

PROYECTO DE IMAGEN: RECONOCIMIENTO DE CARACTERES

Grupo 06

Componentes: Alexis Hermida, Noa Cereixo, André Costas, José Luis Rodríguez



ÍNDICE

Introducción	3
Banco de datos	3
Implementación	3
Tarea 1: Binarización	4
Función <i>binarizacion</i>	4
Tarea 2: Segmentación de filas	5
Función <i>numFilas</i>	5
Tarea 3: Segmentación de caracteres	6
Función <i>numCaracteres</i>	6
Tarea 4: Procesamiento de caracteres	6
Función <i>procesCaract</i>	7
Tarea 5: Creación del alfabeto	7
Función <i>alfabeto</i>	8
Tarea 6: Reconocimiento de caracteres	8
Función <i>reconocimiento</i>	8
Función <i>espacio</i>	9
Resultados	10
Análisis de errores	10
Posibles aplicaciones	10
Anexo	11



INTRODUCCIÓN

Este proyecto tiene como objetivo el reconocimiento de distintos caracteres presentes en imágenes de formato PNG. Este reconocimiento permite, tanto la lectura del contenido de dichas imágenes, como su escritura en diferentes documentos de texto. Para ello, se debe facilitar un alfabeto con los caracteres de referencia que permitan identificar los hallados en las imágenes. Se debe tener en cuenta que, para garantizar un correcto funcionamiento del programa, este alfabeto se debe corresponder, tanto en contenido como en orden, con el alfabeto inglés.

Para su desarrollo, se han implementado las seis tareas propuestas por el profesorado de la asignatura (binarización, segmentación de filas, segmentación de caracteres, procesamiento de caracteres, creación del alfabeto y reconocimiento de caracteres), con exitosos resultados.

BANCO DE DATOS

Para la evaluación del correcto funcionamiento de este programa, se ha generado un banco de datos que incluye un total de cinco archivos de imagen, entre los que se encuentran imágenes en color e imágenes desenfocadas. Es posible acceder a estos archivos, haciendo clic [aquí](#).

Además, también se ha generado un alfabeto de referencia que permitirá la evaluación del objetivo final de este proyecto: el reconocimiento de caracteres. Es posible acceder a este alfabeto, haciendo clic [aquí](#).

IMPLEMENTACIÓN

El código de este proyecto se ha desarrollado en un documento Live Script, dividido en seis secciones, que se corresponden con las seis tareas propuestas por el profesorado de la asignatura. A continuación, se detallan los procesos empleados en la implementación de estas tareas, así como los de las funciones



creadas en cada una de ellas.

TAREA 1: BINARIZACIÓN

En esta primera tarea, se leen y binarizan las imágenes contenidas en los archivos del banco de datos. Tras este último proceso, el fondo de la imagen se visualiza en negro (valor 0 en la matriz) y los caracteres en blanco (valor 1 en la matriz). Es posible observar las imágenes obtenidas en este proceso a partir de las imágenes del banco de datos haciendo clic [aquí](#).

Código:

```
im=imread("nombredelarchivo"); %instrucción 1  
imshow(im) %instrucción 2  
im=binarizacion(im); %instrucción 3
```

- instrucción 1: mediante la función *imread* se obtiene la matriz de la imagen contenida en el archivo *nombreachivo* y se almacena en *im*.
- instrucción 2: mediante la función *imshow* mostramos por pantalla la imagen original.
- instrucción 3: mediante la función *binarizacion* obtenemos una matriz, a partir de la matriz dada en la instrucción 1, que toma valores 0 (negro) en el fondo y 1 (blanco) en los caracteres. Se almacena la matriz obtenida en *im*.

Función binarización

Esta función devuelve la imagen original como una imagen de fondo negro (valor 0 en la matriz) y caracteres blanco (valor 1). Además, muestra el resultado por pantalla.

