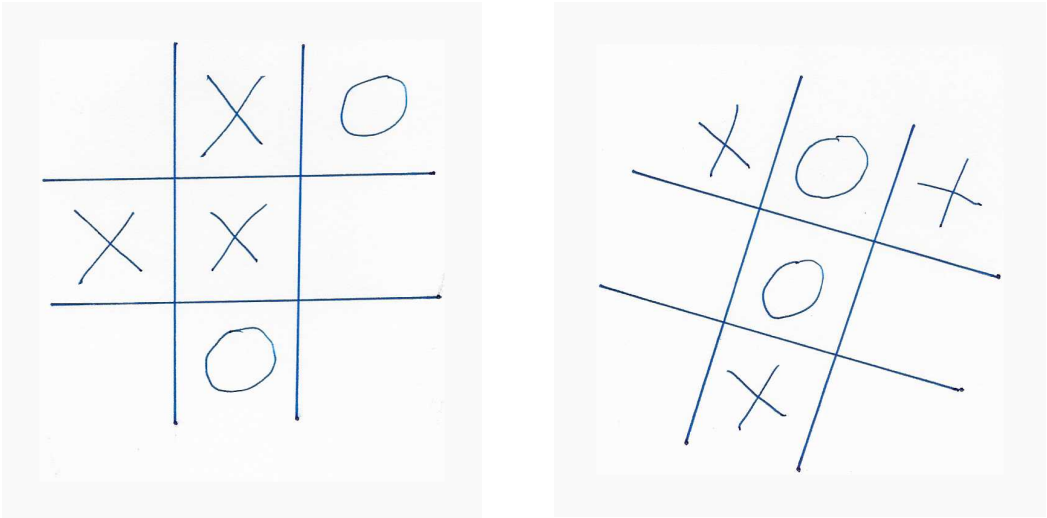
631 Computer Vision
Homework 8

Name: Kriti Sharda

1. Overview:

The main routine sets the path to TEST_IMAGES folder and loops over all the images present in the TEST_IMAGES folder. Each image is pre-processed first, a suitable channel is selected, noise is removed and binary image is calculated. Lines in the tic-tac-toe are calculated using the Hough Transform. Hough transform is also used to get the angle by which the image needs to be rotated in order to segment out the 9 regions in the image. Histogram projection is used on each segment of the image to detect the 'X' or 'O' or an empty space. Finally, the final output is printed on the screen.



Original Images

2. Pre-Processing:

- The image contains blue and green colored lines, so, the red color space channel is selected for these set of images.
- Complement of the image is taken. This makes the background color dark and brightens the lines, the Os and the Xs in the image.
- Median filtering is done to remove any kind of noise.
- This image is converted into a binary image using imbinaze, to obtain the tic-tac-toe lines, the Xs and the Os. Binarizing also removes any noise in the image.
- Image is rotated in order to effectively detect the Xs and the Os in the image.

- The image is cropped to remove the extra spaces around the tic-tac-toe. This way we get the exact region of interest that needs to be worked upon.

- Image is segmented into 9 parts, each part displaying either an X, or an O or an empty space.

- Processes like hough transform, histogram projection, XOR etc are also done on the image. This is explained in details in the next section.

