

Sistema de Reconocimiento de Matrículas de Vehículos Mediante Clasificación por Redes Neuronales

Procesamiento de
Imágenes por
Computador

Máster en Robótica y
Automatización

Universidad Carlos III de Madrid
2012



Autores

Verónica González Pérez
(100289866@alumnos.uc3m.es)

Raúl Péruela Martínez
(raul.perula@alumnos.uc3m.es)

Índice

1. Introducción	2
2. Objetivos	3
3. Sistema de reconocimiento de matrículas usando visión artificial	4
3.1 Descripción de Operaciones Morfológicas utilizadas	4
3.2 Descripción del Sistema de Reconocimiento de Matrículas	5
3.3 Localización y segmentación de la matrícula	6
3.3.1 Localización respecto al eje vertical	6
3.3.2 Localización respecto al eje horizontal	9
3.4 Extracción de caracteres	9
3.4.1 Umbralización	9
3.4.2 Segmentación y etiquetado	10
4. Sistema de reconocimiento de caracteres usando redes neuronales	12
4.1. Características de la red neuronal	12
4.1.1. Entradas (Inputs)	12
4.1.2. Objetivos (Targets)	12
4.1.3. Tipo de red neuronal	13
4.1.4. Función de transferencia	13
4.1.5. Función de entrenamiento	14
4.1.6. Función de error	15
4.2. Método de entrenamiento	15
5. Resultados	16
6. Conclusiones y Futuras mejoras	26
Bibliografía	27

1. Introducción

Hoy en día existen miles de vehículos circulando por nuestras carreteras, y por ello es necesario un control riguroso de todo este tráfico en masa de manera eficiente y rápida. Este trabajo puede ser desempeñado por un operario pero habría que destacar un inconveniente muy importante como es la ineficiencia en la rapidez a la hora de identificar el número de placa del vehículo. Para solventar este problema se han diseñado sistemas para el reconocimiento automático de matrículas facilitando y agilizando la tarea del operador.

Esta tecnología se ha convertido en una aplicación cotidiana en nuestros días, para la seguridad y control de todos los vehículos que transitan a diario, siendo estos sistemas instalados en parkings, peajes, controles de acceso, etcétera. De esta forma la vigilancia en masa se convierte en una tarea fácil y agilizada, pudiendo saber por ejemplo, el tiempo que ha permanecido un coche estacionado en un aparcamiento almacenando la imagen en una base de datos y más tarde buscar y comparar la misma matrícula sabiendo así la duración exacta. Para el acceso a lugares restringidos con un reconocimiento previo de la matrícula, e incluso para controlar infracciones provocadas por ciertos vehículos en vías públicas o carreteras.

Todo esto es posible gracias a una captura de imagen y un software capaz de extraer la matrícula para más tarde analizarla e identificar su número en la placa por medio de técnicas de visión artificial. En estos sistemas las cámaras utilizadas suelen ser infrarrojas para tener una visión semejante durante el día y la noche. En nuestro caso la cámara no será de este tipo debido a la falta de medios, pero el sistema intentará ser lo más eficaz posible a la hora de analizar la matrícula.

Además, existen diversos métodos de reconocimiento de caracteres en imágenes. Uno de los métodos que tiene más relevancia en estos tiempos y de los que se están basando muchas ramas investigadoras como problemas de clasificación y reconocimiento de patrones de voz, imágenes, señales, etc., incluso se han utilizado para encontrar patrones de fraude económico, hacer predicciones en el mercado financiero, hacer predicciones de tiempo atmosférico, etc., son las redes neuronales.

Las redes neuronales son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso natural. De ello, se ha extraído el conocimiento suficiente para aplicarlo en la clasificación de los caracteres a partir de las imágenes de las matrículas de los coches pudiendo extraer en forma de texto para poder ser analizado posteriormente.

En este trabajo se ha realizado el reconocimiento mediante visión artificial para la detección de las matrículas de los vehículos y la extracción de cada uno de los caracteres alfanuméricos que la propia matrícula contiene. Posteriormente se ha realizado el procesamiento de clasificación de cada una de las imágenes de cada carácter que contiene la matrícula para ser clasificado mediante una red neuronal.

2. Objetivos

El objetivo principal de este trabajo es crear una aplicación realizada con Matlab para el reconocimiento automático de matrículas en imágenes captadas con una cámara, junto con la detección individual de sus caracteres.

Para el reconocimiento de matrículas, se utilizarán herramientas de visión artificial y de redes neuronales proporcionadas por los toolbox específicos de Matlab, como técnicas de segmentación y transformaciones morfológicas, además de realizar la implementación de las funciones necesarias para que el sistema tenga un rendimiento lo más óptimo posible y pueda ser funcional en una situación real.

Posteriormente, se identificarán todos los caracteres de la matrícula del vehículo mediante la clasificación con redes neuronales artificiales, mostrando así el número y las letras identificativas de la matrícula captada.

3. Sistema de reconocimiento de matrículas usando visión artificial

3.1 Descripción de Operaciones Morfológicas utilizadas

En el trabajo se utilizarán ciertas operaciones morfológicas que serán descritas a continuación, dando una breve explicación de en qué consiste cada una de ellas.

Dilatación

La dilatación se basa en la expansión de los valores positivos. En este caso en una imagen en escala de grises, los píxeles positivos se aglomerarán unos con otros siempre y cuando estos se encuentren cercanos entre sí, en el caso de que estos no estén próximos se expandirán. Para una imagen binaria, los píxeles unitarios se expandirán por medio de una máscara (M). Al igual que en una imagen en escala de grises, los píxeles cercanos con el mismo valor se unirán, mientras que los que estén alejados se expandirán.

