M21CS7.404: Digital Image Processing
Monsoon-2021
Assignment-3
Posted on: 05/11/2021
Due on: 23:55hrs, DUE DATE: 18/11/2021

Introduction

1. All functions in this assignment need to be implemented from scratch unless specified otherwise.
2. Follow the specified repository structure.
3. `src` will contain the Jupyter notebooks used for the assignment.
4. `images` will contain images used for the questions.
5. Make sure you run your Jupyter notebook before committing, to save all outputs.
6. Make sure you commit and push your work regularly.

Questions

1. (30 points) You have already written codes for median filtering, edge detection, contrast stretching and histogram equalization for grayscale images. Now you are required to extend those codes to colorful RGB images using the methods discussed in the class.
    1. Implement a function `medianFilter` which takes in an RGB image `im`, filter size `k` and use it for the image `NoisySimba.png` and report the output.
    2. Write a function `colorLinContrastStretching` which takes an image `im` as input and returns a contrast-enhanced version of the image. Apply the function on `fog.jpeg` and display your result.
    3. Perform histogram equalization on the image `fog.jpeg` and show the result.
    4. Demonstrate ”Vintage effect”, ”Matrix effect”, Vignetting and duo-tone on any image of your choice.
       Note: You are allowed to use `cv2.LUT()` for duo-tone.
2. 1. (20 points) Find the cell organs of interest in the image `cell.png`. Use the same code to show your outputs on `flower.jpeg`. See figure 1.

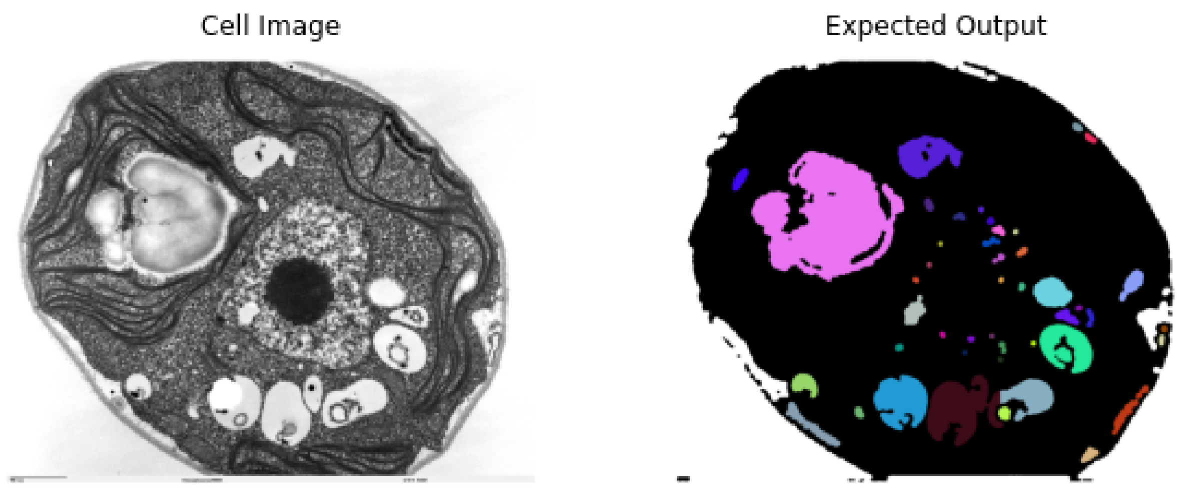


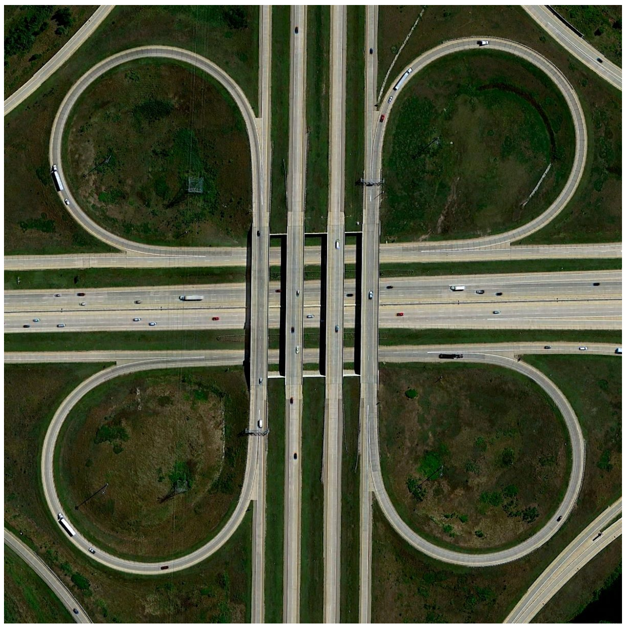
Figure 1: Expected Output

2. (20 points) Read in the binary image `objects.png` and write a script which uses the image as input and answers the following questions.
   (a) How many objects have one or more holes?
   (b) How many square objects are in the image?
   (c) Identify the square objects that have holes.
   (d) Identify the circular objects that have no holes
3. (20 points) Read in the binary image `circles.jpg` and write a script which uses the image as input and outputs a new image containing
   (a) only the coins touching the boundary of the image
   (b) only the coins which overlap with each other
   (c) only non-overlapping coins

3. (40 points) Display the segmented image for `coins.jpg` using below mentioned methods and count the number of coins in each method:

   1. **Thresholding**
      (a) Show the results on multiple threshold values. Use morphological methods to fill gaps in the thresholded image and obtain the number of coins without manually counting.
      (b) Implement **Otsu's thresholding** and then perform the counting. Compare your results with `(a)`.
   2. **Edge-based segmentation** : Show the results by using different detection methods like sobel, canny and laplacian to detect the edges and further use morphological methods to obtain segmented image.
   3. Compare and analyse the results obtained from the above segmentation methods.

