Intelligent Template Matching: Un Nuevo Acercamiento para Reconocimiento Automático de Patentes Argentinas

Nicolás Fernando Gazcón[1,3], Carlos Iván Chesñevar[2,3], Silvia Mabel Castro[1]

[1]Laboratorio de Investigación y Desarrollo en Visualización y Computación Gráfica
[2]Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Departamento de Ciencias e Ingeniería de la Computación
Universidad Nacional del Sur
Avenida Alem 1253, Bahía Blanca, Buenos Aires, Argentina. CP 8000

[3]Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Argentina
{nfg,cic,smc}@cs.uns.edu.ar

Resumen El reconocimiento automático de patentes es un problema que comenzó a ser estudiado a principio de los años '90, y recientemente ha ganado interés a partir de las particularidades de patentes de diversos países. Se han desarrollado así acercamientos eficientes utilizando diferentes técnicas de reconocimiento de patrones. Si bien existen sistemas comerciales para reconocimiento de patentes usados en Argentina, la ausencia de criterios estandarizados de evaluación de desempeño dificulta el estudio de su real efectividad. Este trabajo presenta un acercamiento novedoso denominado Intelligent Template Matching para reconocimiento automático de patentes automotores de Argentina. Se compara la performance obtenida con otros acercamientos competitivos (como redes neuronales), y se muestran las ventajas de su utilización. Se realizaron experimentos que evidenciaron una mejor performance en exactitud de reconocimiento, sin requerir un conjunto de datos de entrenamiento. Asimismo, los resultados obtenidos son extrapolables a otros sistemas de representación de patentes diferentes a las patentes argentinas.

Palabras clave: Reconocimiento Automático de Patentes – Redes Neuronales Artificiales – Reconocimiento Optico de Caracteres – Template Matching

1. Introducción

El incremento de la población y sus necesidades naturalmente establecen la misma tendencia en el transporte. De esta manera la gran cantidad de vehículos obliga a automatizar ciertas tareas, como por ejemplo su identificación. El reconocimiento automático es de utilidad en un gran número de tareas de control tales como la verificación de accesos, niveles de tráfico y velocidades, o bien el ingreso a fronteras, por nombrar algunas. De hecho, términos empleados en las

grandes urbes como *Sistemas de Tráfico Inteligentes* o *Autopistas Inteligentes*, están fuertemente ligados a la identificación automática de vehículos, y por consiguiente, a la extracción de su número de patente. El tema sobre Reconocimiento Automático de Patentes (ANPR) no es nuevo, los primeros sistemas comerciales datan de los 80', aunque recién en los 90' se introducen aproximaciones robustas mediante técnicas de reconocimiento óptico de caracteres [1]. A lo largo de los años se han propuesto muchas estrategias que atacan las diferentes etapas de esta problemática [2]. De la misma manera, han surgido varias propuestas comerciales, incluso en Argentina [3,4,5]. Sin embargo, tal como se postula en [6], la ausencia de criterios estandarizados de evaluación de desempeño dificulta la interpretación de los valores que acusan los sistemas comerciales.

Puntualmente podemos clasificar el problema de ANPR dentro del área de Procesamiento de Imágenes y Visión por Computadoras. El objetivo será la identificación del número de patente a partir de imágenes de vehículos. Más allá de ser un problema relacionado con el hardware utilizado (por ejemplo cámaras infrarrojas), tiene una fuerte dependencia del algoritmo empleado. Desde la captura de la imagen del auto en la región en la que se encuentra la patente, hasta la identificación de la misma, se pueden considerar varias etapas:

Captura: adquisición de la imagen. Está fuertemente relacionada con el hardware empleado y las características del ambiente incluyendo calidad y tipo de cámara, iluminación, condiciones meteorológicas, etc.

Mejoramiento de imagen: manipulación de la imagen para mejorarla, por ejemplo quitándole ruido.

Identificación y extracción de la patente: búsqueda, segmentado y normalización de la patente en la imagen. Puede optimizarse si se basa en las características puntuales de patentes de determiandas regiones o países [7,8].

Segmentado de caracteres: segmentado de cada caracter individual en la patente ya extraída.

