

Localización y reconocimiento de matrículas de automóviles utilizando redes neuronales

Elizabeth Xicotencatl-Flores, Aldrin Barreto-Flores,
Salvador E. Ayala-Raggi, Verónica E. Bautista-López

Benemérita Universidad Autónoma de Puebla,
Facultad de Ciencias de la Electrónica,
México

elizabeth.xicotencatl@alumno.buap.mx, {aldrin.barreto,
salvador.raggi, veronica.bautistalo}@correo.buap.mx

Resumen. Este artículo propone un método de reconocimiento de placas vehiculares usando el modelo de intersección cortical con conexiones nulas (ICMNI) para la localización de los caracteres y un perceptrón multicapa (MLP) de una sola capa oculta para su clasificación por separado. En contraste a otros trabajos reportados, este algoritmo excluye el proceso de localización de la placa y el preprocesamiento de la imagen permitiendo reducir el tiempo de procesamiento, además de aumentar la tasa de reconocimiento en imágenes donde los niveles de intensidad no son homogéneos. Para el entrenamiento se creó un dataset con 6,600 caracteres considerando únicamente 10 números y 23 letras. Se llevaron a cabo pruebas del algoritmo propuesto en un conjunto de 660 imágenes traseras y frontales de automóviles obtenidas de diversos sitios web compatibles con el formato asignado al estado de Puebla generando como resultado una tasa de reconocimiento del 96.2%.

Palabras clave: Mexican license plate recognition, intersecting cortical model, pulse coupled neural network, multilayer perceptron.

Localization and Recognition of Car License Plates Using Neural Networks

Abstract. This article proposes a vehicle license plate recognition method using the null-connected cortical intersection model (ICMNI) for character localization and a single hidden layer multilayer perceptron (MLP) for their separate classification. In contrast to other reported works, this algorithm excludes the plaque localization process and the image preprocessing, allowing to reduce the processing time, as well as increasing the recognition rate in images where the intensity levels are not homogeneous. For training, a dataset with 6,600 characters was created considering only 10 numbers and 23 letters. Tests of the proposed algorithm were carried out on a set of 660 rear and front images of automobiles obtained from various websites compatible with the format assigned to the state of Puebla, generating as a result a recognition rate of 96.2%.

Keywords: Mexican license plate recognition, intersecting cortical model, pulse coupled neural network, multilayer perceptron.

1. Introducción

Los Sistemas de reconocimiento de matrículas han sido estudiados a nivel mundial a través de múltiples aplicaciones (que van desde el cobro automatizado de estacionamientos [1] hasta sistemas de vigilancia [2]), cada uno con sus propias restricciones por las necesidades a cubrir en su diseño contemplando factores como: cambios de iluminación, resolución y posición de la cámara, oclusión, deterioro de las matrículas y el más importante, la tipografía de cada país. Este es el motivo de que cada metodología sea independiente y se limite a ciertas condiciones y fondos para evitar un bajo desempeño en imágenes con altas variaciones.

Actualmente es posible identificar dos enfoques en este tipo de sistemas, los de una sola etapa y los multietapa [3]. Los primeros se basan fundamentalmente en algoritmos de aprendizaje profundo (también conocido como *deep learning*) conformados por redes neuronales artificiales que emplean la detección de objetos en extensas bases de datos donde se requiere ciertos niveles de abstracción para localizar y clasificar las imágenes.

Entre las redes pre-entrenadas más populares se encuentran las convolucionales como *Single-Shot Detector* (SSD), *You Only Look Once* (YOLO), *Region Based Convolutional Neural Networks* (R-CNN) y sus versiones optimizadas [4-7]. En el segundo enfoque se crean tres subprocesos: localización de la placa, segmentación y reconocimiento de los caracteres, los cuales involucran diferentes técnicas de procesamiento de imágenes y clasificadores tales como redes probabilísticas [8], correlación (*template matching*) [9], *K-Means* [10], y *k-nearest neighbors* (k-NN) [11]. Por lo general, la elección de cada técnica depende de los recursos disponibles y de la tasa final de reconocimiento que cada subproceso añade a la etapa anterior.

En este trabajo se presenta un algoritmo para el reconocimiento de matrículas de automóviles basado en el modelo de intersección cortical con conexiones nulas para el proceso de segmentación cuya efectividad recae en la condición de activación continua de cada neurona para la asignación de los parámetros y un perceptrón multicapa de una sola capa oculta para la clasificación de los caracteres.