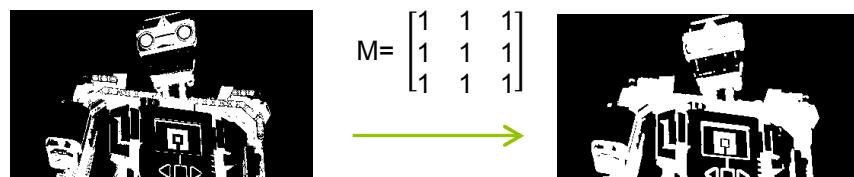


Figura 3.1 Dilatación en una imagen.

Erosión

Es un proceso contrario a la dilatación. En imágenes en escala de grises se basa en la contracción de los picos positivos, mientras los valores negativos son expandidos, es decir, las zonas claras rodeadas de zonas oscuras son cubiertas por estas últimas, mientras que las zonas oscuras rodeadas de zonas claras son agrandadas. En el caso de una imagen binaria este método es un eliminador de ruido, ya que los píxeles inferiores al tamaño de la máscara (M) son eliminados.

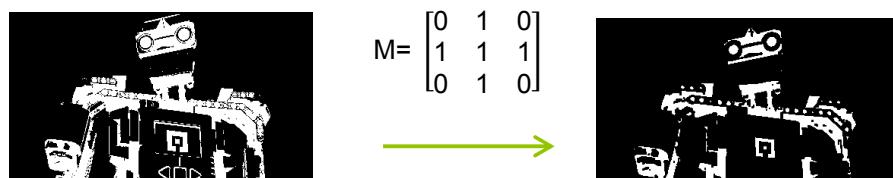


Figura 3.2. Erosión en una imagen.

Apertura

La apertura está basada, primero en la erosión y luego la dilatación, utilizando una máscara de igual valor para las dos operaciones. Este método separa objetos contiguos y simplifica la imagen.

$$\text{Apertura}(A, B) = A \circ B = \text{Dilatación}(\text{Erosión}(A, B), B) = (A \ominus B) \oplus B$$

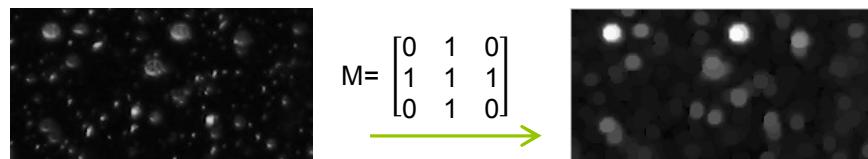


Figura 3.3. Apertura en una imagen (opening).

Es un método útil para eliminar el ruido de una imagen, debido a que por medio de la erosión los píxeles de valor positivo se reducen o son eliminados, mientras que con la dilatación, sólo son expandidos los píxeles no eliminados de valor unitario.

Cerradura

Es el caso contrario a la apertura, primero se hace una dilatación y luego una erosión. Su uso principal es para conectar objetos, ya que por medio de la dilatación se rellenan todos los huecos que existen en la imagen y por medio de la erosión se suavizan todos los bordes. Al contrario que la apertura, no es útil para eliminar el ruido.

$$\text{Cerradura}(A, B) = A \bullet B = \text{Erosión}(\text{Dilatación}(A, B), B) = (A \oplus (B)) \ominus B$$

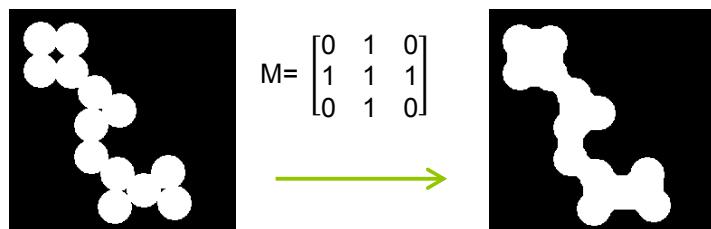


Figura 3.4 Cerradura en una imagen (closing).

3.2 Descripción del Sistema de Reconocimiento de Matrículas

El sistema desarrollado para el reconocimiento de matrículas se puede dividir en varias secciones. Inicialmente será necesario hacer un tratamiento de la imagen para ser analizada, después se segmentará la matrícula del resto de la imagen y una vez que tengamos únicamente la matrícula se procederá a la extracción de sus caracteres.



Figura 3.5 Etapas para la extracción de caracteres.

Así mismo, cabe destacar varios factores, como la reflectancia e intensidad de la matrícula. Se ha estipulado también un rango determinado para la localización de la matrícula, es decir, se ha

supuesto que la cámara siempre estará en una posición determinada, lo cual la altura de la matrícula oscilará dependiendo de la altura del coche, así como su posición derecha o izquierda en la imagen. De esta forma, la imagen podrá ser recortada para reducir información no válida para la detección de la matrícula.

Es necesario tener en cuenta que las imágenes utilizadas en este trabajo están sacadas de bases de datos distintas, lo cual el tipo de matrículas es diferente.

A continuación se mostrarán los pasos necesarios para la identificación de la matrícula a partir de técnicas de segmentación y transformaciones morfológicas en visión artificial.

3.3 Localización y segmentación de la matrícula

Para la obtención de la localización de la matrícula es necesario obtener unas coordenadas verticales y horizontales de donde se encuentra esta. Cabe destacar, como se ha comentado anteriormente, la matrícula tiene una alta característica de reflectancia, la cual hará más fácil su localización.

Primeramente es necesario pasar la imagen a escala de grises. Las imágenes en escala de grises suelen utilizar hasta 256 niveles para representar los tonos intermedios desde el color negro (0) hasta el color blanco (255), dando una gran información sobre el brillo en la imagen y aplicando simplicidad en ella. Aunque el ser humano sea incapaz de reconocer dos niveles de gris parecidos, esto puede ser una información clave a la hora de analizar una imagen.



