Assignment 1

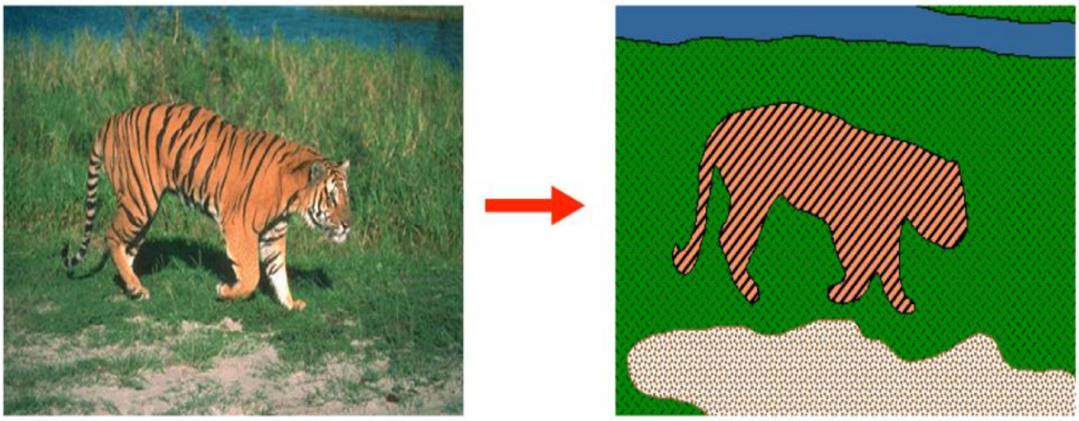
Q1. Write a Python program that takes any one of the RGB image from the given database as input, converts it to a quantized image by using k-Means image quantization at varying ‘k’ values i.e. 1, 2, 4, 8, 16. Also, calculates the Mean Squared Error (MSE).

Your program should include the following steps:

- Read an RGB image as input.
- Implement a function to perform k-Means image quantization.
- Convert the input RGB image to a quantized image using the k-Means algorithm by varying ‘k’.
- Calculate the Mean Squared Error (MSE) between the original RGB image and quantized image.
- Analyze and discuss the effectiveness and limitations of k-Means image quantization in the context of the calculated MSE.

Reference: https://towardsdatascience.com/image-quantization-with-k-means-db7127503bcb

Q2. Write a code to find the connected components present in an image as shown in the given example. Utilize any image from the internet of your choice and identify the components present in it, emphasizing accuracy.



Reference: https://github.com/swetanjal/Digital-Image-Processing

Q3. From the attached database of standard test images, select the gray scale ‘lena_gray_512.tif’ image of dimension pixels. Perform the following operations:

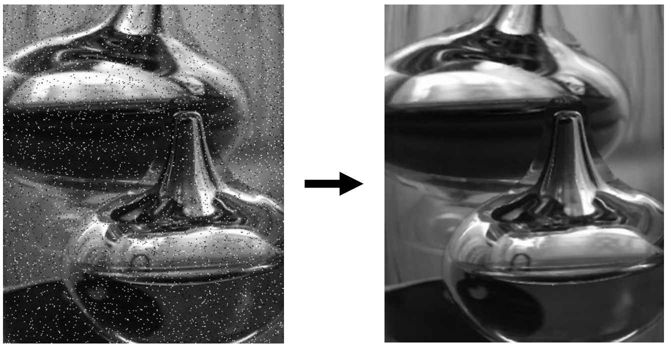
(a) Reduce the image size by down sampling to 256x256, name this image as ‘reduce_lena_256.tif’.

(b) Compare the reduced image ‘reduced_lena_256.tif’ with the given image in the database ‘lena_gray_256.tif’ by finding the PSNR values between them. Also plot the image showing MSE difference for the same.

(c) Now again resize the computed image ‘reduced_lena_256.tif’ to original dimension pixel using these methods:

- Nearest Neighbour Interpolation
- Bilinear Interpolation
- Bicubic Interpolation

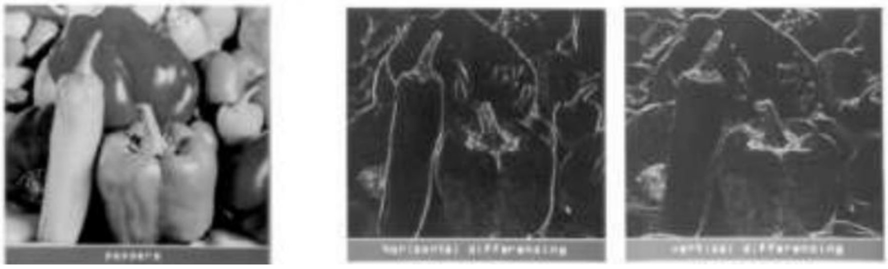
Q4. Develop codes in python to obtain the filtered imaged from the noisy image given to you as input ‘Noisy image’ in the database. Show the steps involved to be applied for filtering and explain why?



Noisy image Filtered image

Q5. Take any one image from the given database and compute:

- First order derivative along ‘x’
- First order derivative along ‘y’
- Gradient image magnitude and followed by thresholding to output edge map.
- Edge map computed using second order derivative
- Edge map using LoG and Canny edge detector



(a) Original image (b) Horizontal magnitude (c) Vertical magnitude

Please Note: The images included in this PDF serve as examples only. Utilize only the images provided in the corresponding database zip file, for each question except Q2 there is a sub folder as Q1, Q3, Q4 and so on.