Identificación individual de cada carácter: reconocimiento de los caracteres extraídos en la segmentación mediante técnicas de Reconocimiento Óptico de Caracteres (OCR). Varios métodos se han propuesto, como por ejemplo template matching [9] o diferentes tipos de clasificadores [10], entre ellos las Redes Neuronales Artificiales (ANN).

Las complicaciones que agrega el medio (condiciones ambientales, iluminación y desgaste o suciedad de las patentes entre otras) obligan a contar con un sistema confiable y robusto. Por ello en el presente trabajo comparamos la identificación de caracteres mediante un enfoque clásico y el nuevo método propuesto: Redes Neuronales Artificiales e Intelligent Template Matching (ITM). Para esto se desarrolló un programa orientado a las patentes Argentinas que respete las etapas mencionadas, utilizando las dos aproximaciones en la implementación del OCR. En base a esto podemos abstraernos y ver la problemática del ANPR como una tarea de Reconocimiento de Caracteres.

El resto de este trabajo está estructurado de la siguiente manera. La Sección 2 resume el proceso general para la identificación de la patente en la imagen junto con las dos implementaciones de OCR. Luego, en la Sección 3, se presentan los resultados experimentales, y en la siguiente sección se discuten los mismos.

Finalmente, la Sección 5 resume las principales conclusiones que se han obtenido y las perspectivas a futuro.

2. Reconocimiento de Patentes con el Sistema -FGL-

El software propuesto para obtener el número de patente, denominado *Sistema -FGL-*[1], respeta en su estructura las fases mencionadas. Su desarrollo fue pensado y optimizado para las patentes de autos Argentinas (Fig. 1(a)). Estas patentes se caracterizan por estar compuestas de un material reflectivo y están enmarcadas con un borde externo blanco. Este marco externo presenta un rectángulo interior negro, el cual contiene seis caracteres. Estos están compuestos de tres letras, seguidos de un espacio vacío (usualmente utilizado para una letra de menor tamaño que identifica patentes replicadas) y finalmente tres dígitos. Sobre el marco blanco y en la parte superior de la patente se presenta, en color celeste, el escudo argentino a la izquierda y el nombre del país en el centro. A continuación se presenta el proceso de reocnocimiento de patente (Fig. 1(b)) describiendo cada una de las etapas involucradas.

Preprocesamiento. El mejoramiento de imagen se decidió obviar y trabajar con la captura *cruda*, de manera de reducir los tiempos de procesamiento. Esta elección no fue azarosa ya que se hicieron tests que permitieron establecer que los resultados obtenidos eran satisfactorios. No obstante dentro de esta etapa, podemos considerar el paso inicial de conversión de la imagen de entrada a escala de grises (valores de brillo entre 0 y 255).

Identificación-Extracción de Patente. La identificación de la patente en la imagen se basaba en una búsqueda de rectángulos. Para esto se realizaba inicialmente la binarización de la imagen mediante distintos umbralados de la imagen en escala de grises. Para los valores de umbral se partía desde un valor máximo (250), y en caso de no obtener los resultados esperados (detectar una patente), se disminuía el valor del umbral realizándose nuevamente el proceso. En el peor de los casos se llega a un valor mínimo de umbral, donde se concluye que no hay patente en la imagen.

Para la búsqueda del rectángulo correspondiente a una patente, se partía de la imagen binarizada y se procedía a una búsqueda de contornos. Con los contornos se realizaba una aproximación poligonal, seleccionádose aquellos que eran rectangulares y con un área mayor a un área mínima determinada (de manera de filtrar los rectángulos que fueran demasiado pequeños). Una vez detectados los rectángulos se procedía a una transformación perspectiva, quitando el sesgo de éstos.

Con la potencial patente normalizada (a tamaño 246x110), se realizaba una búsqueda de símbolos dentro de la imagen. En esta etapa también se empleaba una búsqueda de contornos, evaluando su disposición y tamaño. En caso de encontrar seis elementos dispuestos en las posiciones esperadas y con los tamaños aproximados en relación al tamaño de la imagen, se identifica el rectángulo como una patente. Recordemos que la disposición de los símbolos corresponde a la de patentes Argentinas.

