

UNIVERSIDAD DE CASTILLA - LA MANCHA

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS INDUSTRIALES

INGENIERÍA ELÉCTRICA, ELECTRÓNICA
AUTOMÁTICA Y COMUNICACIONES

Ricardo Camacho Díaz-Cano
Pedro Antonio Estévez Pérez
Alberto García-Fogeda Gómez

SUDOKU

Área de Conocimiento

Visión por Computador

Departamento

Ingeniería Eléctrica, Electrónica,
Automática y Comunicaciones

Escuela Técnica Superior de Ingenieros
Industriales.

Universidad de Castilla - La Mancha

Índice

1. Introducción y preparación	2
2. Carga de la imagen	3
3. Preprocesado de la imagen	4
4. Identificación del sudoku	6
5. Resolución del Sudoku	7

Índice de figuras

1. Imagen inicial.	2
2. Imagen que se introducirá en Matlab.	2
3. Métodos de introducción de la imagen.	3
4. Proceso 1.	4
5. Proceso 2.	4
6. Proceso 3.	5
7. Sudoku identificado por Matlab.	6
8. Sudoku resuelto.	7

1. Introducción y preparación

El objetivo de este proyecto es crear un programa utilizando Matlab que permita reconocer, identificar y resolver un sudoku a partir de una imagen proporcionada por diversos medios.

A lo largo de este documento, se usará como ejemplo la siguiente imagen, en la que se ha impreso un sudoku sobre el cual se han ido realizando las pruebas.

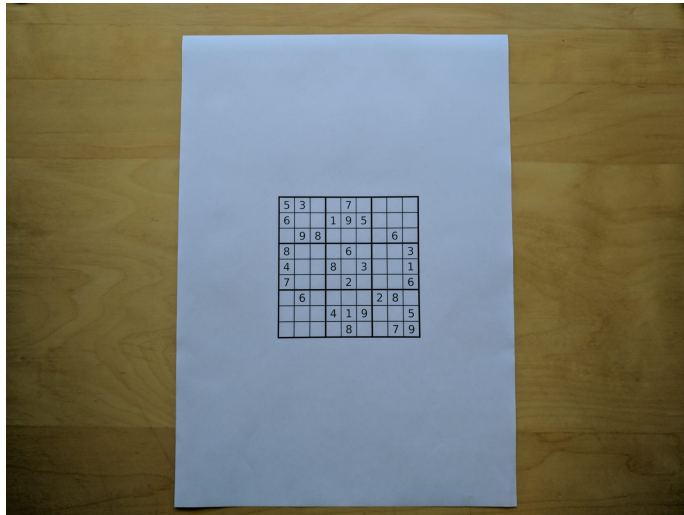


Figura 1: Imagen inicial.

No es necesario realizar una foto del sudoku de una forma muy específica. Basta con encuadrar bien la imagen y que haya luz. No debería haber mayor problema con las cámaras actuales de los smartphones. En este caso, la imagen que se utilizará como ejemplo a lo largo de este documento es la siguiente:

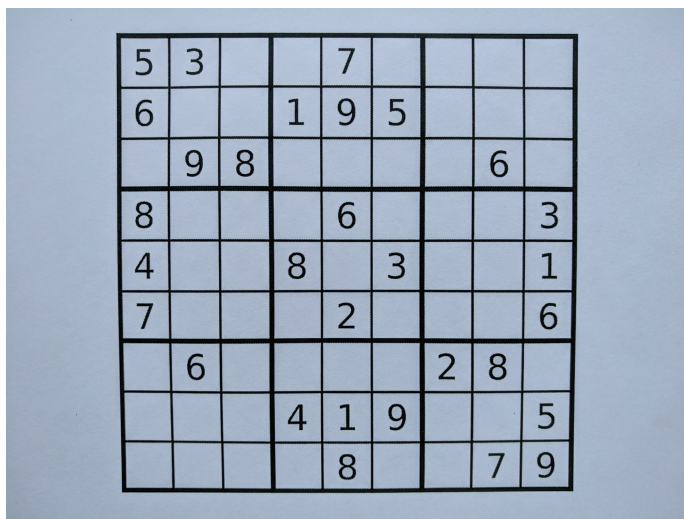


Figura 2: Imagen que se introducirá en Matlab.

2. Carga de la imagen

A la hora de cargar la imagen en Matlab, se le proporcionarán tres métodos al usuario.

- Introducir manualmente la imagen.
- Realizar una foto con la webcam.
- Realizar una foto usando el smartphone como webcam.

El primer método es trivial, mientras que el segundo poco ortodoxo debido a la dificultad a la hora de realizar la foto con comodidad así como por la baja calidad de las webcam de los portátiles.

Así pues, el método más interesante es el tercero. Para llevarlo a cabo, se ha de instalar en el smartphone una app que permita usar la webcam del móvil mediante una IP. Después de realizar diversas pruebas, se recomiendan las siguientes aplicaciones, siendo la primera la utilizada durante las pruebas.

- DroidCam
- IP Webcam

La configuración es muy simple. Una vez seguidos los pasos de la app, se proporcionará una dirección IP, que será la que hay que introducir en Matlab, como se ve en la siguiente imagen.

```
if (opc == 1)
    imag = webcamForm;
elseif (opc == 2)
    IP = 'http://192.168.1.172:4747/video';
    imag = mobileForm(IP);
else
    imag = manualForm;
end
```

Figura 3: Métodos de introducción de la imagen.

