

E9 241 DIP - Assignment 03

October 29, 2021

Due Date: November 14, 2021

Total Marks: 90

Instructions:

- For all the questions, write your own functions. Use library functions for comparison only.
- Your function should take the specified parameters as inputs and output the specified results.
- Also provide the wrapper/demo code to run your functions. Your code should be self contained i.e. one should be able to run your code as-is without any modifications.
- Vectorize your code. Non-optimized code may be penalized.
- For python, if you use any libraries other than numpy, scipy, scikit-image, opencv, pillow, matplotlib, pandas and default modules, please specify the library that needs to be installed to run your code.
- Along with your code, also submit a PDF with all the results and answers to subjective questions, if any.
- Put all your files into a single zip file and submit the zip file. Name the zip file with your name.

Q1. Image Downsampling:

- (a) Downsample the image `barbara.tif` by a factor of 2. Downsample the image by selecting every second pixel in both directions (do not use library functions to downsample). What artifacts do you notice?
- (b) Now, first filter the image with a spatial domain Gaussian Low Pass Filter before downsampling the image. You can use a 5×5 window and $\sigma = 1$. Also experiment with different window sizes and σ values. Do you notice the mitigation of the artifacts? Compare your result with a library function.

(10+15=**25M**)

Q2. Edge Detection: For the grayscale images `Checkerboard.png`, `NoisyCheckerboard.png`, `Coins.png` and `NoisyCoins.png`,

- (a) Smooth the input image using a spatial domain Gaussian filter. You can use a 5×5 window and $\sigma = 5$.
- (b) Use the Sobel/Prewitt operator to compute the image gradients, and then compute the gradient magnitude. Use thresholding to get the edges.
- (c) For the same set of images, use Laplacian operator to detect the edges. For the noisy images, threshold the filtered image before finding the zero crossings.

Analyse the difference between first-order and second-order gradient-based edge detectors for clean and noisy images. (10+10+15=**35M**)

Q3. Interest Point Detection: For the images `Checkerboard.png` and `MainBuilding.png`,

- (a) Implement Harris corner detector. Experiment with different threshold values and report your results.

- (b) Modify the images by rotating, scaling and adding noise. Now detect corners in the modified images. Analyse the difference in performance as compared to the original images.

(15+15=**30M**)