Segmentado de Caracteres. En esta etapa se procede a extraer los símbolos detectados en la etapa anterior, normalizándolos a un tamaño predefinido. Un detalle

[1] Software desarrollado por el primer autor de este artículo para la materia *Procesamiento de Imágenes* (Departamento de Cs. e Ing. de la Computación, Universidad Nacional del Sur, Bahía Blanca, Argentina) en conjunto con Sebastián Farías y Nicolás Losada.

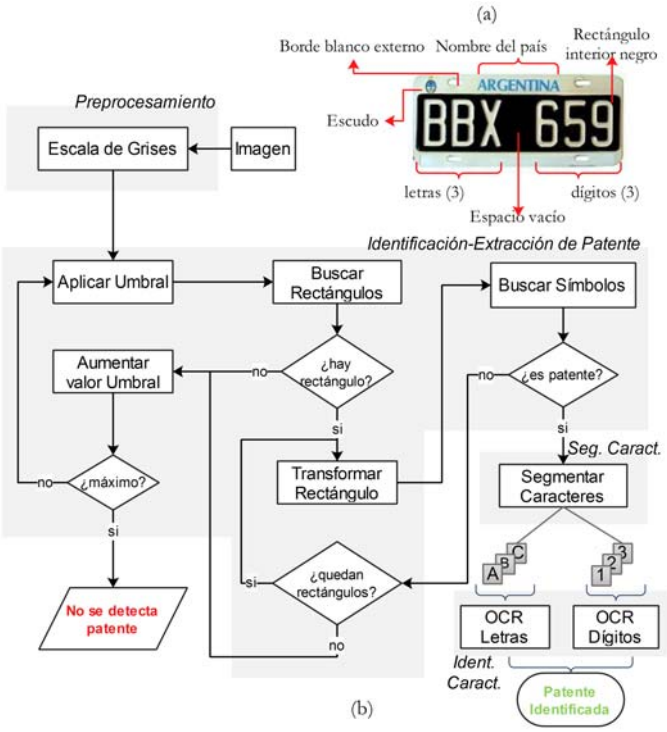


Figura 1. Composición de patente argentina (a) y del proceso de reconocimiento de patente (b).

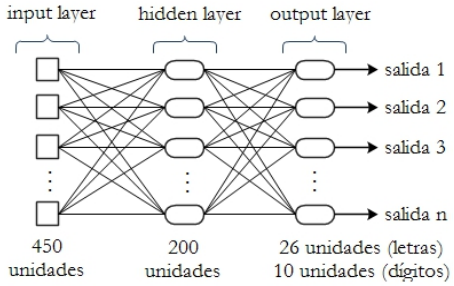
importante es que en este momento, puede asegurarse cuáles de los elementos son letras y cuáles dígitos. Como en el paso anterior se consideraban las posiciones de los símbolos, podemos identificar que los primeros tres caracteres segmentados corresponden a letras y los tres siguientes a dígitos.

Identificación de Caracteres. Este último paso corresponde al OCR, para realizar esta tarea, presentamos dos métodos diferentes. con distinta performance. Uno de ellos corresponde a la identificación del carácter mediante ANN's y el otro mediante ITM. Ambas aproximaciones coinciden en dividir en dos conjuntos los caracteres a identificar: letras y dígitos (gracias a la discriminación de símbolos mencionada en el paso anterior). A continuación se detalla cada método.

2.1. OCR Mediante ANN

En este caso, el reconocimiento de los caracteres se realizó mediante dos ANN's, correspondientes a letras y dígitos. Esta estrategia es una de las opciones clásicas porque presenta fuerte tolerancia al ruido de los datos de entrada. Esto se traduce en robustez para el sistema. Gracias a esto se pueden contemplar condiciones que disten en gran medida de los casos ideales. Esta robustez tiene su costo, dado que se requiere una etapa inicial de entrenamiento fuera de línea para las redes. Esto implica tiempo para entrenar y refinar los parámetros

utilizados, además de la necesidad de contar con una gran cantidad de ejemplos de entrenamiento. Los tiempos de procesamiento de las ANN no son malos, pero tampoco se destacan por ser las estrategias más rápidas. Otro punto a considerar será la memoria necesaria dependiendo de la cantidad de neuronas de la red. En este caso se optó por una *Layered Feed-forward Backpropagation Network*, compuesta por tres capas: entrada, una única etapa de *hidden units* y salida (Fig. 2(a)). A continuación resaltamos los detalles más importantes.