Tanto si se escoge esta opción como la de la webcam, en ambos casos Matlab mostrará por pantalla el vídeo que está siendo capturado por la webcam en ese momento y guardará la imagen en cuanto se presione cualquier botón del teclado.

3. Preprocesado de la imagen

Inicialmente, se le da la opción al usuario de seleccionar la región de interés en caso de que sea imposible realizar una fotografía "limpia" del sudoku. Posteriormente se convierte la imagen a escala de grises y se procede al preprocesamiento.

Primero se realiza un ajuste del contraste de la imagen, para detallar el contorno y los números. A continuación, se realiza una erosión y se reconstruye combinándolo con la imagen ajustada, dando como resultado una imagen donde se resalta tanto el contorno como los números y se realiza el binarizado de la imagen.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Figura 4: Proceso 1.

Se crea otra imagen binaria, que es el complemento de BW, donde la región fuera de la región de interés pasará a negro y el sudoku en blanco. Gracias a esto se consigue obtener las coordenadas donde se encuentra el sudoku dentro de la imagen original.

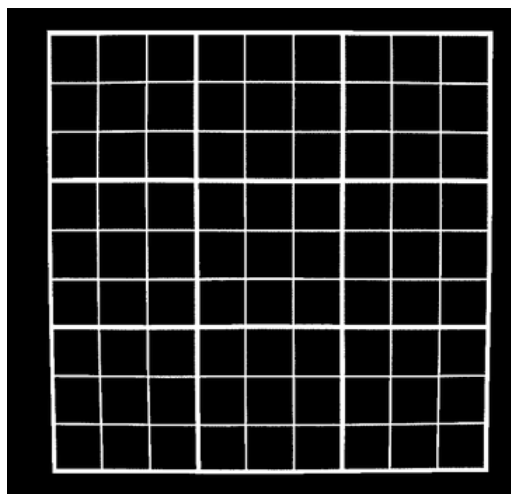


Figura 5: Proceso 2.

Una vez conocidas las coordenadas del sudoku dentro de la imagen, se procede a recortar las imágenes originales y restarlas, dando como resultado los números.

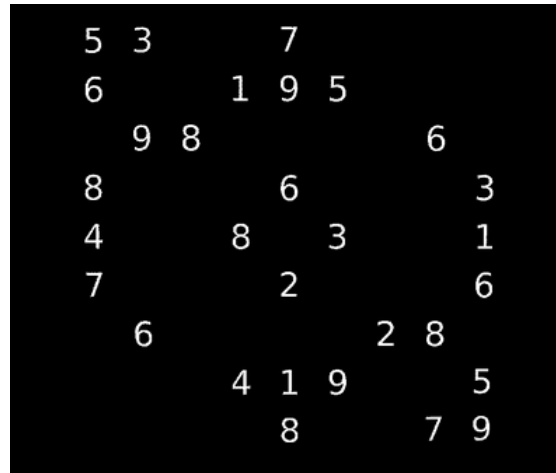


Figura 6: Proceso 3.

De forma alternativa, y ante el caso de que el programa sea incapaz de resolver el sudoku correctamente, existe la posibilidad de usar aplicaciones como CamScanner o similares. Se trata de aplicaciones de escaneo que realizan un preprocesamiento de la imagen de forma independiente, mejorando considerablemente la calidad de lectura de la imagen y permitiendo que Matlab sea capaz de distinguir casos más extremos.

4. Identificación del sudoku

Para resolver el sudoku, existe una función propia de Matlab llamada SudokuEngine. Esta función tiene como entrada una matriz en la que cada fila indica la posición del número y su valor.

Así pues, en este paso se calcularán los centroides de cada casilla así como el número de píxeles de cada una. A partir de aquí, el procedimiento consiste simplemente en ir recorriendo la imagen para almacenar en la matriz las posiciones y los valores de cada número.

Para identificar los números, se utiliza una función propia de Matlab llamada ocr, cuya función es precisamente la identificación de texto.

Una vez tenemos la matriz, podemos redibujar el sudoku usando la herramienta drawSudoku para comprobar que efectivamente, es el mismo que el original y que no ha habido ningún error durante la identificación del mismo.

5	3			7				
6			1	9	5			
	9	8					6	
8				6				3
4			8		3			1
7				2				6
	6					2	8	
			4	1	9			5
				8			7	9

Figura 7: Sudoku identificado por Matlab.

5. Resolución del Sudoku

Como se ha dicho en el apartado anterior, se llama a la función de Matlab y se le proporciona como argumento de entrada la matriz identificada. Ésta resuelve el problema, de forma que podemos representar el resultado.

5	3	4	6	7	8	9	1	2
6	7	2	1	9	5	3	4	8
1	9	8	3	4	2	5	6	7
8	5	9	7	6	1	4	2	3
4	2	6	8	5	3	7	9	1
7	1	3	9	2	4	8	5	6
9	6	1	5	3	7	2	8	4
2	8	7	4	1	9	6	3	5
3	4	5	2	8	6	1	7	9

Figura 8: Sudoku resuelto.