Figura 3.6. (a) Imagen normal, (b) Imagen en escala de grises

La imagen se recortará para reducir la información no deseada a la hora de analizar la imagen para su localización horizontal y vertical.

3.3.1 Localización respecto al eje vertical

Una vez la imagen convertida a escala de grises y recortada, se procederá con su localización respecto al eje vertical. Para ello se utilizará el filtro **Bottom-hat**, caracterizado por hacer resaltar los detalles oscuros sobre fondo blanco, invirtiendo la imagen. En este caso, la matrícula tiene un fondo blanco intenso debido a la reflectancia de esta, por lo tanto, los caracteres negros en la matrícula serán objetos considerablemente resaltados. Si aplicamos este filtro a la imagen en escala de grises, se generan captaciones de bajo nivel que son eliminadas con ajustes en el nivel del umbral, ya que este es considerado como ruido [2].

La imagen resultante del filtro Bottom-hat (I_{BH}) se obtiene aplicando un cierre (close) en la imagen en escala de grises (I_C) y restándola con la imagen original (I_o), ambas ya recortadas.

$$I_{BH} = I_C - I_o$$

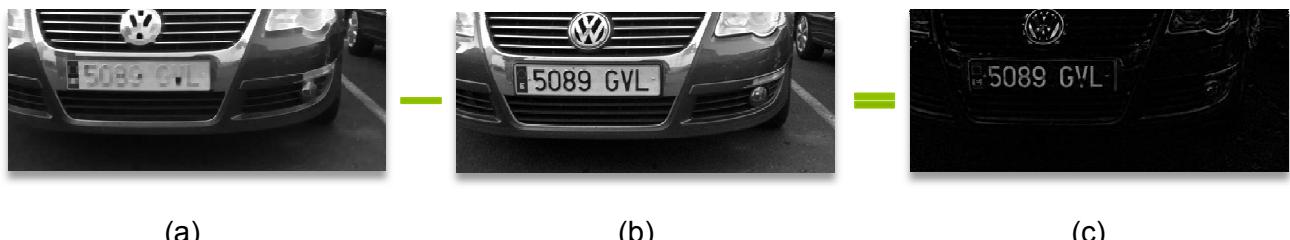


Figura 3.7. (a) Imagen con cerradura, (b) Imagen en escala de grises, (c) Resultado de la resta de imágenes (Filtro Bottom-hat).

Una vez aplicado el filtro Bottom-hat se aplicará una detección de bordes por medio de un **filtro Canny**. Este filtro consta de una serie etapas, primero utiliza un filtro Gaussiano para suavizar la imagen $f(x, y)$, reduciendo así el ruido, por medio de una función Gaussiana $h(x, y)$ y un parámetro escalar (σ).

$$h(x, y) = e^{-\frac{x^2 + y^2}{2\sigma^2}}$$

Más tarde se estima el gradiente definido como:

$$G[f(x, y)] = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial}{\partial x} f(x, y) \\ \frac{\partial}{\partial y} f(x, y) \end{bmatrix}$$

Es necesario que en cada pixel de la imagen se obtenga la dirección y magnitud del gradiente, convolucionando la imagen por medio de estos operadores.

$$|G| = \sqrt{G_x^2 + G_y^2} \quad \phi(x, y) = \tan^{-1} \frac{G_y}{G_x}$$

Una vez obtenida la magnitud y dirección del gradiente de cada pixel, estos son comparados con la magnitud de sus píxeles vecinos en la dirección obtenida, dejando así los píxeles que son máximos en su contorno, a esto se le llama “cálculo de no máximos”.

$$T_1 \leq G(x, y) < T_2 \text{ (detección bordes débiles)} \quad G(x, y) \geq T_2 \text{ (detección bordes fuertes)}$$

Finalmente, el método utiliza dos umbrales (T_1 y T_2) para detectar los bordes fuertes y débiles, conectando los píxeles que se encuentran por debajo del umbral inferior con los que superan el umbral superior.

Una vez visto como funciona un filtro Canny, se procederá a explicar la estrategia utilizada con él. Esta estrategia consta de la aplicación del filtro Canny, junto con el cierre, apertura y dilatación de la imagen. Al utilizar este método con el umbral obtenido automáticamente con Matlab y ajustado para cada caso de imagen (el umbral será explicado más adelante) obtendremos la altura de los números en la matrícula, siendo importante para determinar la altura de esta. Como se puede observar en las imágenes, no es necesario obtener los números de la matrícula a la perfección ya que solo queremos sacar la altura de estos. Las siguientes transformaciones morfológicas son utilizadas para conseguir una región cerrada que nos indique la posición y altura a la que recortar la imagen.

