Computer Engineering Department
TED University

Spring 2023 – CMPE 362 CS563 Digital Image Processing – Assignment 1
Due Date: April 9th, 2023 at 23:59
This assignment is to be done individually!

In this assignment, you will implement an image editing tool which involves putting foreground object of one image into another image that we call as the background image. You will also add effects as smoothing the background and sharpening the foreground.

Part 1:

First, choose an image from MSRA10K[1] dataset whose foreground object is to be put into another image. Let us consider that we choose the image in Figure 1 (Left). Each image in MSRA10K[1] dataset has a corresponding binary image containing mask of the foreground object (see Figure 1 (Right)). The binary image has the same size as the foreground image where its value is 255 at the pixels belonging to the foreground object and 0 at the remaining pixels.

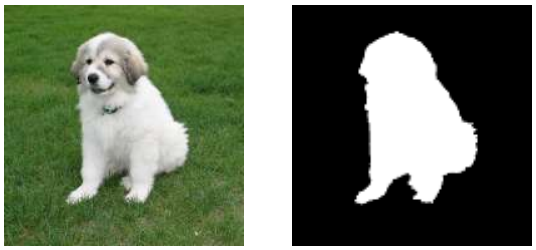

Figure 1. (Left) An image from MSRA10K[1] dataset. (Right) The corresponding binary image containing mask of the foreground object.

Second, choose the background image. Note that you could choose any image you want (it does not have to be an image from MSRA10K[1] dataset). Let us consider that we choose the image in Figure 2 (Left) as the background image.

Third, put the foreground object into the background image. See Figure 2 (Right) for the resulting image.

Prepare a Jupyter Notebook `part1.ipynb` that contains all the code and the results. In Part 1, you are going to implement the function `combineForegroundBackground`.

```
def combineForegroundBackground(fgImg, fgMask, bgImg, topLeft):
```

This function returns an image obtained by combining the foreground and background images so that pixel at (0,0) of the foreground image coincides with pixel at `topleft` of the background image. Notice that the background image is updated only at the pixels whose corresponding pixels in the foreground image belong to the foreground object.

Important Notes:

- You need to implement `combineForegroundBackground` function yourself i.e. you are not allowed to use any library functions combining foreground and background images.

- If there is an overflow when combining foreground and background images, you need to consider only pixels of the foreground image that have corresponding pixels in the background image. In other words, overflowed regions of the foreground image are not considered during combination. See Figure 3 for examples.

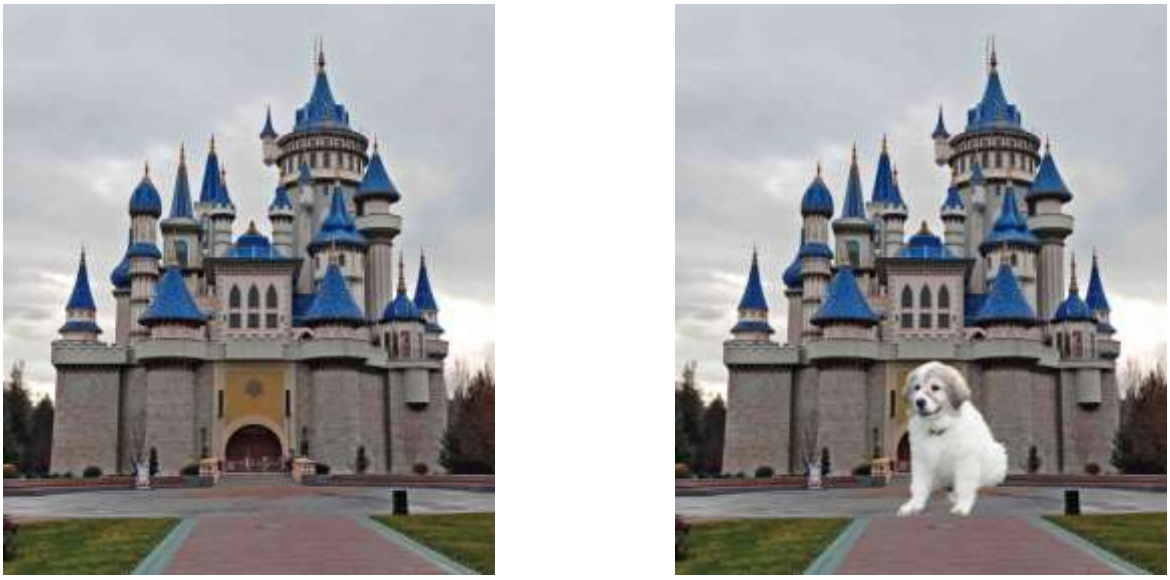

Figure 2. *(Left)* Background image. *(Right)* Result obtained by putting the foreground object into the background image where top left pixel of the foreground image coincides with pixel at (608, 302) of the background image.

