 con un procesador Intel Core i7 y 8GB de RAM. Las constantes utilizadas para el modelo de intersección cortical fueron determinadas empíricamente considerando el comportamiento cíclico de la red entre iteraciones, siendo 12 el valor más adecuado para los n pulsos y $f=0.2$, $g=0.01$ y $h=70$ los que denotaban mejores resultados.

Como lo demuestra la figura 8, el proceso se facilita tras haber estandarizado las imágenes, por lo que de esto último se infiere que la alternativa presentada es viable al colocar la cámara en una posición fija.



Figura 8 Resultado de dos experimentos utilizando homografía y la red ICM.

En los artículos revisados, la fase de umbralización se complica al tener que encontrar un valor adecuado para todo un conjunto de datos por basarse mayormente en propiedades estadísticas del histograma o clústeres [Mitchell, 2010]. Entre los métodos más utilizados se encuentran la determinación de un umbral global con promedio, Otsu y k-means. La comparación entre estos algoritmos se presenta en la tabla 2, donde también se incluye una pequeña descripción y las observaciones al aplicarlos en todas las muestras.

Tabla 2 Comparación entre métodos de segmentación.

Método	Descripción	No. de aciertos	Observaciones
Umbral global con promedio	Si la intensidad del píxel es mayor al promedio, el píxel de destino se establece al máximo valor definido (255 o 1), en caso contrario, cero.	324	Los objetos deben tener un alto contraste y las imágenes deben tener buena resolución.
Otsu	El umbral óptimo se logra cuando la varianza entre clases genera un valor mínimo.	412	Se necesitan más características para descartar regiones de interés.
Clustering means k-	Divide los píxeles de la imagen en grupos con características similares, para ello se consideró la distancia entre muestras.	579	El tiempo de procesamiento es mayor, siendo alrededor de un minuto por imagen.

En la figura 9 se presenta el histograma de la proyección horizontal (suma de los píxeles con valor igual a uno por columna) de una imagen segmentada utilizando el

método del promedio y de Otsu, logrando comprobarse que la distribución de las intensidades es muy similar lo que dificulta el análisis de la imagen.

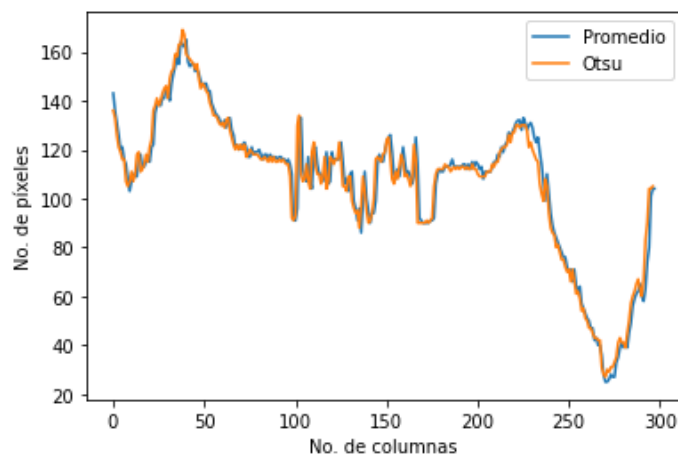


Figura 9 Proyección horizontal utilizando el promedio y el método de Otsu.

4. Discusión

Los resultados obtenidos muestran que el método propuesto es una alternativa viable para escenas donde las condiciones varían mucho entre imágenes por el diseño de las placas, el deterioro de la pintura que define los caracteres o la carencia de iluminación evitando las que son completamente oscuras siendo estos factores los de menor influencia en el desempeño final del algoritmo. No obstante, es primordial definir una posición adecuada para no colocar la cámara a más de dos metros distancia para no disminuir demasiado la calidad de las imágenes a procesar o elegir alguna con mayor resolución para evitar que el movimiento las distorsione. Cada técnica involucrada en la metodología fue determinante en la etapa de resultados, el cálculo de la homografía debido a que se pudo eliminar parte del fondo de la imagen, el modo de color LAB para reducir la contribución de la iluminación, el filtro bilateral para suavizar la imagen sin perder los bordes y la red pulso acoplada para lograr separar las letras y los números. La tabla 1 muestra la evolución del algoritmo resaltando los números 3 y 5, siendo el primero puesto a prueba para la detección de los caracteres y el segundo para la segmentación. El bajo número de aciertos reportados está relacionado en gran medida a que las placas del banco de imágenes corresponden a diferentes estados de la República Mexicana (Puebla,