Para obtener la imagen binarizada, convierte la imagen original en una imagen en escala de grises, le aplica una máscara de desenfoque y la binariza. Posteriormente, resta una unidad a todos los elementos de la matriz y multiplica el resultado por -1, intercambiando unos y ceros. Por último, se elimina el margen surgido al aplicar la máscara de desenfoque.

Signatura: `im = binarizacion (im)`

Código: disponible haciendo clic [aquí](#).



TAREA 2: SEGMENTACIÓN DE FILAS

Mediante la segmentación de filas, se halla el número de filas de la imagen original, así como los índices de inicio y fin de cada fila dentro de la matriz de la imagen.

Código:

```
[filas,indices]=numFilas(im,'Titulo'); %instrucción 1
```

- instrucción 1: para hallar el número de filas de la imagen y sus índices se emplea la función *numFilas* y se almacenan estos datos en *filas* e *indices*, respectivamente.

Función numFilas

Esta función devuelve el número total de filas presentes en la imagen, así como los índices de inicio y fin de cada fila dentro de la matriz de la imagen. Además, si el parámetro *titulo* es distinto de *null*, representará el vector calculado a partir de la suma de las columnas de la matriz de la imagen, así como del vector que toma valor 1 si forma parte de una fila y valor 0 en caso contrario. Es posible observar las gráficas obtenidas a partir de los archivos del banco de datos, haciendo clic [aquí](#).

Para detectar las filas, se normaliza el vector calculado a partir de la suma de las columnas y se multiplica por 10. De esta forma, se puede fijar, para cualquier vector, un umbral de valor 1.1. Si un elemento determinado del vector supera este umbral, se considera que forma parte de una fila.

Signatura: [filas, indices] = numFilas (im, titulo)

Código: disponible haciendo clic [aquí](#).

TAREA 3: SEGMENTACIÓN DE CARACTERES

De forma análoga a la segmentación de filas, en esta tarea se halla el número de caracteres por fila de la imagen original, así como los índices de inicio y fin de cada carácter dentro de la matriz de la imagen.

Código:

```
[caracteres,indCar]=numCaracteres(im,filas,indices,'Titulo')  
%instrucción 1
```

- instrucción 1: para hallar el número de caracteres por fila de la imagen y sus índices se emplea la función *numCaracteres* y se almacenan estos datos en *caracteres* e *indCar*, respectivamente.

Función numCaracteres

Esta función devuelve el número total de caracteres presentes en cada fila de la imagen, así como los índices de inicio y fin de cada uno de estos caracteres dentro de la matriz de la imagen. Además, si el parámetro *titulo* es distinto de *null*, muestra por pantalla tantas gráficas como filas presente la imagen. Cada una de ellas, contiene la representación del vector calculado a partir de la suma de las filas de la matriz de una fila determinada. Es posible observar las gráficas obtenidas a partir de los archivos del banco de datos, haciendo clic [aquí](#).

Para detectar los caracteres, se buscan los elementos de los vectores calculados que cuyo valor es mayor que 0 y se considera que estos elementos forman parte de un carácter.

Signatura: [caracteres, indCaracteres] = numCaracteres (im, filas, indFilas, titulo)

Código: disponible haciendo clic [aquí](#).

TAREA 4: PROCESAMIENTO DE CARACTERES

En esta tarea se busca almacenar las matrices que contienen los distintos caracteres de la imagen. Aunque mayoritariamente se corresponden con submatrices de la matriz original, pueden



alterarse debido a la necesidad de que estas submatrices sean cuadradas. Por tanto, en caso de que alguna de ellas no cumpla esta condición, se rellena con ceros (negro)

Código:

```
carProces=procesCaract(im,filas,indices,caracteres,indCar)  
%instrucción 1
```

- instrucción 1: se almacenan en *carProces* las matrices que contienen los distintos caracteres de la imagen, obtenidos por la función *procesCaract*.