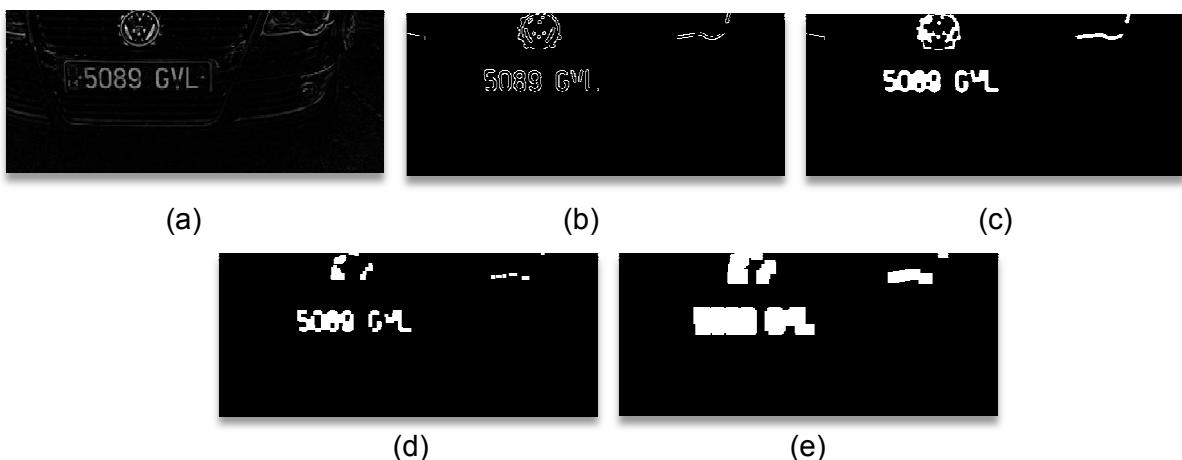


Figura 3.8. Proceso para detectar la altura y posición de la matrícula. (a) Filtro Bottom-hat, (b) Filtro Canny, (c) Close, (d) Erosión, (e) Dilatación.

Una vez aplicadas las transformaciones, se procederá con el **etiquetado y la discriminación** en la imagen.

Como se muestra en la figura 3.8 (e) no siempre quedará un solo área correspondiente con el de la matrícula, si no que habrá más regiones dependiendo de la fisiología del coche. El resultado en la anterior imagen ha sido varias regiones cerradas, de las cuales las que tienen más área y altura son correspondientes a la matrícula. Por lo tanto se calculan las coordenadas de la región que más área tenga atendiendo a un rango de altura, discriminando las áreas más pequeñas o que no cumplan con el rango establecido. El rango establecido es entre 30 y 60 píxeles el cual se obtuvo por diferentes pruebas con todos los tipos de matrículas. La región escogida nos dará las coordenadas de la altura y la posición donde se encuentra la matrícula dando paso a su recorte respecto al eje vertical de la imagen.



Figura 3.9. Matrícula recortada sobre el eje vertical

3.3.2 Localización respecto al eje horizontal

Una vez que tenemos la altura de la matrícula y su localización, es necesario recortar el ancho de esta. Los pasos a seguir serán muy parecidos a los que hemos seguido para el recorte horizontal.

Inicialmente, hacemos un filtro Bottom-hat a la imagen ya recortada de la matrícula, seguidamente hacemos una detección de bordes con el filtro Canny (le aplicaremos distinto umbral que la anterior vez debido a que ahora si necesitamos que se detecten todos los caracteres de la matrícula) y haremos un cierre en la imagen. De esta forma, la zona obtenida debe ser el área de la matrícula. Al hacer el **etiquetado y discriminación** en la imagen siempre se elegirá la región con el área más grande, ya que en este caso esa área siempre corresponderá con la matrícula. De esta forma, la imagen se recortará a lo largo del eje horizontal dado el ancho y las coordenadas de esta área.



Figura 3.10. Proceso para el recorte de la matrícula sobre el eje horizontal.

3.4 Extracción de caracteres

Una vez que tenemos la placa de la matrícula recortada, se extraerán sus caracteres. El primer paso es la umbralización de esta, seguido de la segmentación.

3.4.1 Umbralización

La imagen recortada ya está en escala de grises por lo tanto el paso siguiente es aplicar un umbral óptimo para su binarización. Este umbral será calculado directamente por Matlab con el comando `graythres`, utilizando el método Otsu, ya que cada imagen tendrá un umbral diferente.

El método Otsu [5] es muy utilizado para la extracción de objetos en imágenes donde hay una clara diferencia con el fondo. Este método se basa en la máxima varianza entre el fondo y el objeto (siendo esta una medida de la dispersión de los valores).

Principalmente se tomarán dos segmentos tal como ($K_0(t)$ y $K_1(t)$) a partir de umbral (t), siendo esta la variable buscada. Se establecerá G como el valor máximo de gris y $p(g)$ la probabilidad de ocurrencia del valor de gris $0 < g < G$. La probabilidad de ocurrencia de cada segmento vendrá determinada como:

$$P_0(t) = \sum_{g=0}^t p(g) \text{ y } P_1(t) = \sum_{g=t+1}^G p(g) = 1 - P_0(t)$$

\bar{g} será la media aritmética estimada a partir de todos los valores de gris de la imagen, y \bar{g}_0 y \bar{g}_1 será la media de los valores de gris en cada segmento. Por lo tanto, la varianza en cada segmento corresponderá con:

$$\sigma_0^2(t) = \sum_{g=0}^t (g - \bar{g}_0)^2 p(g) \text{ y } \sigma_1^2(t) = \sum_{g=t+1}^G (g - \bar{g}_1)^2 p(g)$$

Por lo tanto el resultado de la varianza calculada entre los posibles segmentos deberá ser el más grande, es decir, es necesario tomar un valor de umbral t para que el resultado del cociente sea el máximo.

$$Q(t) = \frac{\sigma_{zw}^2(t)}{\sigma_{in}^2(t)}$$

$$\begin{aligned} \sigma_{zw}^2(t) &= P_0(t)(\bar{g}_0 - \bar{g})^2 + P_1(t)(\bar{g}_1 - \bar{g})^2 && [\text{Varianza entre los segmentos}] \\ \sigma_{in}^2(t) &= P_0(t) \cdot \sigma_0^2(t) + P_1(t) \cdot \sigma_1^2(t) && [\text{Varianza de los segmentos}] \end{aligned}$$

En ocasiones, hay ciertos problemas con el umbral establecido, debido a que las matrículas no son perfectas, si no que tienen pequeñas imperfecciones, como suciedad, impurezas o que esta no esté del todo recta. En consecuencia a esto debemos de adecuar el umbral obtenido a estas imperfecciones, siendo el umbral algo menor del establecido por el método Otsu.