Este artículo está organizado de la siguiente manera: en la sección 2 se presentan los fundamentos de las técnicas empleadas, en la sección 3 se muestra el método propuesto a detalle para el reconocimiento de placas vehiculares, en la sección 4 se muestran los resultados experimentales y su comparación con otras técnicas actuales y finalmente las conclusiones se mencionan en la sección 5.

2. Marco teórico

2.1. ICM con conexiones nulas

Una red pulso acoplada (PCNN, *Pulse Coupled Neural Network*) es un tipo de red neuronal que clasifica los píxeles por sus niveles de intensidad generando una secuencia de imágenes binarias a partir de una sola imagen de entrada sin necesidad de una etapa de entrenamiento, razón por la cual ha sido aplicada a la segmentación, codificación, fusión y extracción de características de imágenes [12-15], siendo una técnica poco explorada para la segmentación de matrículas vehiculares.

El modelo de intersección cortical (ICM, *Intersecting Cortical Model*) es una variante de una PCNN diseñada para reducir el costo computacional eliminando algunos parámetros de ajuste del modelo original. En cada iteración la intensidad de los píxeles que conforman la imagen a procesar es introducida a la red a través de una señal conocida como *Feeding* F , la cual se compara con un umbral dinámico E para generar los pulsos de salida Y representados por n imágenes binarias:

$$F_{ij}(n) = fF_{ij}(n-1) + S_{ij} + \sum_{kl} W_{ijkl} Y_{kl}(n-1), \quad (1)$$

$$E_{ij}(n) = gE_{ij}(n-1) + hY_{ij}(n-1), \quad (2)$$

$$Y_{ij}(n) = \begin{cases} 1, & F_{ij}(n) > E_{ij}(n), \\ 0, & \text{en caso contrario.} \end{cases} \quad (3)$$

El comportamiento anteriormente definido puede ser descrito por las ecuaciones (1)-(3), donde los subíndices (i,j) representan la posición de cada neurona y (k,l) la de sus neuronas vecinas, W_{ijkl} es una función asociada a la conexión entre neuronas, f el factor de decaimiento de la señal *Feeding*, g la constante de decaimiento para el umbral dinámico y h un potenciador de activación. Pese a esta modificación, en la práctica los parámetros involucrados en este modelo computacional continúan limitando su rendimiento ocasionando que surjan distintas estrategias de optimización [16-18], es ahí donde surge el término “conexiones nulas” [19] que más allá de realizar una elección automática de los parámetros de ajuste lo que se busca es reducir al mínimo el número de variables eliminando la influencia de los vecinos de cada neurona, simplificando aún más el modelo original de una red pulso acoplada a través de las ecuaciones (4)-(6):

$$F(n) = f F(n-1) + S, \quad (4)$$

$$E(n) = g E(n-1) + h Y(n-1), \quad (5)$$

$$Y(n) = \begin{cases} 1, & F(n) > E(n), \\ 0, & \text{en caso contrario.} \end{cases} \quad (6)$$

Entre los aportes más destacados se encuentran los de Jin, X. et al. [20], quienes demostraron que la información de este tipo de red emite puede ser analizada por la condición de activación continua de cada neurona, lo que permite restringir los parámetros al satisfacer la ecuación (7):

$$S > h \frac{1-f}{1-g}, \quad (7)$$

donde S representa la imagen de entrada y f, g y h los valores de ajuste.

2.2. Perceptrón multicapa

Un perceptrón multicapa (MLP, *Multilayer perceptron*) es un tipo de red neuronal artificial que se caracteriza por tener en su configuración al menos una capa de entrada, una capa oculta y una capa de salida conformada por dos o más neuronas conectadas entre sí.

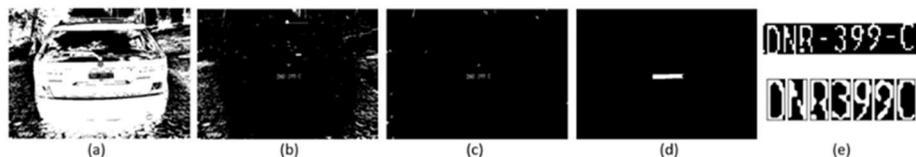


Fig. 1. Proceso de filtrado de objetos en una (a) imagen binaria (b) por área, (c) dimensiones y relación de aspecto, (d) forma geométrica y longitud y (e) segmentación utilizando la red ICMNI para definir los caracteres.

