CIE 6004(Fall 2019) Assignment 1

Author

Name: Jian Zhang
Student ID: 219012058
MSc Program: Data Science

October 6, 2019

Contents

1 Introduction

This report is about the Assignment 1 of "Image Processing and Computer Vision". We mainly discuss the spatial and intensity resolution, image enhance and denoising. For most parts of the exercise, the function is implemented by myself instead of applying the function of library. And my code is based on Python.

The report includes the description of the problems, the algorithm to solve them and the result of the resolution.

2 Exercise

2.1 Spatial and Intensity Resolution

(a) Make a program to downsample an image to 1/2, 1/4 and 1/8, and implement a function to recover the original size by bilinear interpolation.

To downsample an image is to reduce the size of the image. For different new sizes, we jump over some pixels to generate a new image.

Biliear interpolation is a resampling method that uses the distanceweighted average of the four nearest pixel values to estimate a new pixel value. The four cell centers from the input raster are closest to the cell center for the output processing cell will be weighted and based on distance and then averaged.

	X1	X	X2
Y1	Q11		Q21
Y		P	
Y2	Q12		Q22

$$P \approx \frac{(x_2 - x)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} Q_{11} + \frac{(x - x_1)(y_2 - y)}{(x_2 - x_1)(y_2 - y_1)} Q_{21} + \frac{(x_2 - x)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} Q_{12} + \frac{(x - x_1)(y - y_1)}{(x_2 - x_1)(y_2 - y_1)} Q_{22}$$

Figure 1: Bilinear Interpolation

We can apply downsample function and resample function to three different images(Face.png, Cameraman.png, Crowd.png) in different sizes. To make the report more readable, I only take three compared results as example.

The following figures compare the downsampled image and the resampled image.

Figure 2: Face.png downsample by 1/2 and recover

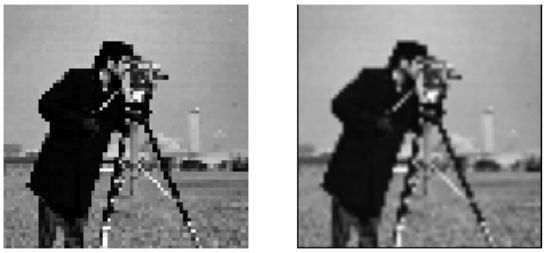

Figure 3: Cameraman.png downsample by 1/4 and recover

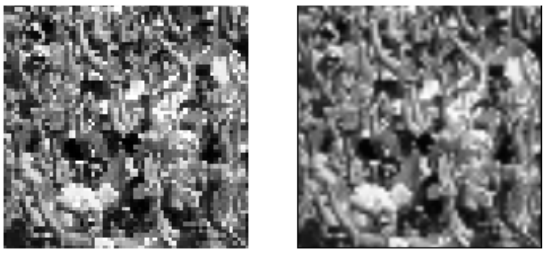

Figure 4: Crowd.png downsample by 1/8 and recover

From the results, we can see that if we downsampled smaller size, then we lost more information of the image. And it is harder to recover an image with less information. Moreover, if the original image has more details like 'Crowd.png', we can hardly recognize people's faces after resampling.

(b) Make a program to reduce the quantization level of an image to 1/2, 1/4 and 1/8 and compare the results with(a).

To reduce the quantization level is to turn the gray level of each pixel into smaller scale. The algorithm is to do a loop for the array of image and the gray level of each pixel is divided by the reducing rate.

The following figures compare the downsampled image in (a) and the image which the quantization level is reduced with the same rate.

Figure 5: Face.png downsample and reduce quantization level by 1/2

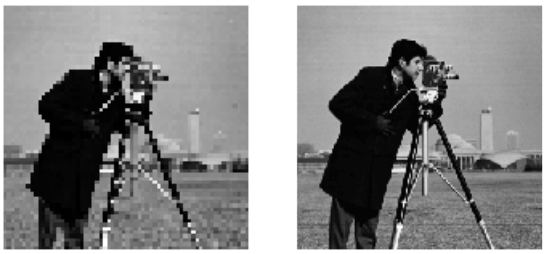

Figure 6: Cameraman.png downsample and reduce quantization level by 1/4

Figure 7: Crowd.png downsample and reduce quantization level by 1/8

From the results, we get blur images after downsampling, while we get images with less contrast ratio after reducing quantization level. Overall, we both lose information of the images after applying two methods.

2.2 Image Enhance and Denoising

(a) Implement the histogram equalization.

Histogram Equalization is a technique to adjust contrast levels and expand the intensity range in a digital image.

The algorithm steps for histogram equalization:

Step1: Find histogram of input image, and find histogram equalization mapping;
Step2: Specify the desired histogram, and find histogram equalization mapping;
Step3: Build lookup table: For each gray level, find and then find a j level so that best matches: Setup a lookup entry lookup[i] = j.

The following figure compare an original image and the histogram equalization image.

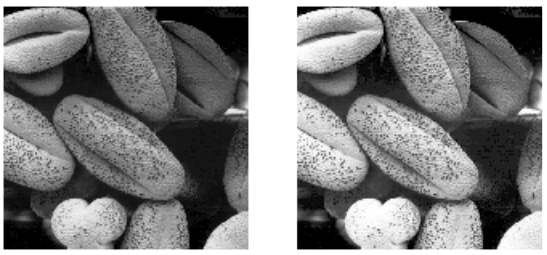

Figure 8: Histogram Equalization

(b) Implement the sharpening spatial filter and apply to the image 'Spatial Filtering.png'

Spatial filtering is an image processing technique for changing the intensities of a pixel according to the intensities of the neighboring pixels.

To generate a new image by applying laplacian operator to each pixel.

Figure 9: Spatial Filtering

(c) Apply frequency filtering to denoise the given image 'Frequency Filtering.png'.

We have many filtering methods to deal with frequency filter, such as mean filter, median filter and gaussian filter.

The Gaussian smoothing operator is a 2-D convolution operator that is used to 'blur' images and remove detail and noise. In this sense it is similar to the mean filter, but it uses a different kernel that represents the shape of a Gaussian ('bell-shaped') hump.

Applying different filtering methods in 'opencv' to denoise the given image. Compare the results below with those achieved by other filters. Gaussian filtering is more effective at smoothing this image.

The following figures show the original image and the image after applying Gaussian smoothing operator.

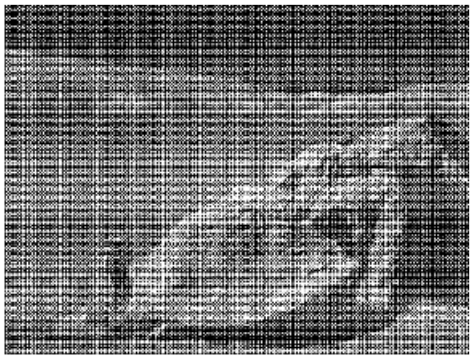

Figure 10: Original image

Figure 11: Frequency Filter: Gaussian Smoothing