Una vez que el proceso de umbralizado se ha completado es necesario invertir la imagen, ya que para el próximo paso (segmentación) los caracteres deben de ser claros y el fondo oscuro. Hay que tener en cuenta que este proceso solo es válido para matrículas de este tipo, ya que si se utilizasen matrículas con fondo oscuro el proceso no sería válido.



Figura 3.11. Proceso de binarización e inversión de imagen

3.4.2 Segmentación y etiquetado

Una vez la matrícula recortada y binarizada con un umbral óptimo se procederá a su segmentación para la obtención de los caracteres.

Como se ha comentado antes, el proceso de segmentado se produce cuando los caracteres de la matrícula poseen un color claro (valor 1) y un fondo negro (valor 0). A la hora de hallar todas las

regiones de color blanco en la imagen con el etiquetado, se establecerán rangos para el tamaño de las letras. En este caso, la altura de los caracteres debe de estar entre [17, 40] píxeles y la anchura entre [5, 24] píxeles, siendo el rango estándar de todos los caracteres en la placa, teniendo en cuenta la distancia que debe de haber entre la matrícula y el coche. Estos rangos se han determinado por medio de pruebas prácticas sobre la base de datos utilizada de las matrículas.



Figura 3.12. Determinación de coordenadas de los caracteres

Una vez halladas las coordenadas de donde se encuentra cada carácter se aplica una mejora de intensidad a la imagen de la matrícula original para obtener un carácter lo más bueno posible, ya que en la determinación de estas coordenadas no se ha tenido en cuenta la calidad de este. Despues se aplica una binarización de acuerdo con el umbral de la nueva imagen mejorada. Cabe destacar también, que el símbolo de matrícula europea ha sido seleccionado pero no se tendrá en cuenta para el resultado final.

Una vez que todos los caracteres han sido detectados, recortamos la imagen acorde a estas coordenadas, haciendo una dilatación y una erosión para eliminar el mayor ruido posible que haya en la imagen de cada carácter. Los caracteres quedarán como se muestra en la siguiente figura:

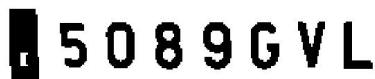


Figura 3.13. Caracteres extraídos de la matrícula

Cuando las imágenes estén listas, se aplicará una redimensión de estas dejando todos los caracteres de un mismo tamaño y serán devueltos para ser reconocidos mediante una red neuronal.

4. Sistema de reconocimiento de caracteres usando redes neuronales

Una vez realizado todo el reconocimiento de la matrícula y extraídos los caracteres uno a uno para su clasificación se procede a la clasificación como tal. Esta clasificación se realiza mediante una configuración óptima de los parámetros de la red neuronal artificial y obteniendo un método efectivo de entrenamiento de la red neuronal para obtener unos resultados lo mejores posibles.

Lo más complicado a la hora de clasificar mediante redes neuronales artificiales es ajustar los parámetros internos de la red, así como, decidir cuantas capas ocultas debe tener y cuantas neuronas debe tener cada capa. En este caso, se ha decidido tener una capa oculta de 60 neuronas todas interconectadas y una capa de salida con tantas neuronas de salida como resultados se desean, es decir, 26 neuronas (letras) más 10 neuronas (números) que en total serían 36 neuronas en la capa de salida.

Estos parámetros no son triviales su elección, para ello se han realizado una serie de pruebas previas a su elección y se han estudiado varios “papers” o artículos de investigación mediante redes neuronales para hacer la elección de los parámetros y del tipo de red neuronal que posteriormente se especificará cual se ha escogido.

4.1. Características de la red neuronal

4.1.1. Entradas (Inputs)

Como entrada, la red neuronal recibirá las imágenes individuales de cada carácter reconocido por la parte de análisis de visión artificial. Estas entradas deberán tener unas dimensiones específicas, una matriz de 7x5 transformada en forma de vector quedando con unas dimensiones de 35x1, para que se pueda realizar la clasificación, en caso contrario se procederá a parar el sistema dando a conocer el motivo de la parada.

La elección de estas dimensiones se ha tomado para no tener que dar unas dimensiones muy grandes a las imágenes de entrada de la red neuronal artificial. Esto conlleva simpleza a la entrada pero complejidad en el caso de que la parte de visión va a tener que realizar un reconocimiento bastante preciso.

4.1.2. Objetivos (Targets)

Los objetivos de la red neuronal son una base de datos creada la cual contiene los valores en perfecto estado de los caracteres, ya sean letras o números. Esto servirá para ajustar los parámetros de los pesos y “bias” y, de este modo, una vez entrenada la red neuronal se pueda realizar una comparación mediante correlación entre matrices para obtener el resultado.

La correlación indica la fuerza y la dirección de una relación lineal entre dos variables aleatorias. Se considera que dos variables cuantitativas están correlacionadas cuando los valores de una de ellas varían sistemáticamente con respecto a los valores homónimos de la otra: si

tenemos dos variables (A y B) existe correlación si al aumentar los valores de A lo hacen también los de B y viceversa. La ecuación que utiliza es:

$$r = \frac{\sum_{m} \sum_{n} (A_{mn} - \bar{A})(B_{mn} - \bar{B})}{\sqrt{\left(\sum_{m} \sum_{n} (A_{mn} - \bar{A})^2 \right) \left(\sum_{m} \sum_{n} (B_{mn} - \bar{B})^2 \right)}}$$

Donde \bar{A} es la media de los valores de A y \bar{B} es la media de los valores de B.