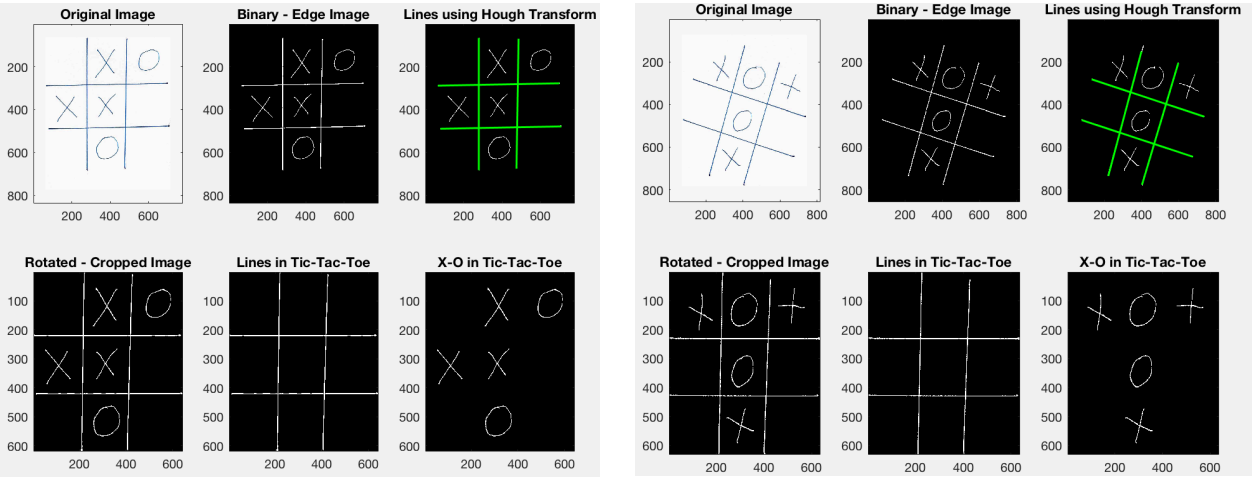
3. Algorithm:

- After the pre-processing is complete, we get a binary image that displays the lines in the tic-tac-toe and the Os and Xs in the image. The first round of **Hough Transform** is performed on this image. The 4 longest lines from the original image are detected and plotted on the binary image. The angle of rotation is calculated using these lines.

- The function **Rotate_Image** is called. This function takes the angle of rotation and the image to be rotated as the input parameters and returns the rotated image as the output.

- The second round of **Hough Transform** is performed on this rotated image. The lines and x,y coordinates on the end points of the lines are calculated. This helps us to crop the image and get rid of extra spaces around the tic-tac-toe.

- The in-build matlab function **Bwareaopen** is used to fetch the 4 longest lines in the image. This way we get an image that contains only the structure of the tic-tac-toe and get rid of the Xs and Os. The **XOR operation** is performed between the cropped tic-tac-toe image and the tic-tac-toe structure image in order to get just the Xs and the Os.



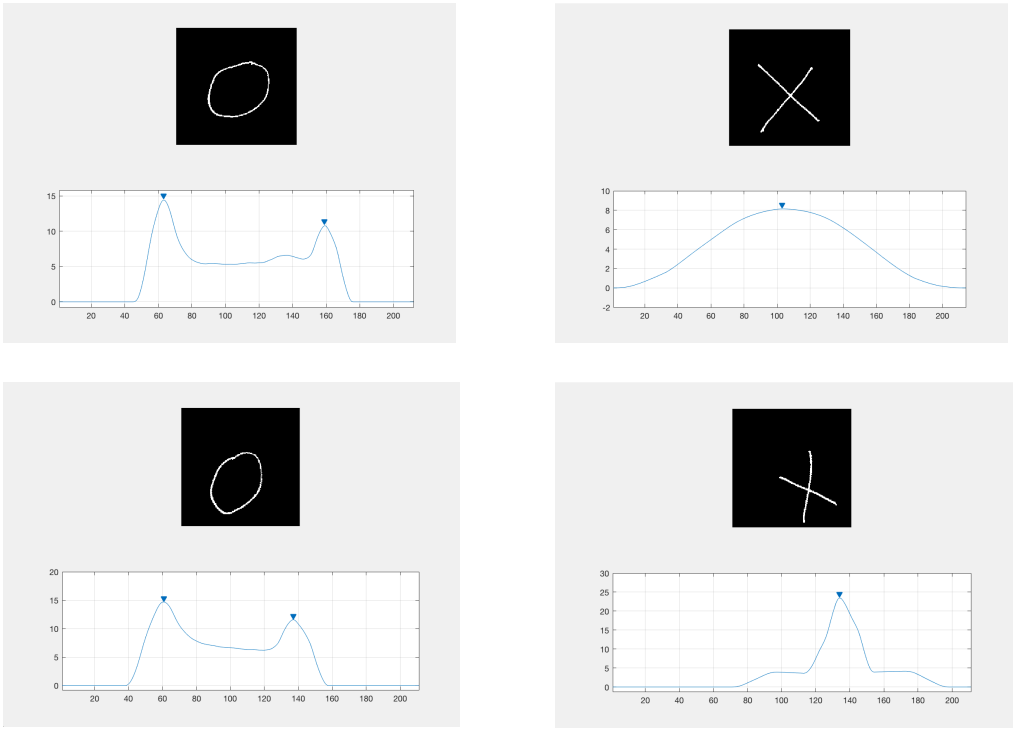
Pre-processing steps and different stages of the algorithm