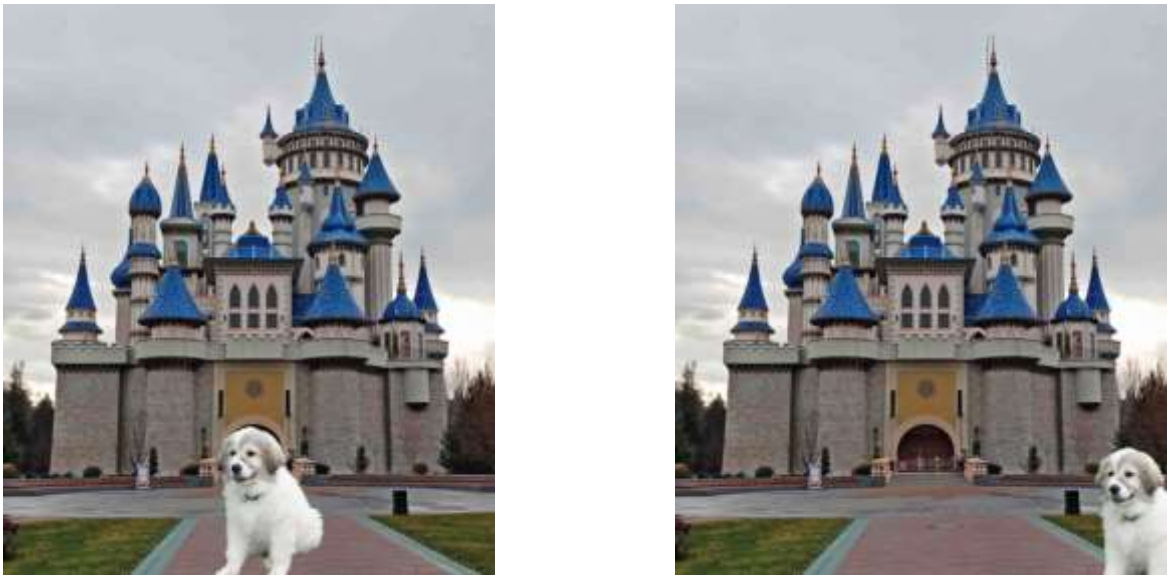

Figure 3. Results obtained when top left pixel of the foreground image coincides with the following pixels of the background image: *(Left)* pixel at (728, 279) *(Right)* pixel at (766, 656).

Part 2:

In this part, you will add visual effects by smoothing the background image and sharpening the foreground image.

```
def gaussianSmoothing(img, sigma):
```

First, implement `gaussianSmoothing` function using `getGaussianKernel` and `sepFilter2D` functions (or `getGaussianKernel` and `filter2D` functions) of OpenCV (you are not allowed to use `GaussianBlur` function of OpenCV). Note that size of the filter should be the first odd integer greater than or equal to 6*`sigma`. Use `gaussianSmoothing` function to smooth the background image for different `sigma` values. Figure 4 shows results obtained by applying Gaussian smoothing to the background image for two different `sigma` values.

```
def unsharpMasking(img, sigma):
```

Second, implement `unsharpMasking` function using `gaussianSmoothing` function. Use `unsharpMasking` function to sharpen the foreground image for different `sigma` values. Figure 5 shows results obtained by sharpening the foreground image for two different `sigma` values.

Third, combine the sharpened foreground image with the smoothed background image. See Figure 6 for example results.

