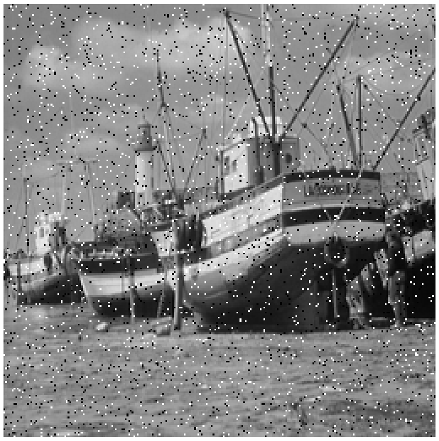
Project 2

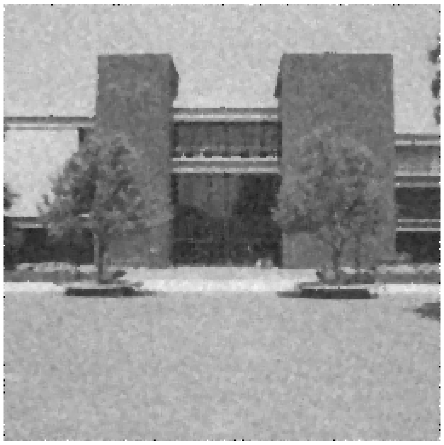
Each group should

- Submit one report per group containing procedures, results, and analysis by Feb. 2 to Canvas. Follow the same Report Guidelines as in Project 1.
- Attach your well-commented code to the report.
- Include a section describing the work done by each member of the group.
- You can use any programming language to do the project. MATLAB is preferred.

Part A – Image Filtering

In the following you will find two imperfect images (electronic versions can be found on Canvas). First analyze these images and discuss ways to find out problems in them. Then use image filtering techniques covered in class to reduce noise in these images. Display your filtered images and submit it along with the report. It should include the technique(s) you applied, comparisons of different techniques you used (include linear and nonlinear filtering, and masks of different sizes), if applied, and a discussion on the results. ***Please write your own code (i.e. no MATLAB built-in function) to implement image filtering.***





Part B – Edge Detection

Test your edge detectors on the following images (electronic versions on Canvas). For color images, convert them to gray-scale images before edge detection.

1. Implement the Prewitt Operator and the Sobel Operators. Discuss which operator gives better results. In your report, include the followings for *one* test image.
    a. The original image.
    b. Compute and display the sum of squared gradient magnitude.
    c. Select a threshold that produces the best edge map and display it.

2. Implement the LoG Operator. Consider the masks approximating the LoG with σ = 0.5, 1, and 1.2. Discuss the effects of mask size on the output. In your report, include the followings for one test image.
    a. The original image
    b. Output image from LoG with various values of σ
    c. Determine the σ value that produces the best edge map.
3. Apply the Canny Edge Detector to the images. You can use MATLAB built-in function for this step. Discuss the effects of threshold and σ on the output.
    a. The original image
    b. Output image with several threshold and σ values for one test image, then show the best results for each of the four test images.
4. Using outputs from the optimal operator from the above edge detectors, apply the Hough Transform to extract lines from the test images. You can use MATLAB built-in function for this step. Discuss your method for selecting likely lines from the Hough Space. In your report, include results for all four test images.
    a. the original image
    b. image of detected lines overlaid onto original image

- You can earn extra credit if you implement your own Canny Edge Detector and Hough Transform.