Función procesCaract

Esta función devuelve las matrices (cuadradas) que contienen los caracteres de la imagen.

Para ello, empleando los índices obtenidos en las funciones anteriores, se localiza el carácter en la matriz. Si la submatriz localizada es cuadrada se almacena en el parámetro *caracteres*. En caso contrario, se rellena con ceros hasta igualar el número de filas y columnas y se almacena el resultado en *caracteres*.

Signatura: [caracteres] = procesCaract (im, filas, indFilas, caract, indCaract)

Código: disponible haciendo clic [aquí](#).

TAREA 5: CREACIÓN DEL ALFABETO

Mediante procesos semejantes a los descritos anteriormente, se obtienen las matrices correspondientes a los caracteres de referencia en la tarea final. Se debe tener en cuenta que la imagen de la que se obtienen estos caracteres debe contener el alfabeto inglés y que las matrices resultantes, al igual que en la tarea anterior, deben presentar el mismo número de filas que de columnas (cuadradas).

Código:

```
alf=alfabeto('alfabeto.png') %instrucción 1
```



- instrucion 1: se almacenen las matrices con los caracteres de referencia obtenidas por la función *alfabeto* y se almacenan en *alf*.

Función alfabeto

Esta función devuelve las matrices cuadradas que contienen los caracteres de referencia.

Para ello, calcula tanto el número de filas y sus índices de inicio y fin, como el número de caracteres por fila y sus correspondientes índices. Posteriormente, procesa estos caracteres de referencia y los almacena en el parámetro *alfabeto*.

Signatura: `alfabeto = alfabeto (alf)`

Código: disponible haciendo *clic* [aquí](#).

TAREA 6: RECONOCIMIENTO DE CARACTERES

El reconocimiento de caracteres consiste en la asociación de cada carácter presente en la imagen original con un carácter de la imagen que contiene el alfabeto, lo que facilita su identificación y su transcripción a un archivo de texto.

Código:

```
f=fopen("nombreadarchivodetexto", 'wt'); %instrucion 1
fprintf (f, reconocimiento (alf, craProces, indCar,
caracteres)); %instrucion 2
fclose(f); %instrucion 3
```

- instrucion 1: abre el archivo de texto sobre el que se desea escribir con el tipo de acceso *wt* (escritura).
- instrucion 2: escribe sobre el archivo abierto el *String* devuelto por la función *reconocimiento*.
- instrucion 3: cierra el archivo de texto.

Función reconocimiento

Esta función identifica los caracteres presentes en las imágenes, a partir de sus matrices, tomando como referencia los caracteres



del alfabeto facilitado. Además, devuelve un String con los caracteres identificados introduciendo los espacios y saltos de línea necesarios.

En la identificación de un carácter de la imagen, este se compara con todos los caracteres de referencia y se asocia al más semejante. Para determinar esta semejanza, se halla el valor absoluto de la resta de la matriz del carácter seleccionado y la matriz de cada uno de los caracteres de referencia y se selecciona el carácter de referencia con el que se obtuvo un menor valor absoluto. A continuación, se determina si debe suceder al carácter un espacio o un salto de línea mediante la función *espacio* y se almacena en el parámetro *texto*. Se repite esta operación para todos los caracteres presentes en la imagen.

Signatura: `texto = reconocimiento (alfabeto, caracteres, indices, numCaract)`

Código: disponible haciendo clic [aquí](#).

Función espacio

Esta función devuelve los valores 0, si no se precisa de espacios ni saltos de línea tras el carácter identificado, 1 o 2, si un espacio o una salto de línea sucede a dicho carácter, respectivamente.

Se considera que se precisa de un salto de línea si se trata del último carácter de una fila, siempre y cuando, esta fila no sea la última.

Se considera que se precisa de un espacio cuando la diferencia entre el índice de fin del carácter seleccionado y el índice de inicio del carácter siguiente es, al menos, 45 (último carácter de una palabra).

