En esta sección se presentará el procedimiento y los pasos seguidos para la identificación del número.

Una vez que se han obtenido las casillas del sodoku, tal y como explica la sección anterior. Dicha función envía una estructura con todas las imágenes de las casillas.

Por su parte, la función de reconocimiento de números comienza haciendo un reconocimiento de las regiones de la imagen. De forma que si no se encontrase ninguna suficientemente grande se determina que es una casilla vacía, y si se encontrase alguna región suficientemente grande se reconocería como un número y se pasaría a otra función denominada "Num\_identification".

La función "Num\_identification" posee como entrada la propia imagen, una variable que indica si es necesario comprobar si la imagen está rotada o si se debe rotar la imagen y una varible que indica si se deben mostrar o no las imágenes resultantes del proceso. Y como salida, el número identificado y otra variable que indica si se detecta rotación en el número.

La función comienza por rotar la imagen si fuese necesario (90 ó 180º). Posteriormente mejora el contraste mediante la ecualización del histograma. La binariza mediante un umbral global, y la invierte para tener los números en blanco y el fondo en negro. Para reconocer el número se trabaja con el exoesqueleto, por ello la imagen no solo saca el exoesqueleto, sino que además realiza una operación de cerrado para evitar problemas con la continuidad del mismo. Por último, obtiene sus propiedades; siendo de vital importancia la localización de los puntos finales (se alamcenan como posición realtiva respecto del centroide), el número de puntos finales, el área convexa y el número de Euler (indicativo del número de huecos presentes en la imagen).

Una vez obtenidas dichas propiedades comienza con la identificación del número: En primer lugar los divide por el número de Euler; si tiene dos huecos es el número 8. Si tiene un solo hueco puede ser el 6 o el 9 en caso de poseer un solo punto final, o el 4 en caso de poseer varios. Para distinguir el 6 del 9 se emplea la localización de dicho punto final, ya que diferirán en signo. Y en caso de no tener hueco alguno podría ser el 5, si tuviese 4 puntos finales. El 2 o el 7 si tuviese 2; se separan ambos casos por el área, ya que el área del 2 es considerablemente más grande que el del 7. Y en caso de tener 3 puntos finales, separamos el 1 por su reducida área, el 5 por la localización del punto final y el 3 por descarte de los otros dos.

De esta forma se consigue obtener el número representado en la imagen. ¿Pero qué pasa si viene girado? Si el número viene tumbado basta con examinar su bounding box, pues será más ancha que alta. Y si el número está invertido se podrá detectar sólo en caso de ser el número 1, 2 ó 7.

El número uno, se puede detectar si está invertido observando su punto final, ya que la localización en 'X' respecto del centroide cambiará de signo. En el número 2 aumentará considerablemente su localización 'Y' respecto del punto final. Y en el 7, al igual que en el 1, cambiará el signo de la localización 'X' del punto final respecto al centroide. De esta forma, y gracias a uno de estos tres números, seremos capaces de averiguar si el número está invertido u orientado correctamente.

Una vez obtenida la respuesta de dicha función debemos estudiarla y guardarla. Pues si el número está tumbado habrá que avisar a la función en las iteraciones posteriores, para que no lo vuelva a comprobar y gire el número. Lo que no se puede determinar es el sentido de rotación del mismo, por lo que siempre lo girará en sentido horario y a continuación comprobará si el número está invertido, para saber si la rotación ha sido correcta.

De la misma forma, si la función detecta que un número está invertido significa que todos sus predecesores y todos sus sucesores lo estarán, por lo que se deberá indicar a la función que invierta el número en todas sus iteraciones posteriores y que de alguna forma revise todas las iteraciones pasadas, pues ciertos números al estar invertidos se podrían confundir con otros; en concreto el 5, el 6 y el 9 pueden dar problemas al estar invertidos, mientras el resto se deberían identificar adecuadamente.

Para realizar la revisión de las iteraciones pasadas se ha decidido ir creando una matriz de igual dimensión a la matriz que almacenará el Sudoku estructurado para su resolución. En dicha matriz se almacenará un cero en caso de que el número pueda ser problemático por estar invertido, y un uno en caso de no serlo. De forma que si detectamos una inversión de números solo deberemos comprobar aquellos que estén guardados como cero en dicha matriz.

Una vez terminado todo el proceso, se obtiene el Sudoku como una matriz de 9x9. En la cual el cero indicará que la casilla está vacía.

Por último, cabe mencionar que este código sólo contempla los números escritos por ordenador, ya que la dificultad y variedad de los números escritos manualmente se escapan a las posibilidades de este. En caso de querer detectar números manuales se ha pensado hacer uso de una red neuronal capaz de aprender las características comunes y poder identificarlos. En concreto, tras una búsqueda con este fin, se ha hallado una red neuronal preparada para la diferenciación de números y ya implementada sobre Matlab. Se trata de la herramienta OCR de reconocimiento de caracteres. Para poder emplear esta herramienta antes habría que entrenarla, por lo que se propone como avance futuro, ya que por falta de tiempo no se ha podido implementar en el presente proyecto.