Name (netid): Muhammad Rafay (mrafay2)
CS 445 - Final Project (Individual): Shadow Removal

Motivation and impact:

The presence of shadows in images create difficulties in image processing tasks like image segmentation, object detection, and tracking. Moreover, shadow-free images are of great interest for image editing, computational photography, and augmented reality.

Hence this project is based on the relighting shadow removal technique used in Guo et al. 2013 paper. I learned one of the best shadow removal techniques that are based on relighting rather than the older reintegration techniques. This allows removing lighter shadows (penumbra) from the image as well.

Approach:

The approach is based on the relighting-based shadow removal technique used in Guo et al.'s 2013 paper. This technique uses a single image with user-defined hard masks that distinguish between shadow and non-shadow regions. The technique uses the closed-form matting technique proposed in A. Levin D. Lischinski and Y. Weiss. A Closed-Form Solution to Natural Image Matting to generate soft masks. The python implementation of closed-form matting is available on GitHub and was used to generate these soft masks.

The purpose of the soft masks is to identify the intensity (**Ki***)* of the shadow at a pixel ***i*** which ranges from [0,1] where 1 means non-shadow pixel, 0 means an umbra and a value in between is a penumbra.

The shadow-free image was generated using the following equation (12) from the paper where **r** is the ratio between direct light (**Ld**) and the environment light (**Le**) in the image. We assume that the intensity of both of these lights is the same across the image.

$$\begin{aligned} \mathbf{I}_i^{shadow\_free} &= (\mathbf{L_d} + \mathbf{L_e})\mathbf{R}_i \\ &= (k_i\mathbf{L_d} + \mathbf{L_e})\mathbf{R}_i \frac{\mathbf{L_d} + \mathbf{L_e}}{k_i\mathbf{L_d} + \mathbf{L_e}} \\ &= \frac{\mathbf{r} + 1}{k_i\mathbf{r} + 1}\mathbf{I}_i \end{aligned}$$

To calculate the values of **r,** we used the equations (13-15) from the paper. Where i and j are corresponding shadow and non-shadow patches around the shadow boundary. To avoid misdetections we perform voting in the joint RGB ratio space. From each patch pair, you compute **r** for each color channel, so each **r** is 3d. Then you compute a 3d histogram of the **r** values with

$$\mathbf{I}_i = (k_i\mathbf{L_d} + \mathbf{L_e})\mathbf{R}_i \quad (13)$$

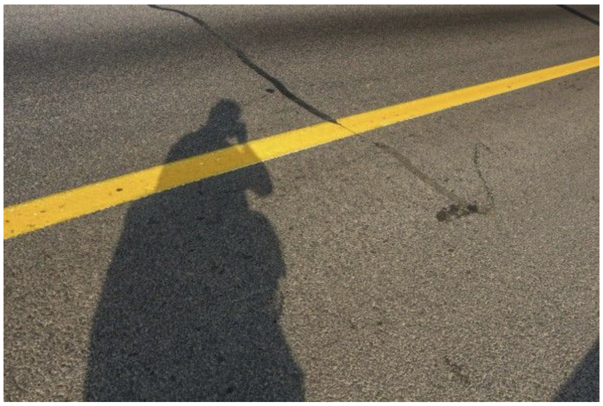
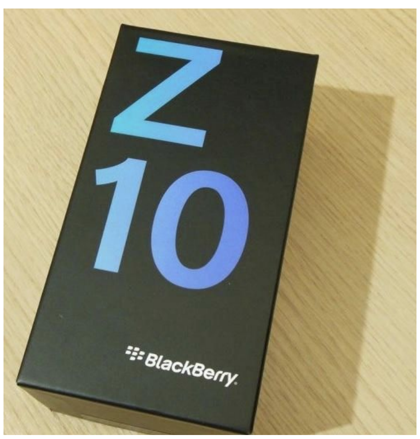
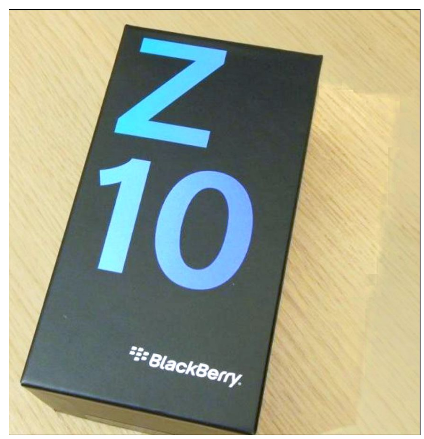
$$\mathbf{I}_j = (k_j\mathbf{L_d} + \mathbf{L_e})\mathbf{R}_j \quad (14)$$

with $\mathbf{R}_i = \mathbf{R}_j$.
From the above equations, we can arrive at:

$$\mathbf{r} = \frac{\mathbf{L_d}}{\mathbf{L_e}} = \frac{\mathbf{I}_j - \mathbf{I}_i}{\mathbf{I}_i k_j - \mathbf{I}_j k_i} \quad (15)$$

a bin of 0.1. E.g. 0,0.1,0.2,0.3,0.4,0.5...5.0 as the bin boundaries. Then we take the mode, the most common ratio **r,** and plug it into the equation (12) to recover the shadow-free image.

Results:
Some of the good results are as below:

Shadow Image	**Recovered Shadow Free Image**
	
	
	

Challenge / innovation:

Among the challenges I faced was understanding the unclear steps from the paper on how the patching was being done on both sides of the shadow edges and then how they performed the voting in the joint RGB ratio space to get the final ratio.

So initially as a naive approach, I took the ratio between the whole shadow and non-shadow region but the results were not satisfactory. So then I used patching but that wasn't working as well. So I identified that I need to get an equal amount of patch across the edges from both regions so I added a check for that, each time the kernel reached an edge the patch wasn't considered until it contains roughly a 50/50 ratio of both regions, the results improved but they were still not up to the mark. Then I tried understanding what was meant by "voting in the joint RGB ratio space to get the final ratio" and I found out that we have to make a histogram of the ratios in the RGB and then use the most common one. I implemented the histogram technique and the results got quite up to the mark.

So the last problem that remained was the strong boundary effect on the edges where the shadow was removed. This was because shadow transitions a bit smoother on the edges. Therefore I then used a closed-form matting algorithm to generate the soft mask, it identifies the intensity of the shadow at each pixel. The boundaries having a transition effect.

To generate a soft mask the closed-form matting algorithm requires scribbles from the user on the image to identify what is foreground and the background is in the image. That requires very careful labeling and one might not be successful at the first attempt.

Lastly, a problem that remains unresolved is that when we scale an image with the calculated ratios the range of some of the resulting pixels goes out of the range and we need to clip them for the final output result.

There were quite a several unclear steps in the technique that required a lot of guessing and trial to reach a stable algorithm for shadow removal, although it still has quite a room for improvement I believe. This shadow removal technique based on relighting was devised by Prof. Derek and his students which is one of the recent and the best. Given the challenges involved, I expect to get at least more than 10 points in the challenge/innovation component of grading.