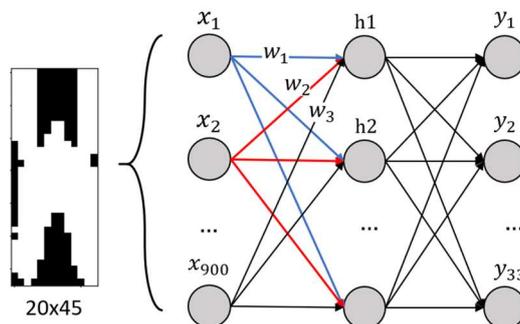


Fig. 2. Arquitectura inicial del perceptrón multicapa.

Durante el proceso conocido como *feedforward* o “propagación hacia adelante” cada neurona recibe como entrada las salidas de todas las neuronas de la capa anterior para poder obtener una salida (en este proceso se realiza una sumatoria ponderada de todos los datos y se aplica una función de activación generalmente del tipo sigmoide [21]) que dependiendo del tipo de análisis que se realice, se podrá interpretar para resolver problemas de regresión o de clasificación binaria/multiclase.

El entrenamiento se puede llevar a cabo mediante la retropropagación del error (*backpropagation*) entre capas a través del descenso del gradiente [22] para que los pesos se actualicen en función del aporte en que han contribuido a ese error y como función de coste global se suele utilizar el error cuadrático medio, que representa la suma de los errores resultantes de la diferencia entre la salida ideal y la salida de la red.

Generalmente no se alcanza un error absolutamente igual a cero, pero se puede definir un límite de error cercano a ese valor o un número de iteraciones máximas también conocidas como épocas.

3. Método propuesto

3.1. Localización y segmentación de los caracteres

El algoritmo que realiza el proceso de localización está basado en la red ICMNI para binarizar la imagen de entrada de acuerdo con lo siguiente:

1. Estandarizar las dimensiones de todas las imágenes (S) y convertirlas a escala de grises para su posterior manipulación.



Fig. 3. Ejemplos del conjunto de entrenamiento.

2. Normalizar entre 0 y 1 los niveles de intensidad de S. Esto se logra dividiendo el valor de cada píxel entre el máximo nivel de intensidad dentro de la imagen.
3. Inicializar los valores de Y, F y E en cero con las dimensiones de S.
4. Definir los parámetros de la red condicionados por la ecuación (7) y sumar la constante obtenida a la imagen normalizada.
5. De acuerdo con el comportamiento de la red establecer un número máximo de iteraciones.
6. Almacenar los pulsos de salida para analizarlos y verificar que todas las neuronas se activen al menos una vez (estadística de disparo).
7. Al término de las N iteraciones aplicar el operador unión a los pulsos obtenidos para generar una sola imagen de salida.

Extracción de las regiones de interés. Para ubicar los caracteres dentro de la imagen se realizó una serie de filtrado de objetos (véase la Fig.1) utilizando como referencia características geométricas [23]. Primero eliminando el área más grande correspondiente al vehículo empleando la técnica de componentes conectados y después limitando las dimensiones y la relación de aspecto de los posibles candidatos con *bounding boxing*.

Posteriormente a los objetos resultantes se les aplicó el operador dilatación para extraer los caracteres que pudieran haber sido descartados en la etapa anterior utilizando como estructura de referencia una matriz de 1×25 .

Tras obtener la región de la Fig.1(d), se filtraron nuevamente los objetos de la zona tomando como referencia la imagen binaria original para eliminar los guiones entre caracteres y por fin aislar cada dígito y letra utilizando una vez más el operador dilatación con una nueva estructura de referencia de 50×1 . Por último, se empleó la red ICMNI en los caracteres por separado para facilitar el proceso de reconocimiento.

3.2. Reconocimiento de la matrícula

Para la elección de la arquitectura del MLP (Fig.2) se realizaron diversas pruebas del algoritmo desarrollado en el lenguaje de programación de Python considerando una sola capa oculta, 900 neuronas de entrada correspondientes al total de píxeles de la imagen binaria y 33 neuronas de salida que representan el número de clases posibles.