(a)

A	B	C	D	E	F	G	H	I	J	K	L
56	67	29	47	29	45	51	37	56	11	12	10

M	N	O	P	Q	R	S	T	U	V	W	X
5	9	23	8	13	8	26	31	15	9	14	16

Y	Z	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
9	19	66	44	35	57	78	106	98	37	46	75

(b)

Figura 2. *Layers* de la ANN (a) y cantidad de ejemplos de entrenamiento utilizados (b).

Neuronas de Entrada. La entrada de las redes se decidió en base a los pixels del carácter segmentado. En este caso constaba de una imagen de 15x30, dando un total de 450 neuronas de entrada.

Neuronas de Salida. Dependen de las características de la red neuronal. La red de letras poseeía 26 unidades respectivas a cada carácter, y naturalmente la de dígitos contenía 10 salidas.

Hidden Layer. Las neuronas correspondientes a la capa intermedia se aproximaron de manera experimental. Se consideró la regla empírica en cuanto a que la cantidad debería ser un número entre la cantidad de neuronas de entrada y salida. En ambos casos se consideró evaluar 50 a 450 unidades, con incrementos de 50, buscando un error menor a 0.001. Los mejores resultados se obtuvieron con 200 unidades.

Función de Activación. Tanto en la *hidden layer* como en la capa de salida se optó por la función sigmoide, ya que es una de las usualmente utilizadas.

Entrenamiento. Se decidió utilizar un algoritmo modificado de *back-propagation* sugerido por la librería utilizada (detalles en Sección 3) como algoritmo por defecto. No requiere elegir un *learning rate*, debido a que este algoritmo es adaptativo. La librería implementa el iRPROP [11] que es una modificación del RPROP [12]. Se utilizó como parámetro de corte alcanzar un error de 0.001. Por parte de los datos de entrenamiento, se utilizaron los caracteres extraídos de más de 100 patentes. Sin embargo, el entrenamiento de algunas letras fue acotado debido a que no fueron muy numerosas en el conjunto de datos, Fig. 2(b). En la Sección 4 se detallan las conclusiones del entrenamiento.

2.2. OCR Mediante ITM

Nuestro nuevo enfoque para el reconocimiento caracteres mediante *Template Matching* se basó en las características estructurales de las letras y números, para

reducir los candidatos a comparar. Mediante un árbol de búsqueda (Fig.3) se utilizaron estas características particulares para descartar comparaciones. Una vez alcanzadas las hojas del árbol, no se realizaba el contraste típico de *template matching*; en lugar de ello se introduce la comparación mediante los esqueletos de los caracteres candidatos.