4.1.3. Tipo de red neuronal

El tipo de red neuronal que se ha usado es una red de tipo feed-forward con retropropagación (**newff**), esta se encuentra configurada con varias capas. La primera capa tiene los pesos que llegan de la entrada. Cada subcapa tiene un peso que viene de la capa anterior. Todas las capas tienen "bias". La última capa contendrá los valores de la salida.

La retropropagación es un algoritmo de aprendizaje supervisado que se usa para entrenar redes neuronales artificiales. El algoritmo consiste en minimizar un error (comúnmente cuadrático) por medio de descenso de gradiente, por lo que la parte esencial del algoritmo es cálculo de las derivadas parciales de dicho error con respecto a los parámetros de la red neuronal.

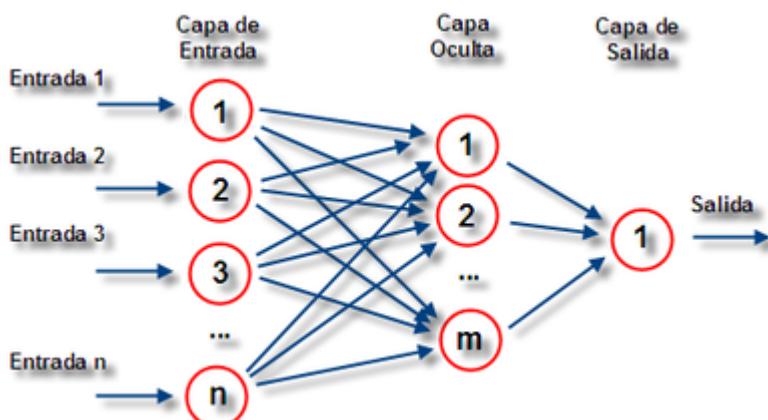
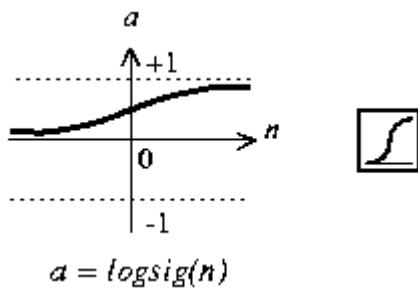


Figura 4.1. Gráfico de Redes Neuronales

4.1.4. Función de transferencia

La función de transferencia usada en la capa intermedia y en la capa de salida ha sido una función de tipo logarítmica sigmoide.

La curva que representa dicha función es la siguiente:



Dicha curva sigue la siguiente ecuación:

$$P(t) = \frac{1}{1 + e^{-t}}$$

Donde P sería lo que en Matlab se le llama “**logsig**” y ‘ t ’ sería el parámetro de entrada, que en nuestro caso sería una matriz de $N \times M$.

4.1.5. Función de entrenamiento

Uno de las elecciones más importantes para configurar una red neuronal artificial es la función de entrenamiento, ya que será ésta la encargada de ajustar los pesos y que posteriormente se dé un buen resultado o no de las salidas. La función que se ha escogido ha sido la de Levenberg-Marquardtbackpropagation.

En Matlab llamada “**trainlm**”. Esta función realiza el entrenamiento de la red neuronal actualizando los valores de los pesos y “bias” de acuerdo a la optimización de Levenberg-Marquardt.

La función “**trainlm**” suele ser a menudo el algoritmo de retropropagación más rápido de todos los del toolbox y es altamente recomendado como algoritmo de control supervisado, aunque requiere más memoria que otros algoritmos. Además, soporta entrenamiento con conjuntos de validación y test pudiendo así dividir los datos para su entrenamiento.

La retropropagación se usa para calcular la jacobiana j_X del rendimiento $perf$ con respecto a los pesos y “bias” de la variable X . Cada variable se ajusta de acuerdo a la ecuación de Levenberg-Marquardt,

$$j_i = j_X * j_X$$

$$j_e = j_X * E$$

$$dX = -\frac{j_i + I * mu}{j_e}$$

Donde E es todos los errores e I es la matriz de identidad.

Para cada época, si el rendimiento decrece hacia el objetivo, entonces la tasa de entrenamiento se incrementa por el factor ‘lr_inc’. Si el rendimiento incrementa por encima del factor máximo umbral, la tasa de entrenamiento se ajusta por el factor ‘lr_dec’.

El entrenamiento termina cuando alguna de las siguientes condiciones ocurre:

- El número máximo de épocas (repeticiones) se alcanza.
- El tiempo máximo de espera se alcanza.
- El rendimiento es minimizado consiguiendo el objetivo deseado.
- El rendimiento del gradiente cae por debajo del mínimo establecido.

4.1.6. Función de error

La función de error usada es una función de error cuadrático (Sum squared error, SSE) la cual se basa en la suma de los cuadrados de los valores objetivo y los valores medios de la variable responsable. La ecuación que utiliza es la siguiente:

$$\text{ESS} = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2$$

4.2. Método de entrenamiento

El método de entrenamiento lo es todo para que el resultado salga lo más parecido al esperado. Por ello, hay que decidir cómo y cuantas veces se va a entrenar la red neuronal artificial, eligiendo en cada momento los valores de las entradas y de los objetivos.

En este caso, se ha realizado un entrenamiento triple, es decir, se ha entrenado la red neuronal tres veces por carácter de entrada. La primera vez se realiza un entrenamiento “limpio”, es decir, se entrena la red neuronal sin ruido, en los valores de entrada y objetivos serán los proporcionados por la función de creación de los caracteres sin nada de ruido. Además se ajustan los valores de las condiciones de parada que serían, el valor umbral del SSE a 0.1, el número de épocas a 5000 y se establecería el valor del ‘momento’ a 0.95.