Para ingresar las muestras a la red, primero se categorizó cada imagen en carpetas según el número o letra que representaba (0,1,2,3, etc.), después se convirtieron a escala de grises normalizando los valores entre 0 y 1 y, por último, se manipularon como vector para añadirlos a una matriz que definiría el conjunto de entrenamiento y validación. El entrenamiento se llevó a cabo por los siguientes pasos:

1. Revolver aleatoriamente el conjunto de muestras y destinar 1/5 parte al set de validación.
2. Asociar todos los vectores de entrada con su salida deseada $\{x_i, y_i\}$ utilizando la codificación *one hot*.

3. Asignar valores aleatorios pequeños a los pesos $\{w\}$.
4. Definir una tasa de aprendizaje $\{\eta\}$ y un momento $\{\alpha\}$.
5. Para cada muestra de entrenamiento obtener la salida de la red:

$$ai = \sum_{i=1}^n w_n x_n ; y_{real} = h(ai). \quad (8)$$

6. Evaluar el error cuadrático medio:

$$ECM = \sum \frac{1}{2} * (y_{ideal} - y_{real})^2. \quad (9)$$

7. Calcular el incremento parcial de los pesos utilizando el descenso del gradiente y la regla de la cadena:

$$\nabla E = \frac{\partial E}{\partial wo} = \frac{\partial E}{\partial y} \cdot \frac{\partial y}{\partial ao} \cdot \frac{\partial ao}{\partial wo}, \quad (10)$$

$$\frac{\partial E}{\partial wh} = \frac{\partial E}{\partial h} \cdot \frac{\partial h}{\partial ai} \cdot \frac{\partial ai}{\partial wh}. \quad (11)$$

8. Actualizar los pesos de la capa de salida:

$$wo = wh + \alpha(wo - wo) - \eta \cdot \frac{\partial E}{\partial wo}. \quad (12)$$

Actualizar los pesos de la capa oculta:

$$wh = wh + \alpha(wh - wh) - \eta \cdot \frac{\partial E}{\partial wh}. \quad (13)$$

9. Obtener el mínimo coste global y definir un número máximo de épocas.

Al finalizar este proceso, para asignar etiquetas a nuevas muestras solo resta obtener la salida de la red con la ecuación (8) utilizando los pesos actualizados.

4. Resultados

4.1. Dataset

En virtud de la inexistencia de una base de datos con características mexicanas disponible al público, se creó un dataset a partir de imágenes frontales y traseras de automóviles capturadas en video a través de un teléfono celular con una resolución de 13 megapíxeles bajo diversas condiciones de iluminación por la hora del día a la que fueron tomadas.

En total, se logró reunir 730 imágenes las cuales solo se utilizaron para segmentar los caracteres y añadirlos al conjunto de entrenamiento del MLP. En la Fig.3 se muestran algunos ejemplos de los caracteres utilizados para el entrenamiento. El dataset

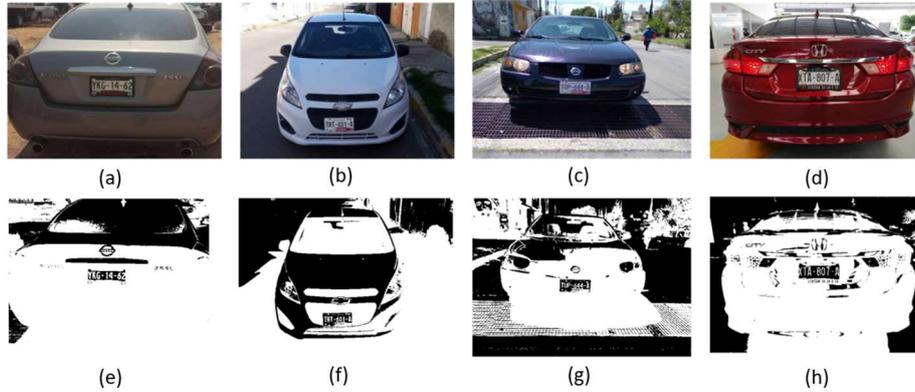


Fig. 4. Resultados del proceso de segmentación utilizando el método propuesto.

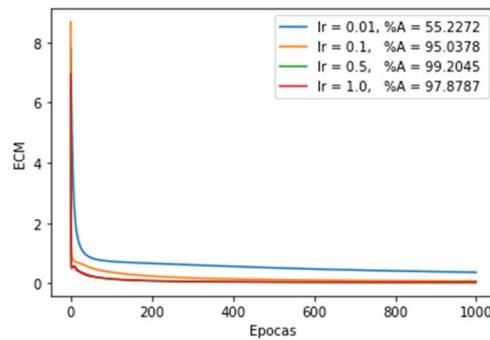


Fig. 5. Evaluación del error de aprendizaje en función de las épocas.

incluye los dígitos del 0 al 9 y las letras mayúsculas del alfabeto latino de la A a la Z excluyendo la I, Ñ, O y la Q.

