Computer Engineering Department
TED University

TED UNIVERSITY

Spring 2023 – CMPE 362 / CS 563 Digital Image Processing
Assignment 2
Due Date: April 30th, 2023 at 23:59

This assignment is to be done individually!

Part 1 [50 pts]:

In this part, you are going to implement template matching using the following four different measures that we have seen in class:

- Correlation

$$\sum_{x=1}^{m}\sum_{y=1}^{n} P(x,y) * T(x,y)$$

- Zero-mean Correlation

$$\sum_{x=1}^{m}\sum_{y=1}^{n} P(x,y) * (T(x,y) - T_{mean})$$

where T_{mean} is the mean intensity of the template image T

$$T_{mean} = \frac{1}{m*n}\sum_{x=1}^{m}\sum_{y=1}^{n} T(x,y)$$

- Sum of Squared Difference

$$\sum_{x=1}^{m}\sum_{y=1}^{n} (P(x,y) - T(x,y))^2$$

- Normalized Cross Correlation

$$\frac{\sum_{x=1}^{m}\sum_{y=1}^{n}(P(x,y) - P_{mean}) * (T(x,y) - T_{mean})}{\left[\left(\sum_{x=1}^{m}\sum_{y=1}^{n}(P(x,y) - P_{mean})^2\right) * \left(\sum_{x=1}^{m}\sum_{y=1}^{n}(T(x,y) - T_{mean})^2\right)\right]^{0.5}}$$

where T_{mean} is the mean intensity of the template image T

$$T_{mean} = \frac{1}{m*n}\sum_{x=1}^{m}\sum_{y=1}^{n} T(x,y)$$

and P_{mean} is the mean intensity of the image patch P

$$P_{mean} = \frac{1}{m*n}\sum_{x=1}^{m}\sum_{y=1}^{n} P(x,y)$$

Implement the following functions each takes an image patch `P` and a template image `T` (where `P` and `T` are of the same size) and returns their similarity/dissimilarity value `val`.

```
def correlationMeasure(P,T):

def zeroMeanCorrelationMeasure(P,T):

def sumOfSquaredDifferenceMeasure(P,T):

def normalizedCrossCorrelationMeasure(P,T):
```

Note that you should convert data types of `P` and `T` to `np.float64` in order to get a correct result.

Perform the following operations.

A. Take the image input.png and template.png (can be downloaded from Moodle).
   1. In order to find the location of the template image in the larger input image, compare the template and neighborhood of each pixel in the input image. This will result in an image with the same size of the input image. Do that comparison using each of the four different measures, namely, correlation, zero mean correlation, sum of squared difference, and normalized cross correlation. Display the resulting images using jet color map.
      - You might skip doing comparison for the pixels close to the boundary of the input image. If the size of the template image is (2a+1 x 2b+1) and the size of the input image is MxN, you might skip doing comparison for the pixels of the input image at rows [0, a-1] and [M-1-a, M-1] and at columns [0,b-1] and [N-1-b,N-1].
      - You need to implement template matching yourself using the following functions you have implemented.
        - `correlationMeasure`
        - `zeroMeanCorrelationMeasure`
        - `sumOfSquaredDifferenceMeasure`
        - `normalizedCrossCorrelationMeasure`
   2. Since correlation, zero mean correlation, and normalized cross correlation are similarity measures, find the pixel whose measure is the maximum. Since sum of squared difference is a dissimilarity measure, find the pixel whose measure is the minimum. Show the neighborhood of those pixels using `rectangle` function of OpenCV. In order to find the location of max/min value, you could use `minMaxLoc` function of OpenCV.
   3. Discuss the results. Which measures give the correct matching and which measures give the incorrect matching? Why?

B. Now, take the image input2.png and template.png (can be downloaded from Moodle).
   1. Repeat the same procedure as in the previous step.
   2. Discuss the results. Which measures give the correct matching and which measures give the incorrect matching? Why?

Part 2 [50 pts]:

In this part, you are going to perform image filtering in frequency domain using Butterworth filter.

Formulation of Butterworth low pass filter is as follows:

$$H_{LowPass}(u,v) = \frac{1}{1 + [D(u,v)/D_0]^{2*n}}$$

Here, $H_{LowPass}$ denotes Butterworth low pass filter shifted to center. $D(u,v)$ denotes Euclidean distance of (u,v) to the center $(h/2, w/2)$ where h and w are height and width of H. D_0 and n are parameters determining shape of the filter.

Butterworth high pass filter can be formulated simply as follows:

$$H_{HighPass} = 1 - H_{LowPass}$$

Implement the following function that takes input image `img`, parameters `D0` and `n`, applies low pass and high pass filtering and returns the resulting image.

```
def butterworthLowpassFilter(img, D0, n):

def butterworthHighpassFilter(img, D0, n):
```

Perform the following operations.

A. Apply Butterworth low pass filter:
   1. Take image Lenna.png (can be downloaded from Moodle) and show it.
   2. Compute its Fourier transform F and show its magnitude.
   3. Choose two different values for the parameter n as 2 and 10.
   4. Choose two different values for the parameter D_0 (one is smaller and the other one is larger).
   5. For each pair of values for the parameters n and D_0 (notice that there are four pairs of values in total), show the filter $H_{LowPass}$, magnitude of the filtering result in frequency domain, and the filtering result in spatial domain.
   6. Discuss the four different filtering results you have obtained.
      i. What is the effect of the parameter n?
      ii. What is the effect of the parameter D_0?
      iii. Have you observed ringing effect in any of the results?

B. Apply Butterworth high pass filter:
   1. Take image Lenna.png (can be downloaded from Moodle) and show it.
   2. Compute its Fourier transform F and show its magnitude.
   3. Use the same pair of values for the parameters n and D_0 as in the low pass filtering.
   4. For each pair of values for the parameters n and D_0, show the filter $H_{HighPass}$, magnitude of the filtering result in frequency domain, and the filtering result in spatial domain.
   5. Discuss the four different filtering results you have obtained.
      i. What is the effect of the parameter n?
      ii. What is the effect of the parameter D_0?

iii. Have you observed ringing effect in any of the results?

Important note for Part 2: While showing your results in the frequency domain, shift them to the center.

What to hand in:

- Code
  For each part and subpart, write your code into an mlx file so that we are able to see your results without running them.
  - `part1a.ipynb`
  - `part1b.ipynb`
  - `part2a.ipynb`
  - `part2b.ipynb`

  Note that `part1a.ipynb` and `part1b.ipynb` should contain definitions of the following functions:
  `correlationMeasure`
  `zeroMeanCorrelationMeasure`
  `sumOfSquaredDifferenceMeasure`
  `normalizedCrossCorrelationMeasure`
- Report
  In your report, you are expected to include the following information:
  - Brief explanation of what you have done
  - The results that you have obtained together with the corresponding parameters
  - Discussion of the results

It is important that you include all your results and discussion in the report.