Ciudad de México y Tlaxcala principalmente) y la distancia de captura por tratarse de grabaciones. También se incluye la tabla 2 que resume la comparación entre algoritmos de umbralización convencionales.

El trabajo futuro pretende unificar la propuesta presentada en este artículo y el desarrollo de una red neuronal para el reconocimiento de los caracteres y posteriormente valorar nuevamente su desempeño como parte de un sistema final de reconocimiento de matrículas en un estacionamiento.

5. Bibliografía y Referencias

- [1] Alegre, E., Pajares, G., & De la Escalera, A. (2016). *Conceptos y métodos en visión por computador*. España: Grupo de Visión del Comité Español de Automática (CEA).
- [2] Cheng, D., Zhao, W., Tang, X., & Liu, J. (2008). Image segmentation based on pulse coupled neural network. *11th Joint International Conference on Information Sciences*, pp. 323-331.
- [3] Durand, F., & Dorsey, J. (2002). Fast bilateral filtering for the display of high-dynamic-range images. *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, pp. 257-266.
- [4] Henry, C., Ahn, S. Y., & Lee, S. W. (2020). Multinational license plate recognition using generalized character sequence detection.
- [5] Hongping, H., & Yanping, B. (2011). A kind of license plate location based on mathematical morphology and edge detection. *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, pp. 2291-2294.
- [6] Jalil, N. A., Basari, A. S. H., Salam, S., Ibrahim, N. K., & Norasikin, M. A. (2015). The utilization of template matching method for license plate recognition: A case study in Malaysia. *Advanced Computer and Communication Engineering Technology*, pp. 1081-1090.
- [7] Kulkarni, P., Khandebharad, A., Khope, D., & Chavan, P. U. (2012). License plate recognition: a review. *2012 Fourth International Conference on Advanced Computing (ICoAC)*, pp. 1-8.

- [8] Kakani, B. V., Gandhi, D., & Jani, S. (2017). Improved OCR based automatic vehicle number plate recognition using features trained neural network. 2017 8th international conference on computing, communication and networking technologies (ICCCNT), pp. 1-6.
- [9] Li, H., & Bai, Z. (2008). A new PCNN-based method for segmentation of SAR images. 2008 10th International Conference on Control, Automation, Robotics and Vision, pp. 1635-1639.
- [10] Lin, C. H., Lin, Y. S., & Liu, W. C. (2018). An efficient license plate recognition system using convolution neural networks. 2018 IEEE International Conference on Applied System Invention (ICASI), pp. 224-227.
- [11] Ma, Y., Zhan, K., & Wang, Z. (2010). Applications of pulse-coupled neural networks. China: Higher Education Press, pp. 1-199.
- [12] Mitchell, H. B. (2010). Image fusion: theories, techniques and applications. Springer Science & Business Media.
- [13] Ortiz Rangel, E., Mejía-Lavalle, M., & Sossa, H. (2017). Filtrado de ruido Gaussiano mediante redes neuronales pulso-acopladas. *Computación y Sistemas*, pp. 381-395.
- [14] Wang, W., Yang, J., Chen, M., & Wang, P. (2019). A light CNN for end-to-end car license plates detection and recognition. *IEEE Access*, pp. 173875-173883.
- [15] Zheng, L., & He, X. (2011). Character segmentation for license plate recognition by K-means algorithm. *International Conference on Image Analysis and Processing*. Springer, pp. 444-453.