



Reconocedor de Matrículas

Luis José Quintana Bolaño
Javier Osuna Herrera

Grupo Gayumbos
Reconocimiento de Patrones
Grado en Ingeniería Informática

Curso 2013/2014

Índice

1. Técnica de extracción	1
2. Técnica de clasificación	2
3. Tasa de aciertos	3
4. Método de validación	3
5. Implementación	4
6. Referencias	6
A. Comentarios y observaciones	7

Resumen

Este documento explica como se ha realizado el reconocedor de matrículas del grupo Gayumbos, y las técnicas de extracción, clasificación y validación que se han usado para su implementación. Finalmente, incluye un comentario sobre el resultado.

1. Técnica de extracción

En este apartado se van a detallar las técnicas que se han usado para reconocer la matrícula de un vehículo, y recortar cada letra o número de la misma. Primero es necesario recortar la matrícula del vehículo, que hemos realizado siguiendo la siguiente serie de pasos:

- Se ha convertido la imagen de rgb a hsv, para facilitar el aumento de la saturación del color del vehículo.
- Se pasa la imagen resultante a blanco y negro.
- Para detectar la matrícula, se hace una búsqueda de componentes conexos, particularmente cajas rectangulares.
- Por último, se intenta seleccionar aquellos rectángulos que tengan la razón de proporcionalidad de una matrícula. Si encuentra más de un rectángulo con proporciones adecuadas, detecta cual es la matricula por el contraste de la suma de las filas.

Este método no funciona con una precisión de un 100 %, ya que algunas fotos no responden a los filtros como es deseable, generalmente debido a obstáculos visuales en la matrícula, como suciedad u objetos extraños. Para solucionar esto se ha añadido la opción de rechazar el resultado automático y recortar la imagen manualmente.



Figura 1: Elección de matricula automatica o manual.

Una vez que se ha detectado la matrícula, es necesario extraer de la misma cada letra y número. Para recortarlas, se han seguido pasos muy similares al proceso de recortar la matrícula, exceptuando el primero.

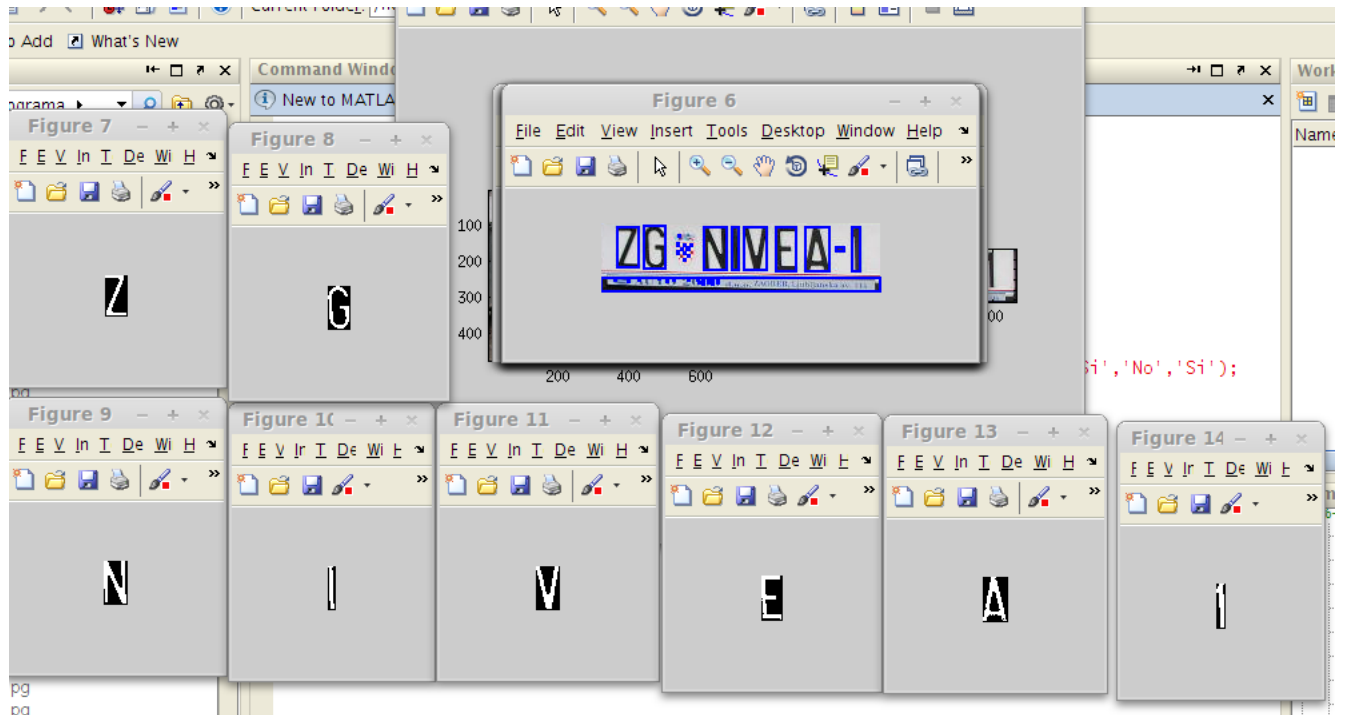


Figura 2: Letras y números recortados.