Para evaluar la metodología propuesta, se creó otro dataset con características similares al primero para evitar mostrarle al algoritmo ejemplos que sirvieron para su aprendizaje. Esta nueva base de datos está conformada por 660 imágenes provenientes de diversos sitios web y al igual que el anterior el formato es compatible al asignado al estado de Puebla.

Aumento de datos. Debido al balanceo de las clases, del conjunto de datos destinado solo se pudieron seleccionar 100 ejemplos por etiqueta, lo cual afectó el entrenamiento del MLP disminuyendo el porcentaje de aciertos. Por esta razón se resolvió mejorar la robustez del algoritmo empleando la técnica de dilatación utilizando la estructura de referencia de la ecuación (14) sobre los 3,300 ejemplos para duplicar la cantidad de muestras:

$$\begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}. \quad (14)$$

Tabla 1. Resumen de las pruebas de entrenamiento al cambiar los parámetros.

Tasa de aprendizaje	Neuronas ocultas	Momento	Épocas	Predicción correcta (%)		ECM	Tiempo
				Entrenamiento	Validación		
22 imágenes x 33 clases, total: 726, validación: 145							
0,3	50	0,5	10,000	100,0	77,931	0,0011	3 min
0,3	50	0,5	30000	100,0	83,448	0,0000	8 min
0,3	50	0,5	50,000	100,0	79,310	0,0000	16 min
100 imágenes x 33 clases, total: 3300, validación: 660							
0,1	180	0,2	30,000	99,8106	95,4545	0.0092	57 min
200 imágenes x 33 clases, total: 6600, validación: 1320							
0,5	250	0,5	30,000	99,981	98,0303	0.0013	2h

Resultados de la localización y segmentación

A través de una serie de pruebas se lograron determinar los parámetros que satisfacen la condición de activación continua (ecuación 15), siendo $f = 0.85$, $g = 0.5$ y $h = 15$ los valores que a su vez permiten visualizar en algún pulso la matrícula durante el proceso de segmentación:

$$S > h \frac{1-f}{1-g} = 15 \frac{0.15}{0.5} = 4.5. \quad (15)$$

Sumar 4.5 a la imagen normalizada además de activar todas las neuronas secuencialmente, también permite determinar el número máximo de iteraciones, dado que a partir de cierta iteración la red comienza a repetir los patrones de salida.

Como se observa en la Fig.4, la implementación de la metodología muestra ser efectiva en imágenes que son variantes entre sí respecto a los niveles de intensidad que la conforman, ya sea por la textura o el ruido implícito. Durante el entrenamiento del MLP se probaron diferentes parámetros respetando la arquitectura que ya había sido determinada en la sección 3.2.

De la Tabla 1 se deduce que el aumento de épocas aumenta considerablemente el porcentaje de aciertos y el número de neuronas está ligado a la cantidad de datos que estén disponibles. El tiempo reportado corresponde a la ejecución del script implementado en el lenguaje de Python utilizando el entorno de desarrollo de Spyder con un procesador Intel i7 y 8GB de memoria RAM. La tasa de aprendizaje se fijó considerando el comportamiento de la Fig.5 evaluando el decremento del error cuadrático medio y el aumento del porcentaje de aciertos para descartar un subajuste o sobreentrenamiento del MLP.

La métrica utilizada para evaluar la metodología en el conjunto de 660 imágenes se obtuvo dividiendo el número de matrículas que se clasificaron correctamente por el total de muestras de prueba multiplicado por 100, dando como resultado un 96.2% de precisión. Solo se tomaron como válidas las placas donde todos los caracteres habían

Tabla 2. Comparación entre técnicas existentes de reconocimiento de placas y el propuesto.

