|  |  |
| --- | --- |
| Name: Hiteshi Shah (hss7374) | Foundations of Computer Vision  Due: 03/18/2018 |

**HW10**

1. **Overview**

In this assignment, we are provided with images that have a Sudoku puzzle and our task is to draw a magenta box around the Sudoku.

2. **Approach Used**

For this assignment, I first convert the image to double and then into a binary image in order to perform morphological operations.

I check if the background is black, since it is hard to apply the Hough transform to images with a black background. If it is black, then I use the erosion operation on the image to get rid of the white specks on the black background and then the dilation operation to reduce the amount of black text on the white paper. I then use the bwlabel() function to get the labelled matrix of the white object (paper) against the black background. Then I find the angle of the white object with respect to the x-axis using the regionprops() method and the ‘Orientation’ property. Using this angle, I rotate the image so that the white object is upright. Then, again using the regionprops() method and the ‘Boundingbox’ property, I calculate the smallest rectangle bounding the white object so as the crop out the black background.

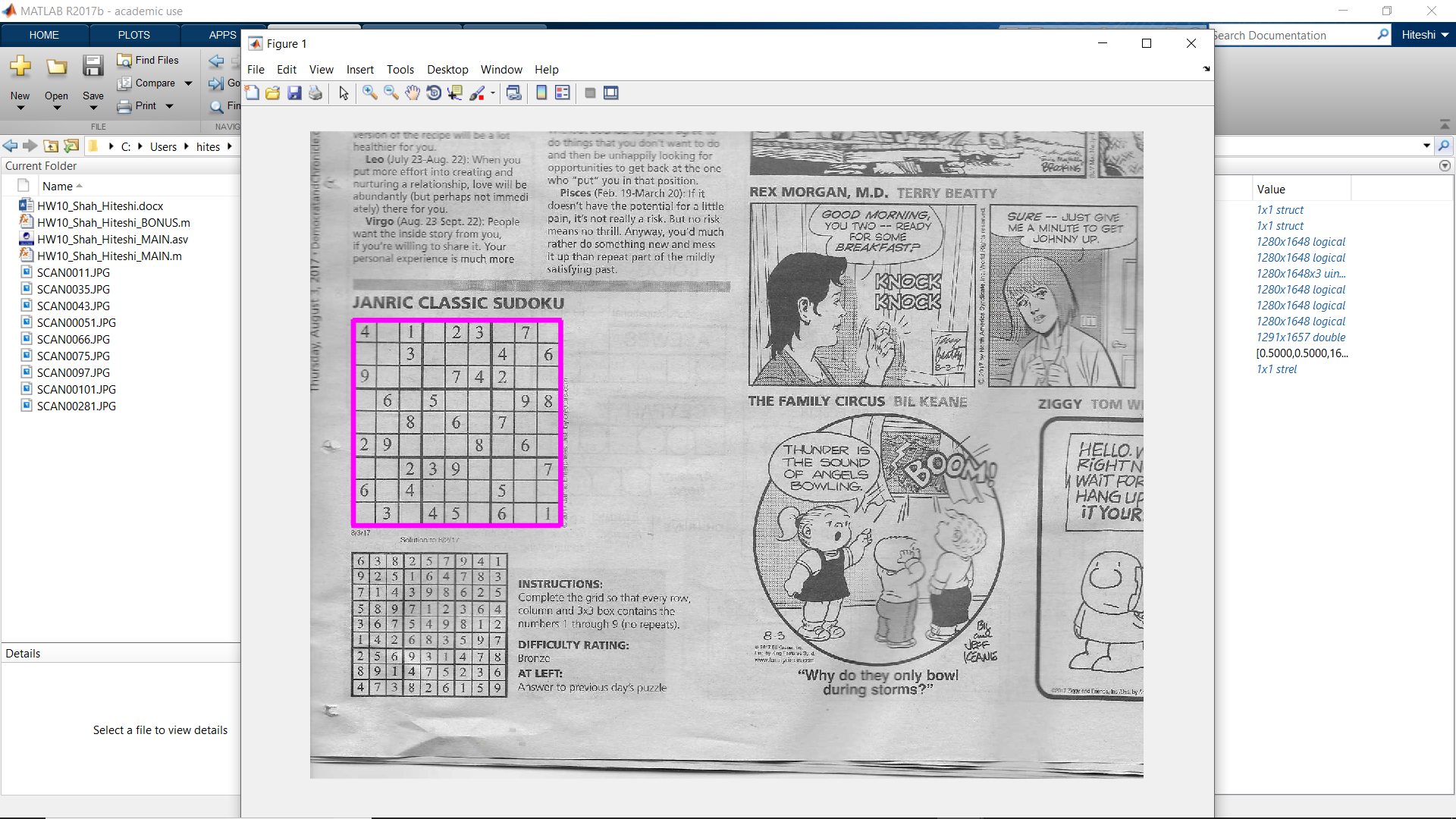
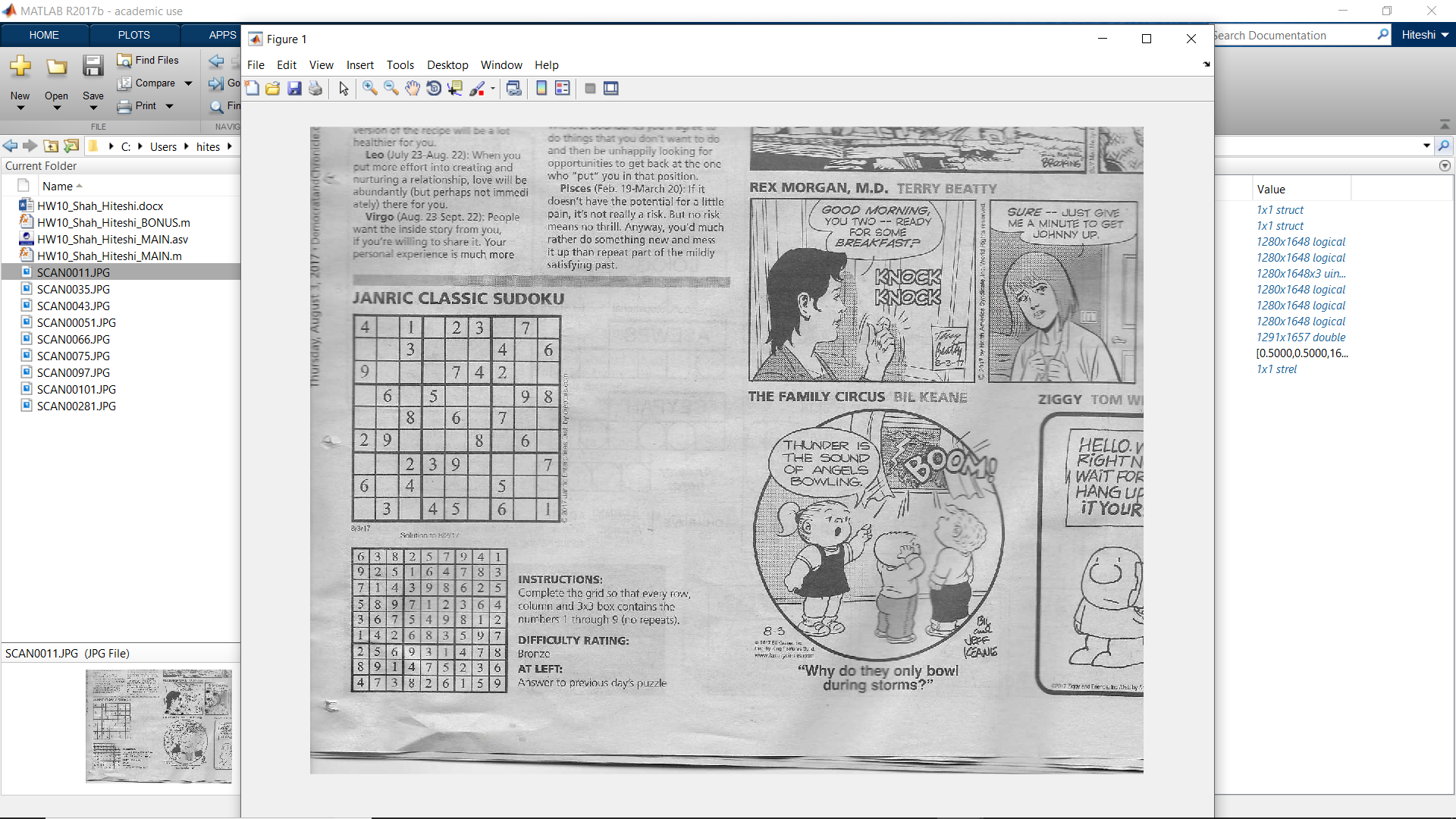
Once the black background has been removed, I continue the program the same way as with the other images that didn’t have a black background. I take the complement of the binary image and perform the erosion operation on the image to reduce the amount of black text. I then use the dilation operation to make the Sudoku borders more prominent so that the Hough transform can detect them.

