CMPE 465: Intoduction to Computer Vision

Homework 0 Report

Efe Berk ERGULEC

Instructor: Venera ADANOVA

October 7, 2019

Introduction

Introduction of Computer Vision lecture includes various application techniques using MATLAB Programming Language. Application of fundamental image processing methods is the first step of Computer Vision. In this homework, my responsibility is to implement these methods and make experiments with them. Mostly filtering methods asked to be used in this assignment. This report includes discussion and comparison of experiment results. There are three sections, each represents answer of one question.

Image that used in report has 256x256 pixel size.

Question 1: Adding Noise

This question has mere steps to implement. First, I imported an image and convert it to grayscale image. After that, I implemented noise function to do noise operation from MATLAB package. These operation gave me a noisy image. Comparison of original and noisy image has shown below (Figure 1).

Figure 1: Without Noise vs. With Noise

Functions implemented during this part explained below:

- `imread()`: Reads image from directory and turns it into a 3-by-height-width 3D matrix.
- `rgb2gray()`: Takes imported image as input and turns it into grayscale image.
- `imnoise()`: Noise function that makes operation. It takes image as an input and applies filter which needs to enter as second input. For this question, gaussian filter is used.

Question 2: Removing Noise

Second question is focusing on removing filter, which is reverse of question 1. However, question asked to implement a gaussian filter that can be arrangeable with respect to two parameters: kernel size and σ (sigma) parameter. For the sake of test efficiency, I wrote source code to make experiments with all permutations of kernel size and σ parameters. I want to explain my test case to help you percept the algorithm.. I wrote four different filter size and five different σ value into two different arrays. My test code gives 20 different output, which helps experimenter to analyze photos better.

Before discussing the results, I want to explain implementation steps and methods used. Implementation includes everything from question 1 because I need a noisy image to turn it again. That's why first step of this implementation consist of first question. Implementing variables is the second step of my code. These implementations include filter size array, σ array and image filter array that keeps all outputs. Last step is to plot all permutations into figure.

Results of my experiment provided in next page (Figure 2). There are 20 different images in Figure 2 with their filter size and σ value above. In addition to that, code gives a figure for observation. In order to see results better, my personal recommendation to you is to expand this figure.

Functions used in first question also used in second question. In addition to them, functions implemented during this part explained below:

- `fspecial()`: This function is using for removing noise from image and takes two parameter. It is reverse of `imread()` function. First input that it takes is filtering method. For this homework, gaussian filter is asked to implement. Second input is σ value that corresponds to standard deviation.

Discussion of Results

There were four different filter sizes and five different sigma values in my test case. I observed general and specific details that affects image. Next paragraphs will cover every aspect of my comments.

First variable I want to talk about is filter size. When I saw outcomes, I definitely felt certain about one result: filter size is dependent to picture size and needs to arrange well in order to remove noise. For example, when filter size is too small, filter doesn't have enough efficiency and noise on image mostly remains. On the other side, if filter size is larger than expected, filter makes image a bit blurry. Thus, arranging filter size has a huge importance for image.

Secondly, I want to talk about sigma. Sigma values are critical to make image smoother and results show me that sigma has significant effects in range $0 \leq \sigma \leq 3$. To prove that, there are notable differences between images where σ values are 0.5 and 1 at Figure 2. Also, $\sigma = 0.5$, which is default value for gaussian filter has most effective results. Smoother picture group is second column in Figure 2 and image with [$\sigma = 0.5$ / filter size $= 5$] is the most smooth picture of output.

Hence, choosing filter size with respect to size of image and using default σ size gives best results for removing noise.

Figure 2: Results with respect to different sigma values and filter sizes

Question 3: Applying Different Filters Onto Image

Last question covers implementation of spatially filtering operations to image and observing their results to resolve their use cases. There are five different filter matrices to test written below:

$$H1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} H2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix} H3 = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} H4 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} H5 = \frac{1}{9} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

In addition, the general formula is given to use in our test.

$$G(x, y) = I(x, y) * H(x, y)$$

H stands for filter function and I is grayscale image (image function) that will be manipulated. Result of this operation G is is the filtered image function.

First phase of my code in this question follows exactly same procedure with first question except noise. Second step is implementation of filter matrices and last step makes five different filtering operations that mentioned in the question using `fspecial()` function. Results of these operation shown below (Figure 3):

Figure 3: Original Picture vs. Filters (upper-left image is original)

All methods that I used in this question includes functions of first and second question. That's why I didn't put any functions used section in there.

Discussion of Results

There are six different images in Figure 3 that indicates original image and results of five different operations. Upper row includes three pictures from left to right: Original Image, H1 filter result and H2 filter result. Lower row includes three pictures from left to right: H3 image result, H4 filter result and H2-H5 filter result. Explanation of all images and filters given below:

- **H1 Filter:** From mathematical approach and results provided in Figure 3, it is certain that this filter only maintains picture. This filtering must gives identical image as a result because it doesn't accept any pixel value except pixel that will change. There are two different pooling techniques, average pooling and max pooling. Both pooling technique need to give same number as a theory because you cannot add 0 to average and maximum number in this matrix needs to be (2,2) which is the center of matrix. Hence, I conclude that this filter can only keeps an image same.
- **H2 Filter:** This filter is using to polish image. To clarify my conclusion with mathematics, pixel numbers of grayscale image must be between 0 and 255, also from black to white. If you put your image into this filter, it multiplies every pixel by two and it makes filtered image brighter.
- **H3 and H4 Filter:** H3 filter shifts picture to left by 1 pixel and H4 does same operation to right side. If you look at Figure 3, you can clearly see a black line between those filter. This black line appeared because mathematical approach takes value on the left side and assigns it to center recursively. H4 does the same thing but takes value on the right side. Hence, it is certain that H3 shifts 1 pixel to right and H4 shifts 1 pixel to left.
- **H5 Filter:** There wasn't any questions that asks to implement this filter individually, but mathematical theory supports that it blurs the image. There are nine same values in this filter matrix and when it enters into test, it only makes calculations to adduct values with each other. This process makes image blurry.
- **H2-H5 Filter:** These two filters gave an impact that I didn't expect and made image sharper. However, when I thought about theory of them, results became reasonable to me.

Conclusion

To summarize, fundamental image processing methods (mostly filtering methods) has implemented in this homework. First question includes adding noise. Second question concentrates on how to remove noise and not to broke its format. Last question helps to understand theory behind filters by implementing them on practice and analyze their results. It was also useful to introduce MATLAB and syntax of it. Thus, in my opinion this homework is very helpful to satisfy all expectations that you wanted us to learn.