La segunda vez se realiza un entrenamiento con ruido, es decir, se utilizan 10 conjuntos de datos que representan caracteres ruidosos y no ruidosos. Los caracteres ruidosos se obtienen al añadir un ruido aleatorio de diferente intensidad para que la red ajuste sus pesos y “bias” en los peores casos que se pueden dar al recibir las imágenes de los caracteres de la parte de visión. Los valores de los parámetros de condición de parada se vuelven a ajustar, el valor del umbral del SSE quedaría a 0.6 y el número de épocas se reduciría hasta 300.

La tercera vez se realiza un entrenamiento sin ruido como la primera vez, sin embargo, los parámetros de las condiciones de parada son diferentes. En este caso el valor umbral del SSE sería 0.1 y el número de épocas sería 500.

Por cada entrenamiento se guardan los valores del SSE para posteriormente obtener la media y la varianza de dichos valores y así obtener valores estadísticos útiles.

5. Resultados

En esta sección se presentarán los resultados obtenidos de las ejecuciones de localización y reconocimiento de las matrículas de los coches, obteniendo de cada matrícula el carácter reconocido junto a su error. Por último se presenta el porcentaje de acierto de cada matrícula.

Coche 1



Imagen matrícula extraída

Carácter	SSE
X	13
7	12
6	17
2	14
H	17
F	14
C	13

X762HFC

Acierto 6/7 (85%)

■4762HFC

Coche 2



Imagen matrícula extraída

Carácter	SSE
T	11
B	20
2	14
D	18
B	20
2	14
J	12

TB2DB2J

Acierto 3/7 (43%)

T023T2■

Coche 3



Imagen matrícula extraída

Carácter	SSE
T	11
U	15
9	17
5	17
2	14
2	14
Y	10

TU9522Y

Acierto 6/7 (85%)

T09522Y

Coche 4



Imagen matrícula extraída

Carácter	SSE
T	11
U	15
Y	10
8	17
9	17
U	15
1	8
C	13

TUY89U1C

Acierto 3/8 (37%)

T07881AC

Coche 5



Imagen matrícula extraída

Carácter	SSE
5	17
7	12
B	20
B	20
8	17
G	15

57BB8G

Acierto 3/7 (43%)

37088G

Coche 6



Imagen matrícula extraída

Carácter	SSE
5	17
0	16
9	17
G	15
6	17
Y	10
L	11

509G6YL

Acierto 3/7 (43%)

5099GVL

Coche 7



Imagen matrícula extraída

Carácter	SSE
5	17
9	17
0	16
6	17
D	18
C	13
B	20

5906DCB

Acierto 6/7 (85%)

7906DCB

Coche 8



Imagen matrícula extraída

Carácter	SSE
6	17
U	15
9	17
5	17
B	20
B	20
E	18

6U95BBE

Acierto 3/7 (43%)

6885BBG

Coche 9

Imagen matrícula extraída

Carácter	SSE
6	17
Y	10
X	13
B	20
C	13
Y	10
P	15

6YXBCYP

Acierto 3/7 (43%)

6741CVP**Coche 10**

Imagen matrícula extraída

Carácter	SSE
8	17
6	17
6	17
B	20
9	17
R	18
B	20

866B9RB

Acierto 0/7 (0%)

SAE5640**Coche 11**

Imagen matrícula extraída

Carácter	SSE
5	17
1	8
E	18
B	20
O	12
X	13
Q	17

51EBOXQ

Acierto 1/7 (14%)

SAE5640

Coche 12

Imagen matrícula extraída

Carácter	SSE
5	17
6	17
6	17
B	20
J	12
X	13
9	17

566BJX9

Acierto 0/7 (0%)

SAE5640

Coche 13

Imagen matrícula extraída

Carácter	SSE
5	17
1	8
6	17
B	20
J	12
1	8
B	20

Acierto 0/7 (0%)

SAE5640

Coche 14

Imagen matrícula extraída

Carácter	SSE
8	17
Q	17
E	18
9	17
1	8
R	18
B	20

8QE91RB
Acierto 1/7 (14%)

SAE5640

Coche 15

Imagen matrícula extraída

Carácter	SSE
5	17
Q	17
7	12
8	17
O	12
1	8
G	15

5Q78O1G

Acierto 0/7 (0%)

SAE5640**Coche 16**

Imagen matrícula extraída

Carácter	SSE
5	17
A	16
9	17
B	20
J	12
R	18
0	16

5A9BJR0

Acierto 2/7 (28%)

SAE5640**Coche 17**

Imagen matrícula extraída

Carácter	SSE
R	18
9	17
E	18
9	17
5	17
X	13
B	20

R9E95XB

Acierto 1/7 (14%)

SAE5640

Coche 18

Imagen matrícula extraída

Carácter	SSE
8	17
1	8
2	14
7	12
0	16
U	15
X	13
81270UX	

Acierto 2/7 (28%)

SAI7084

Coche 19

Imagen matrícula extraída

Carácter	SSE
3	15
J	12
B	20
7	12
D	18
9	17
R	18
3JB7D9R	

Acierto 1/7 (14%)

SAI7084

Coche 20

Imagen matrícula extraída

Carácter	SSE
U	15
6	17
D	18
7	12
G	15
9	17
1	8
U6D7G91	

Acierto 0/7 (0%)

SAI7084

Coche 21

Imagen matrícula extraída

Carácter	SSE
5	17
3	15
D	18
0	16
B	20
B	20
B	20

53D0BBB

Acierto 5/7 (71%)

■5350BJB**Coche 22**

Imagen matrícula extraída

Carácter	SSE
A	16
2	14
3	15
9	17
4	12
U	15
L	11

A2394UL

Acierto 6/7 (85%)

■M2394UL**Coche 23**

Imagen matrícula extraída

Carácter	SSE
5	17
R	18
0	16
G	15
C	13
J	12
R	18

5R0GCJR

Acierto 4/7 (57%)

5406CWR

Coche 24



Imagen matrícula extraída

Carácter	SSE
5	17
6	17
D	18
7	12
G	15
L	11
A	16

56D7GLA

Acierto 5/7 (71%)

5657GLA

Coche 25



Imagen matrícula extraída

Carácter	SSE
C	13
C	13
2	14
2	14
2	14
9	17
U	15

CC2229U

Acierto 6/7 (85%)

CC2249U

Algunos de estos casos han servido para la localización de la matrícula, teniendo diferentes posiciones en las que se encontraban las matrículas o diferentes posicionamientos u orientaciones de los coches, lo que hace que la localización pueda ser más dinámica y no tenga sólo un caso base para reconocer.