2. Técnica de clasificación

Para clasificar las letras y números de nuestra matrícula elegimos como característica, inspirados por la lectura de un documento sobre OCR mediante redes neuronales [1], la sucesión de espacios blancos y negros que presentan horizontalmente. Para ello desarrollamos nuestro propio método de codificación:

- Se redimensionan los números y las letras al mismo tamaño, en nuestro caso a 20x10.
- Se transforma cada imagen en 4 vectores, resultado de dividir la imagen en cuatro y sumar las filas de cada grupo.
- Una vez obtenidas estas, se analiza el patrón de secciones de color negro y blanco de cada vector y se codifica con un número único para facilitar la comprobación de los patrones. El código se obtiene sumando los valores de la siguiente tabla:

Posición	Elemento			
	B	N	b	n
1	2	4	6	8
2	3	6	9	12
3	5	10	15	20
4	7	14	21	28

Donde B equivale a una sección blanca larga (más de 3 píxeles), N a una sección negra larga y b y n a una sección blanca o negra corta, respectivamente.

P. Ej.: El patrón $BNbn$ sería $tabla(B, 1) + tabla(N, 2) + tabla(b, 3) + tabla(n, 4) = 2 + 6 + 15 + 28 = 51$

- A continuación se obtiene el código final realizando $P_1 + (100 * P_2) + (10000 * P_3) + (1000000 * P_4)$, donde P_1, \dots, P_4 son los códigos de cada vector individual respectivamente.

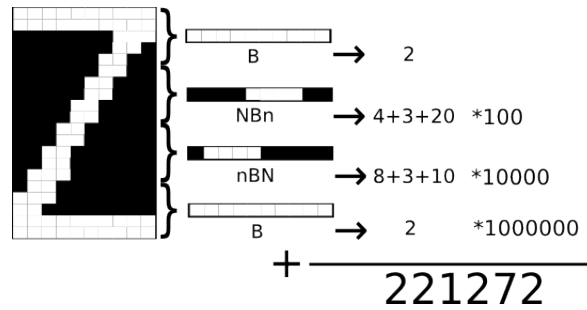


Figura 3: Esquema del proceso de codificación.

3. Tasa de aciertos

- Localización y recorte automático correcto de matriculas: 53 % del conjunto total.
- Clasificación automática de los caracteres en el entrenamiento: sobre el 50 %
- Correcta lectura de la matricula completa sobre el conjunto de prueba:
 - 41 % de aciertos plenos
 - 29,5 % de fallos de un solo carácter.
 - 29,5 % de fallos en más de un carácter.

4. Método de validación

Se ha usado un método de validación cruzada dividiendo el conjunto de datos en 10 grupos, por los siguientes motivos:

- Principalmente, si usamos otro método de validación, como la validación simple, puede ocurrir que al entrenar no se obtenga alguna letra o número que necesitamos para luego validar, y salga un resultado erróneo. Con esta técnica, nos aseguramos de que entrenará con todos los datos.
- Es más eficiente si se tiene un conjunto pequeño de imágenes a validar.

5. Implementación

En esta sección se van a poner las dos funciones más importantes de la practica, que son “entrena.m” y “reconoce.m”. El código tiene funciones auxiliares que ayudan a estas cumplir su cometido. Todas las funciones están dentro de la carpeta principal de la práctica.

