**Project 2 Report**

**Removing Unwanted Caption from Snapchat Images**

ECE-S435 Dr. Tretiak

Group 2:

David Tigreros Matthew Giovannucci

**Objectives:**

The objective of this project was to use MATLAB software to implement an algorithm that would have the capabilities of removing unwanted caption bars and text from images taken with the Snapchat picture messaging mobile application. Using the Snapchat mobile application, users can take pictures with a smartphone and add captions to these images that consist of a bar with text to send to other Snapchat users. Users are also able to save these images for later viewing, this can however lead to issues as once the image is saved the caption bar and text is “burned” onto the image and can no longer be edited or removed. This can be annoying to users as it disrupts the saved images with unwanted objects in the image. The goal for our project is to process these saved images so that the caption and bar are removed and what is left is the unedited original image.

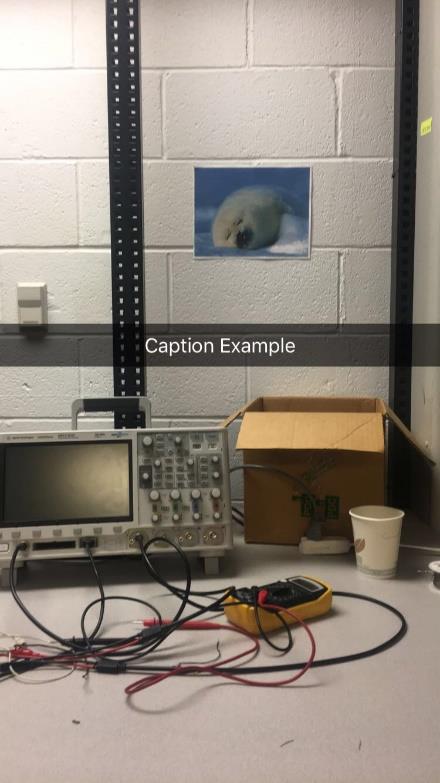
 

Figure 1. Snapchat image with unwanted caption example (left). Goal of removing caption to revert to original image on the right.

**Method:**

In order to develop the MATLAB algorithm that will remove unwanted captions from Snapchat images, we thought the best approach would be to first isolate the transparent caption bar that shows the placed text onto the image. To isolation this caption bar, the Hough transform was utilized to extract the line segments in the given image. Using the Hough transform with a determined threshold and minimum line length, we were able to isolate the caption bar of the Snapchat images. Figure 2 displays an example image where the isolated caption bar is highlighted with a green line.

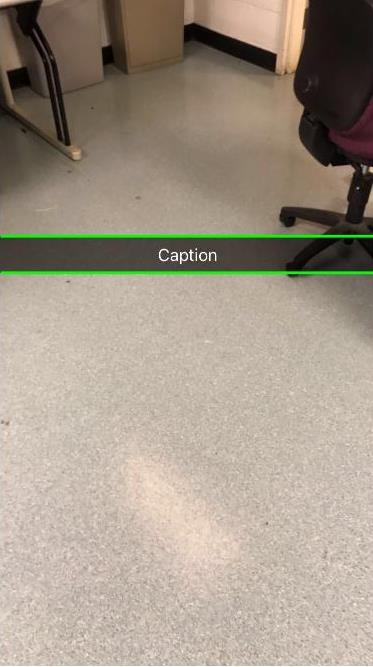


Figure 2. Isolation of the Caption Bar

With the caption bar now isolated, the next step was to isolate the text within the caption bar. In order to properly isolate the text, the image needed to binarized. Binarizing the image to black and white with a high threshold was necessary as it allowed us to differentiate the text from the rest of the image since it was a white text with a mostly black background. After binarizing the image, the image was segmented with regionprops so that the area of each letter of the text, in pixels, could be found. After identifying the pixel area of each letter of the text, this was used to highlight the location of the text on the image. To ensure that only the area of the text was highlighted, the edges of the isolated caption bar were used as a boundary. Figure 3 displays the highlighted text.