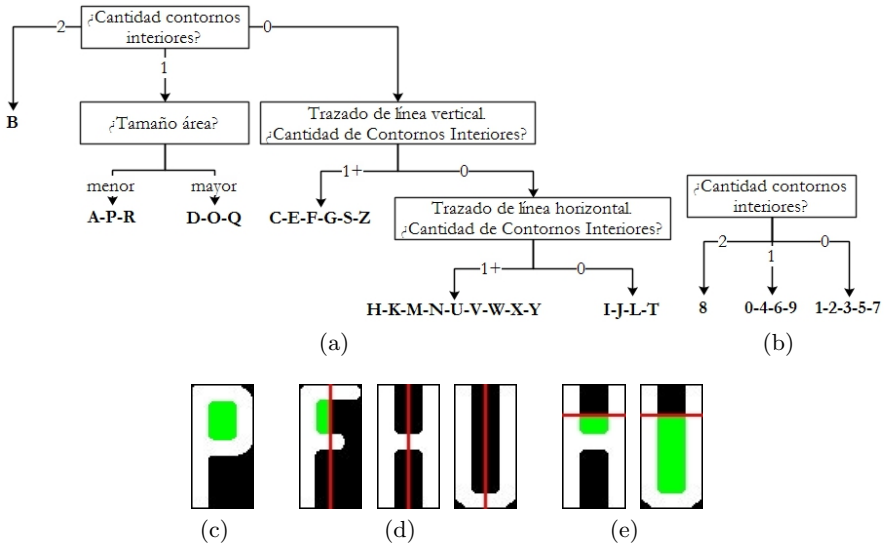


Figura 3. Árboles de búsqueda para letras (a) y dígitos (b). Ejemplo de letra con contorno interior (c) y casos donde es encesario el trazado de una línea vertical (d) y horizontal (e) para la detección de contornos interiores.

Como características en el caso de las letras, se consideró en primera instancia si poseía contornos interiores. De esta manera se podían diferenciar letras como la A, D, P, entre otras, de la C, I, T. Luego en caso de poseer contornos interiores se evaluaba el tamaño del mismo para diferenciar entre dos grupos posibles; uno de estos grupos es el correspondiente a las letras A, P y R donde el área del contorno interior era menor que una determinada constante, y el otro grupo de letras D, O y Q donde se presenta el caso en el que los contornos son de mayor área. Notemos que en el caso de obtener dos contornos, la letra B quedaba inmediatamente identificada. De no poseer contornos se contemplaban dos características más, la generación de contornos interiores mediante el trazado de una línea vertical u horizontal (Fig. 3(d)(e)).

Para la generación de contornos interiores se trazaba un segmento vertical, y al conjunto de caracteres que este trazado no le generase contornos se le realizaba el trazado de un segmento horizontal. Este último se realizaba a una distancia de la parte superior que era proporcional a 1/8 del alto de la imagen segmentada.

Esta decisión se debe a que a varios caracteres de los candidatos en esta situación se les genera un contorno con este trazado (por ejemplo las letras H, K, M, U). En el caso del árbol de búsqueda para los dígitos, sólo era necesario contemplar los contornos interiores para separar en grupos (Fig. 3(b)) y luego comparar con esqueletos.

Por último la comparación con esqueletos se basa en la superposición del caracter segmentado con los esqueletos de los caracteres candidatos. Para esto es necesario contar con las imágenes de los esqueletos de todos los posibles caracteres. En este caso se utilizaron imágenes de 40x80 pixels, por lo que el caracter segmentado debía normalizarse a este tamaño. El símbolo a identificar correspondería con el que coincidiera en la mayor cantidad de pixels al realizar la comparación entre el caracter segmentado y el esqueleto base, similar a la estrategia de *Template Matching* clásico. El beneficio de este método empleado, es que en general el esqueleto que le corresponda al carácter coincidirá prácticamente en la totalidad de los pixels. El caso de la letra C deberá compararse con el grupo de la C, E, F, G, S, Z, donde claramente la coincidencia será total con su letra correspondiente (Fig. 4). En este tipos de casos no tiene sentido seguir con la comparación, por lo que una coincidencia completa termina las comparaciones. Esto obliga a tener cierto cuidado en el orden de las comparaciones, ya que debe tenerse en cuenta el sentido de *inclusión* de los caracteres. Por ejemplo la F está totalmente contenida en la E, entonces primero se debe contrastar con la F y luego con la E.

(a) (b) (c) (d)

Figura 4. Ejemplo de una letra segmentada (a), sobrepuesta sobre un esqueleto incorrecto (b)(c) y sobre el correcto (d).

3. Resultados

Los experimentos realizados con los dos métodos fueron llevados a cabo con imágenes de diferentes tamaños (desde un mínimo de 200x400 hasta una máxima de 1600x1200). Los métodos propuestos fueron implementados en C, utilizando la librería OpenCV [13] para todo lo referente a procesamiento de imágenes y la librería FANN [14] para la implementación de las ANN. La plataforma utilizada en las pruebas fue una Pentium Dual-Core, 1.73 GHz, 2 GB RAM, Windows XP OS.