No obstante, eso dificulta la obtención de los caracteres, con lo que las imágenes individuales no siempre salen limpias de ruido y esto hace que las entradas a la red sean complejas para poder clasificar y obtener un resultado óptimo.

Como se puede observar en los resultados, el porcentaje de acierto depende mucho de la localización y extracción de los caracteres. Y esto hace ver que el sistema rinda bien o no.

La siguiente tabla muestra mejor los porcentajes de acierto:

Nombre	Porcentaje de acierto	Nombre	Porcentaje de acierto	Nombre	Porcentaje de acierto
Coche 1	85%	Coche 11	14%	Coche 21	71%
Coche 2	43%	Coche 12	0%	Coche 22	85%
Coche 3	85%	Coche 13	0%	Coche 23	57%
Coche 4	37%	Coche 14	14%	Coche 24	71%
Coche 5	43%	Coche 15	0%	Coche 25	85%
Coche 6	43%	Coche 16	28%		
Coche 7	85%	Coche 17	14%		
Coche 8	43%	Coche 18	28%		
Coche 9	43%	Coche 19	14%		
Coche 10	0%	Coche 20	0%		

Como se puede observar, 8 de los 25 resultados son muy buenos ya que tienen un alto porcentaje de acierto. Los resultados aceptables, 8 de 25, se han obtenido en esta base de datos siendo la misma cantidad que los resultados buenos. Y 9 de los 25 no han dado el resultado esperado con un acierto bajo o nulo.

Hay que tener en cuenta que esta base de datos es más útil para la localización y reconocimiento de las matrículas en las imágenes de los coches, por ello se ha hecho uso de la misma con ese fin. No obstante, para comprobar que el funcionamiento del sistema es mejor del obtenido con esta base de datos, se ha usado otra base de datos que contiene imágenes como esta:



6. Conclusiones y Futuras mejoras

Aunque los resultados han sido bastante diversos y puesto que no existe una base de datos de libre acceso lo suficientemente grande como para que la red neuronal pudiese entrenar y ajustarse lo suficiente para dar unos datos fiables a un alto porcentaje, se han obtenido resultados aceptables en muchos casos.

Siendo la parte de visión artificial, por supuesto mejorable, ya que teniendo muchos casos de análisis se podría realizar una generalización del método y esto haría que los resultados de los caracteres reconocidos tuvieran una mejor calidad para servir como entradas a la red neuronal artificial.

También se podría intentar mejorar la configuración de la red neuronal probando con otras funciones de transferencia u otro algoritmo de entrenamiento. Incluso se podría implementar otra metodología de entrenamiento.

Como conclusión principal se puede decir que es un sistema bastante complejo. Combinar visión artificial con técnicas de clasificación mediante redes neuronales puede llegar a ser muy útil en sistemas en los que se necesite una alta complejidad computacional. Además, se ha hecho uso de la mayoría de los algoritmos estudiados con lo que se ha obtenido un alto conocimiento y manejo para la resolución de posteriores problemas que se puedan presentar o de mejorar los que ya existen.

Bibliografía

- [1] Vélez Serrano, José Francisco et al. (2003). *Visión por Computador*. Ed.Dykinson. Madrid. Última visita: Enero, 2012.
Enlace: <http://www.escet.urjc.es/%7EVisionc/VisionPorComputador.pdf>
- [2] Carrasco Zambrano, Manuel (2003). *Segmentación de Fallas en Soldaduras utilizando Técnicas de Procesamiento Digital de Imágenes*. Tesis de Grado. Universidad de Santiago de Chile Facultad de Ingeniería. Última visita: Enero, 2012.
Enlace: http://www.vizzion.cl/files/MasterThesis_carrasco.pdf
- [3] Valverde Rebaza, Jorge. *Detección de bordes mediante el Algoritmo de Canny*. Escuela Académico Profesional de Informática. Universidad Nacional de Trujillo. Última visita: Enero, 2012.
Enlace: <http://www.seccperu.org/files/DeteccióndeBordes-Canny.pdf>
- [4] Wikipedia. *Método del valor umbral*. Última visita: Enero, 2012.
Enlace: http://es.wikipedia.org/wiki/M%C3%A9todo_del_valor_umbral#M.C3.A9todo_de_Otsu
- [5] Wikipedia. *Redes neuronales artificiales*. Última visita: Enero, 2012.
Enlace: http://es.wikipedia.org/wiki/Red_neuronal_artificial
- [6] Mathworks. *Neural Networks Toolbox*. Última visita: Enero, 2012.
Enlace: <http://www.mathworks.es/help/toolbox/nnet/index.html>
- [7] Mathworks. *Neural Networks Toolbox*. Última visita: Enero, 2012.
Enlace: <http://www.mathworks.es/products/image/index.html>
- [8] Wikipedia. *Correlación*. Última visita: Enero, 2012.
Enlace: <http://es.wikipedia.org/wiki/Correlaci%C3%B3n>