entrena.m

```
%
%
% Función que utiliza imágenes de coches para recortar la matricula y realizar
% el entrenamiento del reconocedor de letras, creando un archivo de Plantillas.
% Recibe la ruta de la carpeta en la que están las imágenes, terminada en /".
%
function Plantillas = entrena (ruta)
    listafotos = dir(ruta);
    Totales = zeros(1,35);
    for i=3:size(listafotos,1) % Cargando y clasificando todas las fotos
        fnombre = listafotos(i).name
        I = imread([ruta fnombre]);
        [M, T] = obtenerMatricula(I);
        C = obtenerCaracter(M);
        close all;
        for k=1:length(C)
            Cn(:,k)=imresize(C{k}, [20 10]);
        end
        for j = 1 : size(Cn,3)
            aux = clasificaCaracter(Cn(:,j));
            Totales(aux)=Totales(aux)+1;
            Caracteres(:,aux, Totales(aux))= Cn(:,j);
        end
    end
    Plantillas = generaPlantilla(Caracteres,Totales); %Generando plantillas
    save('Plantillas.mat','Plantillas');
    close all;
end

%
% Recibe la imagen de un caracter y devuelve un número que corresponde a la
% posición del número/letra en la lista "1,...,9,A,...,Z,0".
%
function Num = clasificaCaracter( C )
    P1 = zeros(1,10); P2 = zeros(1,10); P3 = zeros(1,10); P4 = zeros(1,10);
    for i=1:5 %Transforma la imagen en 4 vectores
        P1=P1 + C(i, :);
        P2=P2 + C(i+5, :);
        P3=P3 + C(i+10, :);
        P4=P4 + C(i+15, :);
    end
    %Codifica los 4 vectores a ún solo número único
    T1=codifica(P1); T2=codifica(P2); T3=codifica(P3); T4=codifica(P4);
    Cod=T1+(100*T2)+(10000*T3)+(1000000*T4)
```

```

    Num=compara(Cod); %Encuentra a qué caracter corresponde la codificación
end

%
% Recibe una matriz de caracteres ordenados y un vector con la cantidad total
% de muestras de cada caracter y genera las plantillas haciendo la media
% de cada caracter.
%
function Plantillas= generaPlantilla(Caracteres,Totales);
    for i=1:size(Caracteres,3);
        aux=zeros(20,10);
        for j = 1 :Totales(i); %Suma de todas las muestras de cada caracter
            aux = aux + Caracteres(:,i, j);
        end
        for j=1:size(aux,1) %Hace la media de cada caracter
            for k=1:size(aux,2)
                if aux(j,k)/Totales(i)>0.5
                    aux(j,k)=1;
                else
                    aux(j,k)=0;
                end
            end
        end
        Plantillas(:,i)=aux; %Guarda la plantilla de cada caracter
    end
end

```



```

%
%
% Función que recibe la ruta al archivo de imagen de un coche y reconoce
% la matrícula, devolviendo el texto de la misma.
%
function Mtext = reconoce (Inom)
    I = imread(Inom);
    [M, T] = obtenerMatricula(I); %Recorte de la matricula
    figure, imshow(M);
    C = obtenerCaracter(M); %Recorte de los caracteres
    Mtext='';
    for k=1:length(C) %Reconocimiento de cada caracter
        Mtext=[ Mtext ' ' reconocerCaracter(imresize(C{k}, [20 10]))];
    end
    Mtext
    pause;
    close all;
end

%
% Función que recibe la imagen de un caracter y lo reconoce comparandolo con
% las plantillas generadas en el entrenamiento, devuelviendolo en forma de texto
%
function T = reconocerCaracter(C)
    load('Plantillas.mat'); %Carga el archivo de plantillas del entrenamiento
    comp=[ ];
    for n=1:length(Plantillas) %Compara la imagen con cada plantilla
        sem=corr2(Plantillas (:,:, n),C);
        comp=[comp sem];
    end
    N=find(comp==max(comp)); %Encuentra la posición de la más parecida
    T=encuentra(N); %Devuelve el caracter correspondiente a la posición
end

```

6. Referencias

Referencias

- [1] N. Vázquez, M. Nakano, and H. Pérez-Meana. Automatic system for localization and recognition of vehicle plate numbers. *Journal of Applied Research and Technology*, 1(01), December 2009.

A. Comentarios y observaciones

El método de clasificación elegido estaba pensado para simplificar la creación a mano de los códigos que representarían cada número en las condiciones de la función “clasifica”, pero en la realidad las matrículas resultaron tan irregulares e imperfectas que el proceso por el cual extraíamos los caracteres no pudo ser afinado lo suficiente para aportar la homogeneidad necesaria a las figuras extraídas, por lo que en algunos casos terminamos necesitando una cantidad considerable de variaciones del código que identificaba a cada letra, para tener en cuenta las distintas condiciones en las que llegaban al clasificador.

Con un método de extracción de caracteres más preciso y mejores filtros, o con unas fotos más claras, el resultado habría sido más sencillo y limpio.

Debido a esto, con los resultados reales el proceso automático de entrenamiento solo clasifica correctamente sobre el 50 % de los caracteres, pero esto no afecta al reconocedor si se entrena con un número suficiente de datos .