Inicialmente el test se realizó con 73 imágenes pertenecientes al conjunto de fotos utilizado para extraer caracteres para el entrenamiento de las redes. De este grupo se reconoció correctamente el $99,09\,\%$ de caracteres mediante las ANN y el $91,10\,\%$ con el método de ITM (Cuadro 1). Luego se realizó el test con un conjunto de imágenes ajenas al conjunto de datos de entrenamiento,

identificándose la patente en el 84, 67 % de los casos. En este caso los porcentajes de reonocimiento globales fueron de 86, 88 % para las ANN y 91, 08 % por parte de ITM. En promedio la identificación completa de cada patente se realizaba en 800ms. El reconocimiento individual de los caracteres fue de 0.5ms utilizándose redes neuronales y 0.75ms con ITM.

Cuadro 1. Resultados Experimentales.

	Perteneciente a Dataset		Ajenas al Dataset	
	Detecciones	Porcentajes	Detecciones	Porcentajes
Patentes Identificadas	73	-	127/150	84,67 %
Reconocimiento Letras ANN	216/219	98,63 %	305/381	80,05 %
Reconocimiento Letras ITM	197/219	89,95 %	332/381	87,14 %
Reconocimiento Dígitos ANN	218/219	99,54 %	357/381	93,70 %
Reconocimiento Dígitos ITM	202/219	92,24 %	362/381	95,01 %
Resultado Global OCR: ANN	434/438	99,09 %	662/702	**86,88 %**
Resultado Global OCR: ITM	399/438	91, 10 %	694/702	**91,08 %**

4. Análisis de los Resultados

Las pruebas realizadas arrojan buenos resultados en cuanto a la detección de la patente en la imagen. El sistema se comporta de modo robusto en cuanto a los ángulos de las tomas o a los cambios de iluminación. Sin embargo podemos diferenciar dos grupos donde presenta limitaciones.

Uno es el caso de los caracteres que están muy deteriorados y que no tienen una separación clara entre ellos, impidiendo su detección. En la Fig. 5 se muestra cómo afectan estas complicaciones en la imagen donde se realizará la búsqueda de contornos, uniendo caracteres contiguos en un único contorno. Esto amerita la utilización de algoritmos más complejos de segmentado, aunque seguramente estos también estén acompañados de mayor carga de procesamiento para la aplicación. Otra opción es la utilización de una estrategia aún más específica para este tipo de patentes, segmentando las secciones en base a la posición donde debería estar presente cada caracter y el área que éste ocupa en proporción con el tamaño de la patente. No obstante, el sistema contempla casos que no sean muy severos, utilizando la operación morfológica erosión-dilatación, para quitar las posibles uniones entre caracteres.

El otro grupo que impide la identificación de la placa, pertenece a los casos de patentes con protecciones plásticas de color negro. Estos recubrimientos impiden detectar el borde blanco exterior. Este problema podría ser solucionado utilizando para este caso el umbralado inverso (donde ahora el rectángulo exterior negro sería detectado).

En el caso del reconocimiento de caracteres, ambas estrategias introducen un alto grado de reconocimiento. Sin embargo en base a los resultados expuestos en la sección anterior, podemos notar la fuerte dependencia de las redes neuronales en cuanto a su entrenamiento. Esto se remarca con las pruebas basadas en imágenes utilizadas para este fin, donde se alcanza un reconocimiento global de

(a) (b) (c) (d)

Figura 5. Imagen de patente degradada (a) y el resultado de aplicar un nivel de umbral (b). Ejemplos de patentes con protecciones plásticas (c)(d).

99,09 %. En cambio, en el otro conjunto de imágenes la identificación cae a un 86,88 %. Aquí se observa la gran ventaja del método ITM, que no necesita de entrenamiento alguno, y muestra un porcentaje de reconocimiento comparable a las ANN, incluso superandólas en el segundo caso en más de 4 %. Estas falencias junto con confusiones esperables por parte de las ANN, como son los casos E-F, U-V ó O-Q, pueden ser corregidas en buen porcentaje aumentando el tamaño del conjunto de entrenamiento.

Luego en referencia a los tiempos de ejecución, las ANN muestran ser más rápidas que el método ITM. Sin embrago, debemos considerar que las redes utilizan caracteres de tamaño 15x30, en tanto que el de las ITM es de 40x80. Esta diferencia representa aproximadamente un 85 % más de pixels a comparar, implicando un grado mayor de procesamiento en la comparación de esqueletos.