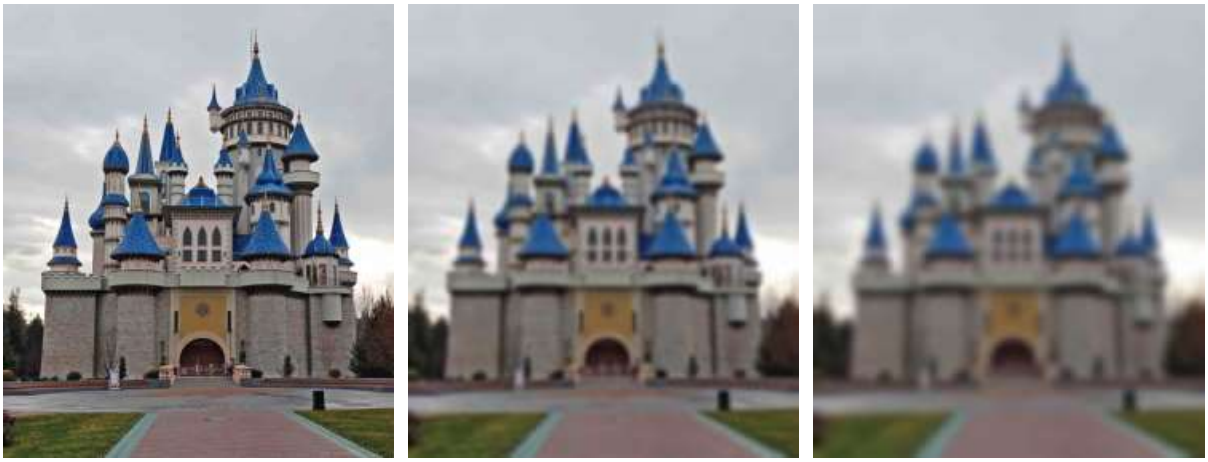

Figure 4. *(Left)* Background image, and its Gaussian smoothing for *(Middle)* `sigma` 5 *(Right)* `sigma` 10.

Figure 5. *(Left)* Foreground image, and its unsharp masking for *(Middle)* `sigma` 3 *(Right)* `sigma` 8.

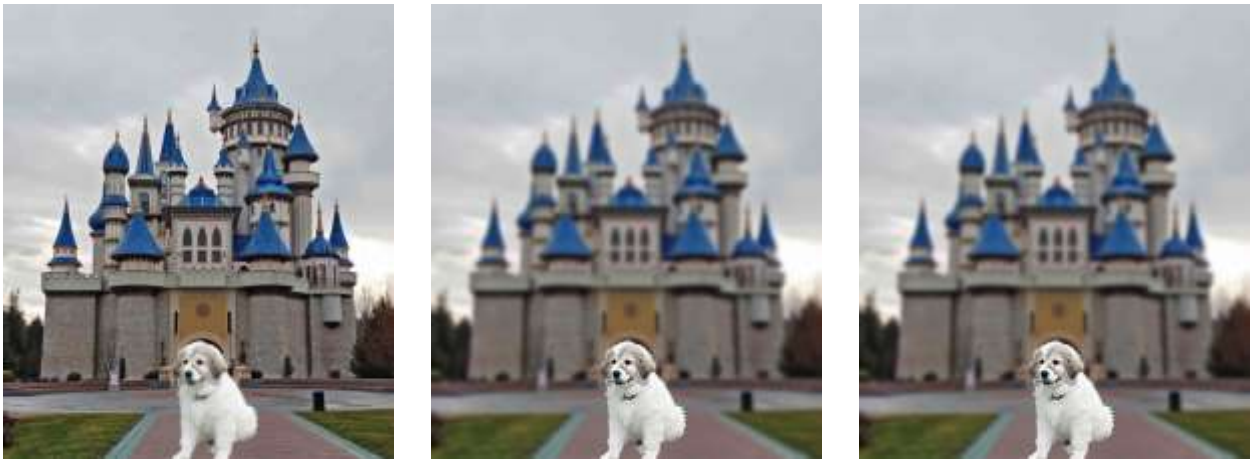

Figure 6. Results obtained when top left pixel of the foreground image coincides with the pixel at (728, 279) of the background image *(Left)* without smoothing background and sharpening foreground *(Middle)* foreground sharpened with `sigma` 3 and background smoothed with `sigma` 5 *(Right)* foreground sharpened with `sigma` 8 and background smoothed with `sigma` 10.

What to hand in:

- Code
  - `part1.ipynb` containing the codes and results for part 1 including the definition of `combineForegroundBackground` function.
  - `part2.ipynb` containing the codes and results for part 2 including the definition of `combineForegroundBackground`, `gaussianSmoothing` and `unsharpMasking` functions.
- Report
  In your report, you are expected to include the following information:
  - Brief explanation of what you have done
  - The results that you have obtained together with the corresponding parameters and discussion of how the results change as the parameters change
    - You could use input images (77162.jpg, 77162.png, sazova_park.png) provided in Moodle while implementing and testing your codes.
      However, you need to choose a new pair of foreground and background images and present your results for Part 1 and Part 2 similar to the results in Figures 1-6. That means, in your report, you need to present the following items for the pair of images you have chosen:
      - foreground image and foreground mask
      - background image
      - combination of foreground image with background image for three different positions where in two of them the foreground image overflows boundary of the background image
      - Gaussian smoothing of background image for two different `sigma` values
      - unsharp masking of foreground image for two different `sigma` values
      - combination of foreground and background image
        - without sharpening foreground and smoothing background
        - sharpening foreground and smoothing background for two different combination of `sigma` values

It is important that you include all your results and discussion in the report.

The results presented in Figures 2-6 are also provided in Moodle so that you could use them for testing your code.

Acknowledgement

This assignment is a simplified version of the one that Erkut Erdem developed for his Fundamentals of Image Processing class.

References

[1] https://mmcheng.net/msra10k/