Técnicas principales y referencias	YOLOv2 + SVM [5]	RPN + R-CNN [6]	R-CNN + CTC[7]	K-means + CNN [10]	ICMNI +MLP
Año	2018	2018	2020	2020	2021
Entrenamiento	8,776	10,873	2,700k	43,615	6,600
Validación	5,850	472	300	16,185	660
Variación en distancia	✓	x	✓	x	✓
Perspectiva frontal	✓	x	x	✓	✓
Loc. placa (%)	94.23	89.12%	93.7	94.8	100
Rec. caracteres (%)	99.2	99.1	98.9	98.1	96.2
Plataforma & procesador	NVIDIA Ge Force 12GB DDR5X DRAM	--	Intel Xeon E5-2680v4 NVIDIA Tesla k80	PC i5 16 GB RAM Python, Tensorflow PyTesseract	PC i7, 8GB RAM, Python, Spyder

sido correctamente etiquetados. En este punto cabe denotar que la red ICMNI facilitó en gran medida la localización de los caracteres alfanuméricos, es por ello que se logró alcanzar un 100% en la etapa de segmentación.

La Tabla 2 muestra una comparación de rendimiento entre métodos de reconocimiento de sistemas vehiculares reportados por diferentes autores y el que se propone en este artículo, además de incluir las características de los datasets utilizados. Como se observa, las técnicas de aprendizaje profundo necesitan grandes cantidades de datos durante el entrenamiento para alcanzar un elevado porcentaje de precisión al clasificar los caracteres, siendo altamente recomendables por ser relativamente insensibles a valores con ruido.

Sin embargo, aún existe apertura a otro tipo de técnicas dado que en la mayoría de los casos se consideran dependientes del hardware. El algoritmo propuesto logra estar al nivel de una red convolucional en imágenes que presentan variación en distancia sin incidir en placas sumamente borrosas y considerando únicamente una perspectiva frontal, por lo que puede utilizarse en aplicaciones que no son tan demandantes como lo son las identificaciones vehiculares múltiples [7].

En comparación a los otros autores, se logró alcanzar un 100% en la localización de la placa al aprovechar que la red ICMNI activa las neuronas según el umbral óptimo durante el proceso de binarización. Esta propiedad permitió obtener una segmentación más robusta en todo el dataset utilizando los mismos parámetros para todas las imágenes sin necesidad de agregar algún filtro (lineal o no lineal) para reducir el impacto del ruido o un espacio de color para aumentar el contraste, procesos que forman parte del preprocesamiento que se suele añadir en la etapa de localización de la placa.

También se alcanzó un porcentaje de reconocimiento de caracteres cercano a las otras contribuciones, por lo que se infiere que este valor puede mejorar aumentando el número de ejemplos para entrenar la red neuronal.

5. Conclusiones

De acuerdo con los resultados obtenidos, la metodología propuesta presenta un buen desempeño en imágenes donde los niveles de intensidad no son homogéneos, siendo útil en casos donde la localización de los caracteres se ve afectado por cierto tipo de

sombras o por cambios bruscos de iluminación, sin embargo, es susceptible a la distancia de captura y la inclinación de la placa.

Este enfoque basado en la localización de la matrícula y no de la placa sirve como alternativa en casos donde no existe un color primario predominante en la zona de la placa que permita aislarla con facilidad o cuando el valor del umbral es variante en todas las imágenes. El proceso de reconocimiento no se ve limitado a un número determinado de caracteres por lo que podría utilizarse en imágenes que incluyan otro tipo de vehículos, como autobuses o motocicletas fijando un rango de distancia de captura y un grado de inclinación adecuado para este sistema.

Para trabajos futuros se considera ajustar la matrícula a un solo perfil para incluir en el dataset de prueba imágenes rotadas o con perspectiva lateral para aumentar el desempeño del algoritmo en otro tipo de escenarios, reduciendo a su vez las restricciones.