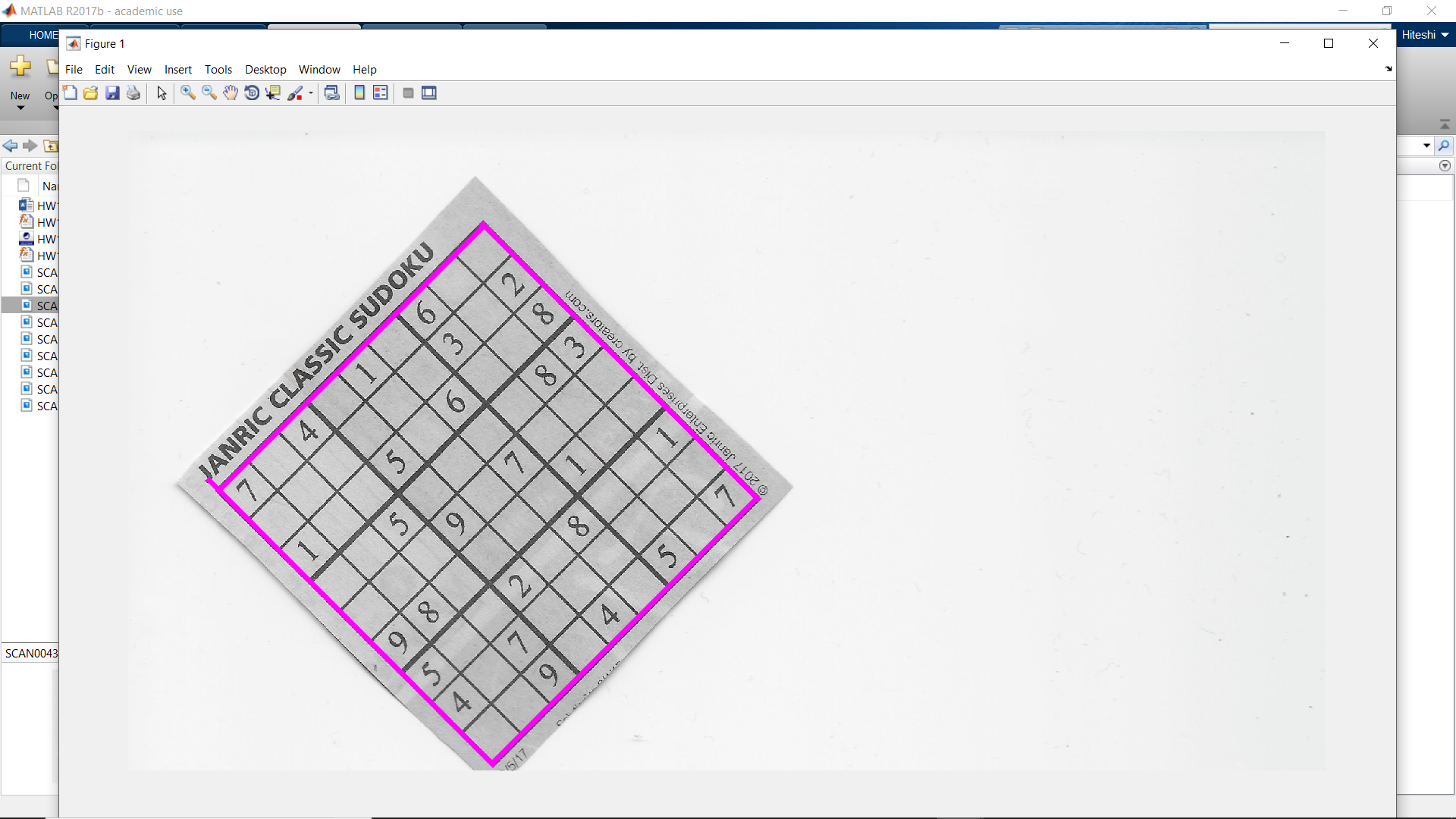
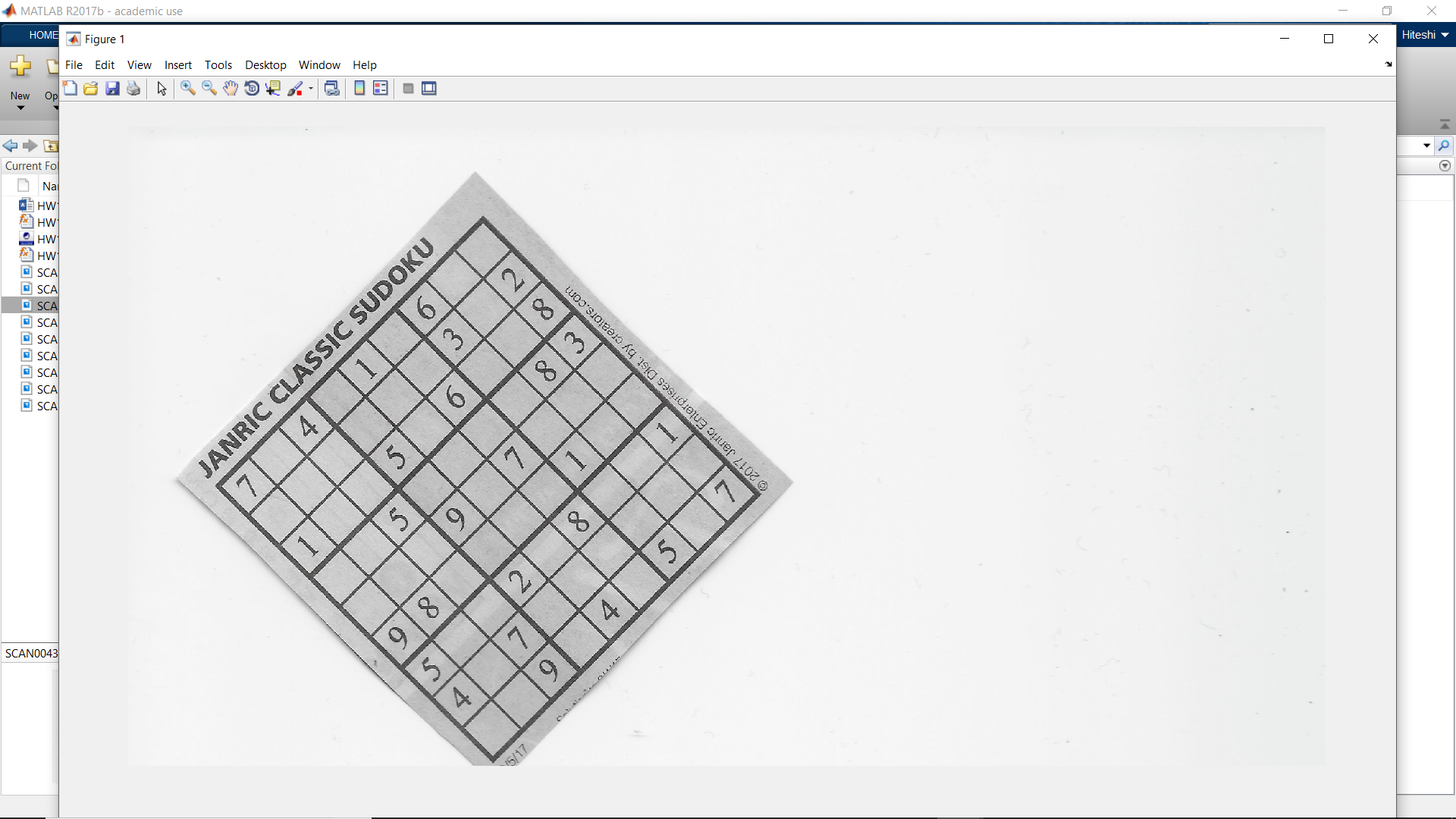
Using the hough(), houghpeaks(), and houghlines() functions, I get a maximum of 30 lines in the image. Once I have these lines, I count the highest number of lines that all have the same length (with some tolerance), under the assumption that these lines belong to the Sudoku. I then check if they are vertical or horizontal.