- Next, the function Get_X_O is called. This function takes the XOR result image as a parameter, that is, the image that contains just the Xs and Os and prints the final output on the console window.

- This function segments the image into 9 parts and deals with each segment of the image independently. For each segment a **Histogram Projection** of the image along the vertical direction is calculated. The plot for the number of potential peaks is studied.

- To find potential peak, a lot of parameters are set according the given input dataset. A minimum height of the peaks is set to 6.5 units and the minimum distance between the peaks is set to 75 units. These parameters have been set by examining all the images and their corresponding histograms projections.

- Finally, by counting the number of potential peaks, the value of 'X', 'O' or '_' is printed as an output.



Detecting 'O' and 'X' using histogram projection

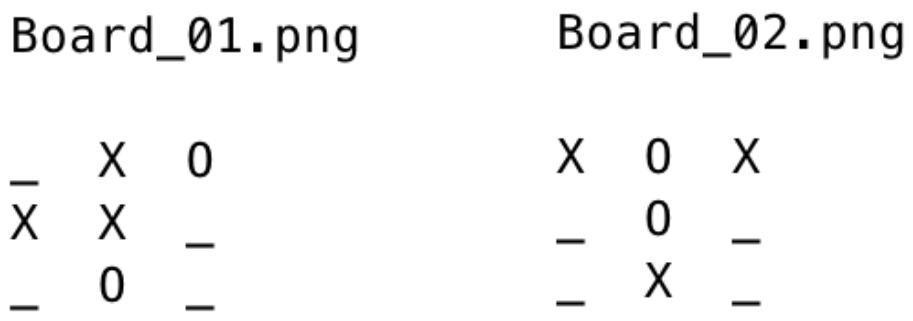
4. Assumptions:

I assumed that the 4 longest lines in the tic-tac-toe are the horizontal and vertical lines that define the structure of the tic-tac-toe. Apart from the tic-tac-toe, the image doesn't contain any other text, symbol etc. Proper lightning is provided and the background and image color is the same as that been provided.

5. Final Result:

I studied all the image histogram projections and tuned all the parameters to calculate the potential peaks accordingly. My algorithm detects all the tic-tac-toes from the given test images correctly. It doesn't fail for any test image.

There might some images in future for which the algorithm might give incorrect results because the parameters are set according to the given test images.



Final output results for the given input image

6. Conclusion:

It was a really interesting assignment and I learned a lot of things on my way to the finish of the assignment. I got to explore how the Hough transform works in matlab. I also got to know about the various function like houghlines and houghpeaks, that are some function using which we can extract information from the Hough Transform plot of an image. I learned how to rotate images by the correct angle, which was obtained by Hough transform again. It is amazing to see how many things can be achieved by using this algorithm.

I got to explore new things about histogram projection. I understand how powerful it is to study the images and extract features from the images. I did not know about the various parameters that we can set while finding peaks, like, minPeakProminence, minPeakDistance, minPeakHeight, sorting the peaks, fetching the coordinates of the peaks, etc.

Overall, I gained a lot of knowledge and loved working on this assignment!