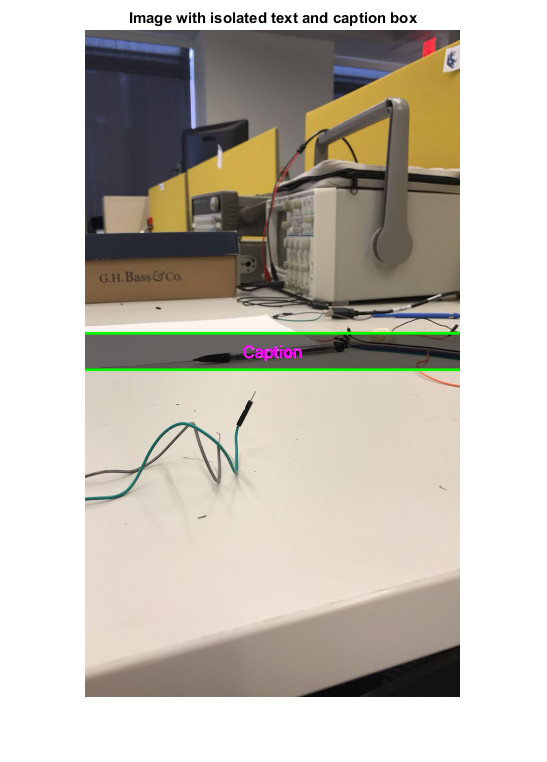


Figure 3. Isolation of the text

Now that the caption bar and the text within the bar were isolated, the next step was to determine the best method of removing the text. The approach used was to identify the text region boundary, that is the entire area of the text instead of the area of each letter as previously found. This was done by taking the maximum and minimum vertical and horizontal values of the known text pixels to create a polygon shaped region around the entire text area. Once the area of the text region was found, the area was filled in using inward interpolation. With inward interpolation, the text area was masked to look like the rest of the surrounding image. The downside to this approach was that to fill the region with inward interpolation, the image needed to be converted to grayscale. Once the inward interpolation of the text region was complete the image was converted from grayscale back to an image that can hold RGB values. Even though the image can hold RGB values, it was still gray. To return the color to the image, the color map of the original image was used to copy the values back into the converted RGB image. However, in the process of using the color map to restore color to the image, the RGB values of the area containing the text of the original image was omitted. Figure 4 displays the image after inward interpolation and after the color mapping.

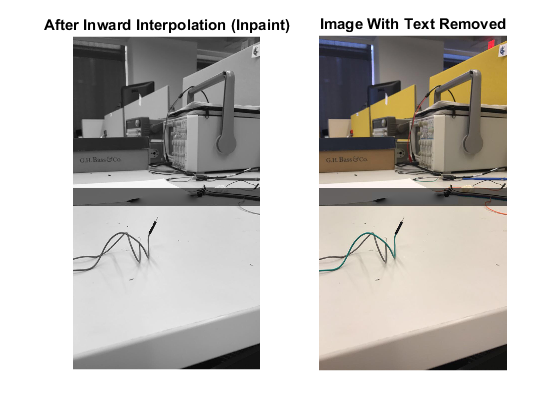


Figure 4. Removal of text after inward interpolation (left). Restoring color with color map of original image (right).

After locating the caption bar, isolating the text, and removing the area where the text was located, the final step was to remove the caption bar. The approach used to remove the caption bar was an averaging filter. An averaging filter will increase the intensity of the pixels that are darken by the snapchat bar. This algorithm takes 5 pixels in a random column and measures the intensity difference between pixels in the same column within the bar. The difference will then be averaged for each RGB value and that value of all the pixels in the bar will be increased by the amount calculated. Once this was complete, the caption bar was removed. Figure 5 displays the image after removal of the caption bar, along with the original image for comparison.

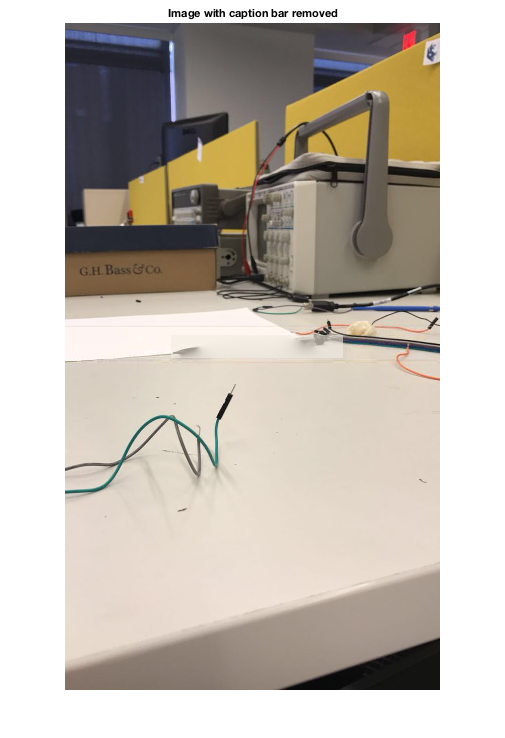
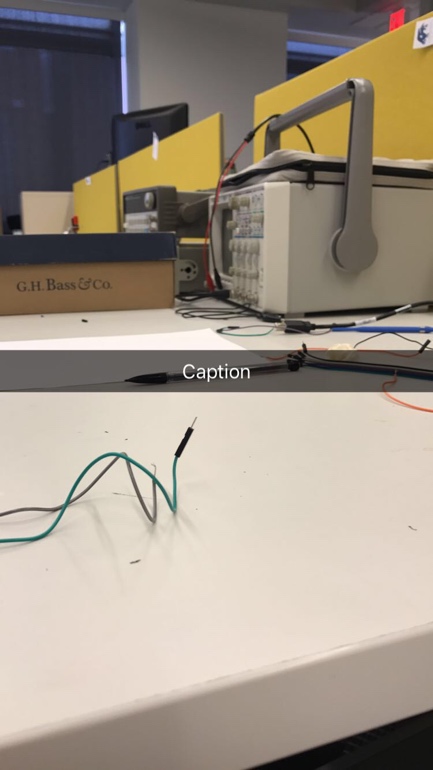


Figure 5. Original image (left). Image with caption bar and text removed (right).

**Conclusion:**

In conclusion of this experiment, the group learned a great deal about various image processing techniques and were able to use many of the techniques learned from the course along with researching and discovering new image processing tools. From the image processing course, techniques such as image segmentation, filtering, and color mapping were utilized. It was interesting to see these techniques that were taught be used to a relevant application. In addition to using learned techniques, researching new tools, such as the Hough Transform, was integral to our project as it was a necessary tool for isolating the caption bar region of the Snapchat images. Although more difficult than anticipated, this project was very interesting and insightful as it allowed us to use image processing to tackle new problems that emerge from the exploding social media and smartphone application industry that continues to utilize images to grow.