También podemos detallar que un beneficio del método ITM es su flexibilidad y transparencia. Al no ser una estrategia de *caja negra*, es totalmente modificable y parametrizable para reconocer distintos conjutos de caracteres, evitando un nuevo entrenamiento como ocurre en el caso de las redes neuronales.

Un beneficio de las ANN es que al ser robustas no agregan tanta responsabilidad al método de segmentado. La tolerancia del método permite lidiar con caracteres de calidad pobre. Esto mismo se logra con ITM, ya que como podemos ver en la Figura 6, los caracteres segmentados se presentan bastante degradados, y sin embargo esto no impide su identificación por ambas estrategias.

(a) (b)

Figura 6. Ejemplos de la calidad de letras (a) y dígitos (b) que se identificaron correctamente por ambos métodos.

5. Conclusión

Presentamos dos métodos que permiten el reconocimiento automático de patentes argentina de autos con un alto porcentaje de efectividad y comparamos ambas alternativas en lo que se refiere al reconocimiento de los caracteres. Es conocido que las ANN son comúnmente utilizadas gracias a su alta tolerancia al

ruido en los datos de entrada. Esto ha sido comprobado en el presente trabajo mediante el diseño y la implementación de una ANN, aunque también destacamos uno de sus puntos limitantes, el costo que implica su entrenamiento. Por ello presentamos un método alternativo denominado ITM, que no requiere de este conocimiento previo, y que tiene un desempeño tanto en reconocimiento como en tiempo totalmente comparable a las ANN's.

Se han obtenidos buenos resultados con ambos métodos logrando un reconocimiento mayor al 90 %. La flexibilidad del ITM lo presenta como un método con muchas posibilidades por lo que se considera seguir mejorando sus prestaciones, utilizando tamaños de esqueletos de tamaños menores y considerando estrategias más eficientes para la implementación de las diferentes etapas del sistema.

Referencias

1. R.A. Lotufo, A.D. Morgan, and AS. Johnson. Automatic number-plate recognition. *Proceedings of First IEEE International Workshop in Electronic Design*, V01.035:6/1–6/6, 1990.
2. D.G. Bailey, D. Irecki, B.K. Lim, and L. Yang. Test bed for number plate recognition applications. In *Electronic Design, Test and Applications, 2002. Proceedings. The First IEEE International Workshop on*, pages 501–503, 2002.
3. SicOCR. Sistema comercial de anpr, www.sicsa.com.ar.
4. Sistema comercial de ANPR, www.addsp.com.
5. SCANTRACK. Sistema comercial de anpr, http://intekio.com.
6. Liam Keilthy. Anpr system performance. *Parking Trend International*, June 2008.
7. S. Alireza Kasaei S. Hamidreza Kasaei, S. Mohammadreza Kasaei. New morphology-based method for robust iranian car plate detection and recognition. *International Journal of Computer Theory and Engineering*, vol. 2:264–268, 2010.
8. Kumar Parasuraman and P.Vasantha Kumar. An efficient method for indian vehicle license plate extraction and character segmentation. In *2010 IEEE International Conference on Computational Intelligence And Computing Research*, 2010.
9. M.I.Khalil. Car plate recognition using the template matching method. *International Journal of Computer Theory and Engineering*, vol. 2:683–687, 2010.
10. J. J. Rodriguez J. Maudes C. Garcia-Osorio, J. F. Diez-Pastor. License plate number recognition - new heuristics and a comparative study of classifiers. In *5th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2008.
11. Christian Igel and Michael Husken. *Improving the Rprop Learning Algorithm*, pages 115–121. Proceedings of the Second International Symposium on Neural Computation. 2000.
12. Martin Riedmiller and Heinrich Braun. A direct adaptive method for faster backpropagation learning: The rprop algorithm. In *IEEE International Conference On Neural Networks*, pages 586–591, 1993.
13. Open Source Computer Vision (OpenCV). http://opencv.willowgarage.com/.
14. Fast Artificial Neural Network (FANN). http://leenissen.dk/fann/wp/.