Fatih Mehmet Yiğitel

CMPE 362 HW 2

PART 1

In part A

1 – code implementation

2 –code implementation

This code performs template matching between an input image and a template image using four different measures of similarity/dissimilarity: correlation, zero-mean correlation, sum of squared differences (SSD), and normalized cross-correlation (NCC). The aim of template matching is to find the location in the input image where the template image appears, or is most similar to.

The four different measures of similarity/dissimilarity are defined as functions: correlationMeasure, zeroMeanCorrelationMeasure, sumOfSquaredDifferenceMeasure, and normalizedCrossCorrelationMeasure. Each function takes two input images as arguments (P and T), and returns a single scalar value that represents the similarity/dissimilarity between the two images.

Each of the four measures of similarity/dissimilarity is computed between P and the template image, and the resulting score is stored in the corresponding pixel of result_image.

Finally, the pixel with the maximum similarity/dissimilarity score in result_image is found using the unravel_index and argmax functions from numpy. If SSD is the measure used and there are multiple pixels with the minimum score, the first minimum will be used.

A rectangle is drawn around the matched template in the input image using cv2.rectangle. The location of the rectangle is determined by the maximum score pixel in result_image. The input image with the matched template rectangle highlighted is displayed using cv2.imshow.

3 - The results may show that zero-mean correlation and normalized cross-correlation measures provide more accurate matching, while the correlation measure and sum of squared difference measures might be less accurate. Zero-mean correlation and normalized cross-correlation are generally more robust to variations in lighting.

In part B

1- code implementation
   This is the same code as the part1a
2- ?

PART 2

This code is implementing Butterworth lowpass and highpass filters to an input image and displaying the filtered images along with their magnitude spectra.

The butterworthLowpassFilter function takes an input grayscale image img, a cutoff frequency D0, and an order n as input arguments. It applies a Butterworth lowpass filter to the input image in the frequency domain using the provided parameters, and returns the filtered image in the spatial domain.

The butterworthHighpassFilter function takes the same inputs as butterworthLowpassFilter, but applies a Butterworth highpass filter to the input image. It first applies a lowpass filter to the image using the butterworthLowpassFilter function and then subtracts the filtered image from the original image to obtain the highpass filtered image.

The code reads an input image in grayscale using OpenCV's imread function and displays the original image and its magnitude spectrum using show_image_and_spectrum. It then applies Butterworth lowpass filters with various values of n and D0 to the input image using a loop, and displays the resulting filtered images and their magnitude spectra using show_image_and_spectrum.

The params list contains the different parameter combinations to use for the lowpass filter, and the loop iterates over each set of parameters, applies the filter, and displays the filtered image and its magnitude spectrum.

A)

6) – 1

In this code, the parameter n is the order of the Butterworth filter. It determines the smoothness of the transition between the passband (the frequency range that is allowed to pass through the filter) and the stopband (the frequency range that is attenuated or blocked by the filter).

A higher value of n results in a sharper transition between the passband and the stopband, leading to better frequency domain filtering.

6) - 2

In the code, D0 is the cutoff frequency of the Butterworth filter. A higher D0 allows more high-frequency components to pass, resulting in less blur for lowpass filters and less detail for highpass filters. A lower D0 causes more blur in lowpass filters and more detail in highpass filters.

6) - 3

That ringing artifacts may occur when applying a filter with a sharp transition between the passband and the stopband, such as a Butterworth filter with a high order n.

B)

5) - 1

In the given code, the parameter n is the order of the Butterworth filter. The value of n determines the sharpness of the cutoff frequency in the frequency domain.

A higher value of n will result in a sharper cutoff frequency, meaning that the high frequencies beyond the cutoff will be attenuated more quickly. Conversely, a lower value of n will result in a more gradual cutoff, meaning that the high frequencies beyond the cutoff will be attenuated more slowly.

5) - 2

D0 controls the amount of detail preserved in the image after the filtering process. A small value of D0 will remove more high-frequency details and preserve only low-frequency components, resulting in a smoother image, while a larger value of D0 will retain more high-frequency details and preserve more of the original image structure.

Therefore, the value of D0 controls the trade-off between smoothing and preserving details in the image.