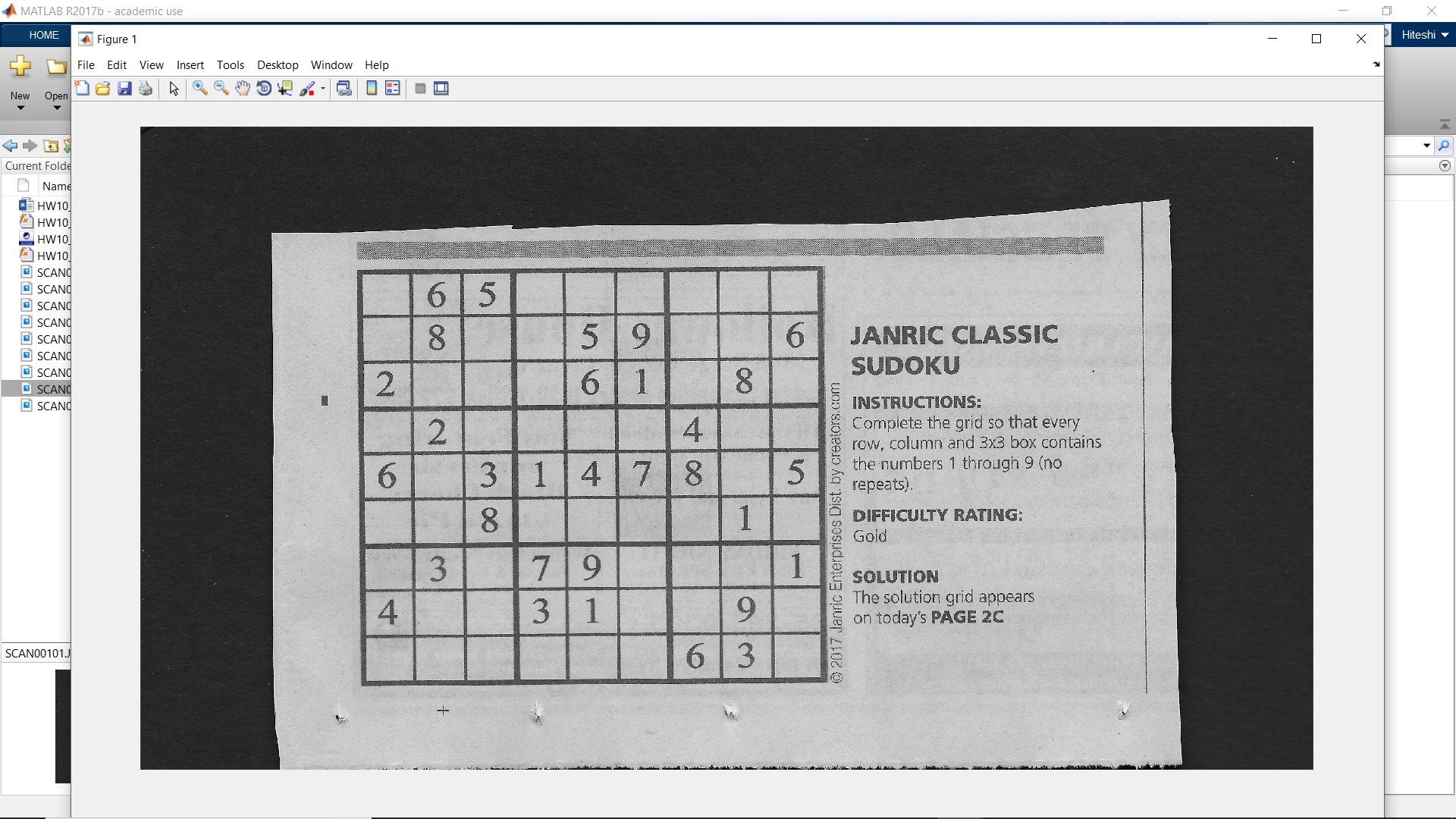
If they are horizontal, I first find the top and bottom lines of the Sudoku box using the y-coordinates of the lines. Then, for all the other lines that intersect with the top line, I check if they are at a 90-degree angle with the top line and if they are, I find the left and right lines of the Sudoku box using the x-coordinates.

Similarly, if the lines are vertical, I first find the left and right lines of the Sudoku box using the x-coordinates of the lines. Then, for all the other lines that intersect with the left line, I check if they are at a 90-degree angle with the left line and if they are, I find the top and bottom lines of the Sudoku box using the x-coordinates.

3. **Results**







4. **Discussion**

The most challenging part of this assignment was tweaking the parameters in such a way that it works for all the images, not just one, such as different morphological operations, as well as the amount of tolerance when finding lines of the same length.

5. **Conclusion**

In this assignment, I’ve learned how to use the Hough transform in Matlab using functions like hough(), houghpeaks(), houghlines(), etc. to identify lines in the image and how different parameters in these functions can affect the output. I also learned how to plot these lines as well as calculate the angles of the lines.