Referencias

1. Yimyam, W., Ketcham, M.: The automated parking fee calculation using license plate recognition system. In: Proceedings of International Conference on Digital Arts, Media and Technology (ICDAMT), IEEE pp. 325–329 (2017)
2. Zhu, L., Wang, S., Li, CH., Yanget, Z.: License plate recognition in urban road based on vehicle tracking and result integration. *Journal of Intelligent Systems*. vol. 29, no. 1, pp. 1587–1597 (2020)
3. Shashirangana, J., Padmasiri, H., Meedeniya, D., Perera, C.: Automated license plate recognition: A survey on methods and techniques. *IEEE Access*, vol. 9, pp. 11203–11225 (2020)
4. Weihong, W., Jiaoyang, T.: Research on license plate recognition algorithms based on deep learning in complex environment. *IEEE Access*, vol. 8, pp. 91661–91675 (2020)
5. Lin, C. H., Lin Y. S., Liu, W. C.: An efficient license plate recognition system using convolution neural networks. In: Proceedings of IEEE International Conference on Applied System Invention (ICASI), pp. 224–227 (2018) doi: 10.1109/ICASI.2018.8394573
6. Yang, Z., Du, F. L., Xia, Y., Zheng, C. H., Zhang, J.: Automatic license plate recognition based on faster R-CNN algorithm. In: Proceedings of International Conference on Intelligent Computing, Springer, Cham, pp. 319–326 (2018) doi: 10.1007/978-3-319-95957-3_35
7. He, M. X., Hao, P.: Robust automatic recognition of Chinese license plates in natural scenes. *IEEE Access*, vol. 8., pp. 173804–173814 (2020)
8. Anagnostopoulos, C. N. E., Anagnostopoulos, E. I., Loumos, V., Kayafas, E.: A license plate-recognition algorithm for intelligent transportation system applications. In: Proceeding of IEEE Transactions on Intelligent transportation systems. vol. 7, no. 3, pp. 377–392 (2006) doi: 10.1109/TITS.2006.880641
9. Jalil, N. A., Basari, A. S. H., Salam, S., Ibrahim, N. K., Norasikin, M. A.: The utilization of template matching method for license plate recognition: A case study in Malaysia. In: Proceedings of Advanced Computer and Communication Engineering Technology. Springer, Cham, pp. 1081–1090 (2015) doi: 10.1007/978-3-319-07674-4_100
10. Pustokhina, I. V., Pustokhin, D. A., Rodrigues, J. J. P. C., Gupta, D., Khanna, A., Seo, C., Joshi G. P.: Automatic vehicle license plate recognition using optimal K-means with convolutional neural network for intelligent transportation systems. *IEEE Access*, vol. 8, pp. 92907–92917 (2020)
11. Escalante, S., Murray, V.: Automatic recognition of peruvian car license plates. In: IEEE XXVII International Conference on Electronics, Electrical Engineering and Computing (INTERCON), IEEE, pp. 1–4 (2020)

12. Ma, Y.: Applications of pulse-coupled neural networks. Higher Education Press, Beijing, China (2010)
13. Lindblad, T., Kinsler, J. M.: Image processing using pulse-coupled neural networks. Heidelberg: Springer (2005)
14. Mohsen, H., El-Dahshan, E. S. A., Salem, A. B. M.: A machine learning technique for MRI brain images. In: Proceedings of 8th International Conference on Informatics and Systems (INFOS), IEEE, pp. BIO-161–BIO-165 (2012)
15. Mokhayeri, F., Akbarzadeh-T, M. R.: A novel facial feature extraction method based on ICM network for affective recognition. In: Proceedings of International Joint Conference on Neural Networks, IEEE, pp. 1988–1993 (2011)
16. Dharwal, D., Shanker, R., Bhattacharya, M.: Automatic parameter setting of pulse coupled neural network for image segmentation. In: International Conference on Communication and Signal Processing (ICCSP), IEEE, pp. 2157–2161 (2016)
17. Hernández, J., Gómez, W.: Automatic tuning of the pulse-coupled neural network using differential evolution for image segmentation. In: Mexican Conference on Pattern Recognition, Springer, Cham, pp. 157–166 (2016)
18. Deng, X., Ma, Y.: PCNN model analysis and its automatic parameters determination in image segmentation and edge detection. Chinese Journal of Electronics, vol. 23, no. 1, pp. 97–103 (2014)
19. Prieto, C., Rodríguez, A.: A soft image edge detection approach based on the time matrix of a PCNN. In: 2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence), IEEE, pp. 463–469 (2008)
20. Jin, X., Zhou, D., Jiang, Q., Chu, X., Yao, S., Li, K., Zhou, W.: How to analyze the neurodynamic characteristics of pulse-coupled neural networks? A Theoretical Analysis and Case Study of Intersecting Cortical Model, IEEE, Transactions on Cybernetics, vol. 52, no. 7, pp. 6354–6368 (2021)
21. Nunes, I., Hernane, D.: Artificial neural networks: A practical course. Springer (2018)
22. Isasi, P., Galván, I.: Redes de neuronas artificiales. Un enfoque práctico, Pearson Educación, Madrid, España (2004)
23. Cuevas, E., Zaldivar, D., Perez-Cisneros, M.: Procesamiento digital de imágenes con Matlab y Simulink. Alfaomega, vol. 10, no. 1, pp. 77–78 (2010)