En otro caso, no precisa de espacio ni salto de línea.

Signatura: `esp = espacio (pos, numCaract, indices)`

Código: disponible haciendo clic [aquí](#).



RESULTADOS

Los resultados obtenidos a partir de los archivos del banco de datos coinciden con los esperados. Es posible consultar la imagen de entrada junto con el archivo de texto de salida correspondiente, haciendo clic en los siguientes enlaces:



[loqueelviento](#)
[elpadrino](#)
[casablanca](#)
[elimperiocontraataca](#)
[apocalypsenow](#)

ANÁLISIS DE ERRORES

Como se muestra en el apartado *Resultados*, todas las salidas del programa coinciden con las esperadas y, por tanto, no se han detectado errores.

POSIBLES APLICACIONES

En nuestro día a día encontramos múltiples aplicaciones del reconocimiento de caracteres en archivos de imagen. Entre ellas, destacan algunas de las implementadas por Google, y que tenemos disponibles en nuestros dispositivos móviles como la traducción de textos presentes en libros, carteles..., búsquedas en Internet a partir de fotografías... También encontramos ejemplos más sofisticados como puede ser la lectura de la matrícula de los vehículos, entre otras.



ANEXO

```
im1=imread("loqueelviento.png");
imshow(im1)
im1=binarizacion(im1);

im2=imread("elpadrino.png");
imshow(im2)
im2=binarizacion(im2);

im3=imread("casablanca.png");
imshow(im3)
im3=binarizacion(im3);

im4=imread("elimperiocontraataca.png");
imshow(im4)
im4=binarizacion(im4);

im5=imread("apocalypsenow.png");
imshow(im5)
im5=binarizacion(im5);

[filas1,indices1]=numFilas(im1,'Lo que el viento se llevó');

[filas2,indices2]=numFilas(im2,'El padrino');

[filas3,indices3]=numFilas(im3,'Casablanca');

[filas4,indices4]=numFilas(im4,'El imperio contraataca');

[filas5,indices5]=numFilas(im5,'Apocalypse now');

[caracteres1,indCar1]=numCaracteres(im1,filas1,indices1,'Lo que el
viento se llevó');

[caracteres2,indCar2]=numCaracteres(im2,filas2,indices2,'El
padrino');

[caracteres3,indCar3]=numCaracteres(im3,filas3,indices3,'Casablanc
a');
```



```
[caracteres4,indCar4]=numCaracteres(im4,filas4,indices4,'El
imperio contraataca');

[caracteres5,indCar5]=numCaracteres(im5,filas5,indices5,'Apocalyps
e now');

carProces1=procesCaract(im1,filas1,indices1,caracteres1,indCar1);

carProces2=procesCaract(im2,filas2,indices2,caracteres2,indCar2);

carProces3=procesCaract(im3,filas3,indices3,caracteres3,indCar3);

carProces4=procesCaract(im4,filas4,indices4,caracteres4,indCar4);

carProces5=procesCaract(im5,filas5,indices5,caracteres5,indCar5);

alf=alfabeto('alfabeto.png');

f1=fopen("loqueelviento.txt",'wt');
fprintf(f1, reconocimiento(alf,carProces1,indCar1,caracteres1));
fclose(f1);

f2=fopen("elpadrino.txt",'wt');
fprintf(f2, reconocimiento(alf,carProces2,indCar2,caracteres2));
fclose(f2);

f3=fopen("casablanca.txt",'wt');
fprintf(f3, reconocimiento(alf,carProces3,indCar3,caracteres3));
fclose(f3);

f4=fopen("elimperiocontraataca.txt",'wt');
fprintf(f4, reconocimiento(alf,carProces4,indCar4,caracteres4));
fclose(f4);

f5=fopen("apocalypsenow.txt",'wt');
fprintf(f5, reconocimiento(alf,carProces5,indCar5,caracteres5));
fclose(f5);
```