4. (40 points) Consider the image `roads.jpg` and solve the following:


(a) Road Image

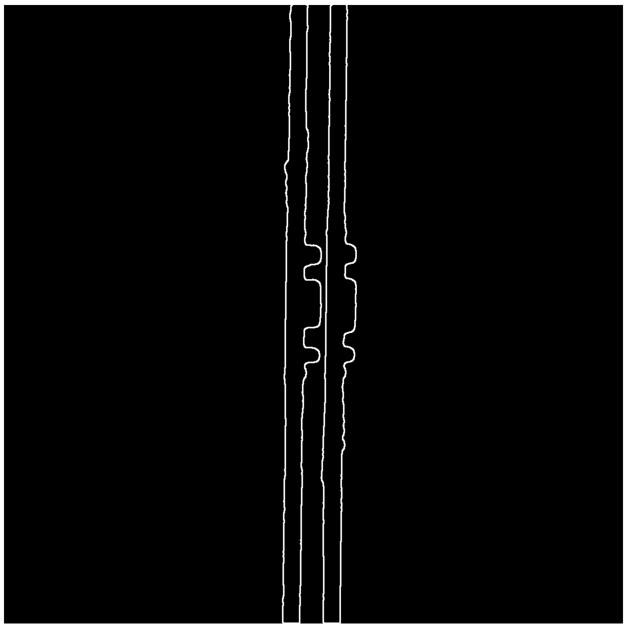
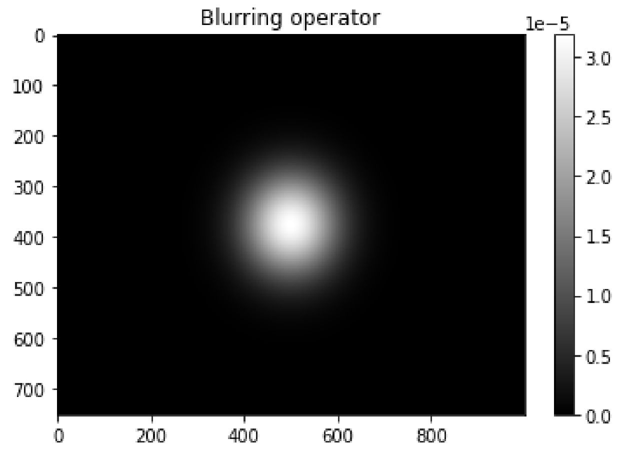

(b) Segmented vertical carriageways

Figure 2: .

1. Show the outline of the vertically oriented middle carriageways which are not connected to the circle like interchanges. Refer figure 2a.
2. Display the best fitting circle for each of the circle-like interchanges and output the radius of each in pixels.
3. Segment all the large trailer trucks visible in the picture. (Hint : Select the seeds in side the trucks, use region growing to segment those trucks)

5. (50 points) Given the image `shinchan_blur.png` that is deliberately blurred using a Gaussian point spread function of width σ.



1. Write a program that creates an array, of the same size as the photo, sampled from Gaussian point spread function (*psf*) f(x,y) with σ of your own choice. Display this

array, remember that the point spread function is periodic (along both axes), which means that the values for negative x and y are repeated at the end of the interval. Since the Gaussian is centered on the origin, this means there should be equally bright pixels in each of the four corners of your picture like the above displayed image.

2. Deconvolute the blurred image with *psf* calculated above. (Hint : To get the blurred image, we convolute the focused image with the *psf* to blurred image. Use fourier transform to deconvolute.)
   (a) Calculate the Fourier Transforms of Blurred image and *psf*.
   (b) Divide Fourier Transform of Blurred image with the Fourier Transform of *psf*.
   (c) Perform Inverse fourier transform.
3. Try the deconvolution of blurred image with *psf* for various values of σ.
4. Display the your results for the σ that deblurred the best.

6. (40 points) Kalyan, who only uses discord wants to share the image `shinchan.pkl`, stored as a string, which is of size 15.4 MB to his friend. Discord only supports files of size less than 10 MB.

   1. Help Kalyan to compress the color image using Huffman encoding to share the image to his friend. You can learn about huffman encoding here.
      (a) Create the dictionary of of all the letters in the image array.
      (b) Generate the Huffman Tree from the tree generated by considering all the letters in the dictionary as the leaf nodes. Print the Huffman tree.
      (c) Assign the code words 0 and 1 to the nodes of the Huffman tree.
      (d) Generate binary code of the image by the binary representation from the above tree.
   2. Display the image stored in the file `shinchan.pkl`.
   3. Save the generated binary code of image in `compressed.bin`. This is supposed to be less than 10MB in size. He will be sharing this file to his friend.
   4. Print the size of extracted text from `shinchan.pkl` and size of `compressed.bin` in MB.
   5. Write a function to help his friend decode the original image from `compressed.bin` and display it.
   6. Verify if the image obtained after decoding and the original image are the same.