OpenCV Exercise

1 Installation and Tutorials

Install OpenCV-python in Windows:
https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_setup/py_setup_in_windows/py_setup_in_windows.html#install-opencv-python-in-windows

OpenCV-Python Tutorials:
https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_tutorials.html

2 Object Detection using Template Matching

In this section, we are going to write a template matching algorithm for object detection with OpenCV – python. The template matching algorithm detects the object by finding the part of an image that matches best with the template. For example, the template image can be a photo of an object, and the source image can be a photo of a person holding the same object. Then the template matching algorithm will slide the template image inside the source image, and at each position, the algorithm will calculate the correlation between the template image and the part of the source image. The estimated location of the object in the source image is where the correlation is maximum.

In this section, we use the normalized correlation coefficient (NCC) to define the correlation of two images of the same size. The NCC is defined as follows:

$$NCC = \frac{1}{N_{rows}N_{columns}} \cdot \frac{\sum_{k=1}^{N_{rows}} \sum_{l=1}^{N_{columns}} (v_1(k,l) - \overline{v_1})(v_2(k,l) - \overline{v_2})}{\sigma^2(v_1)\sigma^2(v_2)}$$

where N_{rows} and $N_{columns}$ are the number of rows and columns of the two images, $v_1(k,l)$ and $v_2(k,l)$ are the pixel values of image 1 and image 2 at (k,l), $\overline{v_1}$, $\overline{v_2}$, $\sigma^2(v_1)$ and $\sigma^2(v_2)$ are the means and variances of pixel values of the two images.

To complete this section, you need to:
1. Take a photo of an object and use it as the template image;
2. Take a photo yourself holding the same object and use it as the source image. Downsize the source image for shorter running time. (Recommended size: 640×480);
3. Resize the template image to roughly the same size as the object in the source image;
4. Download TemplateMatching.py and complete the code. For simplicity, convert both images into grayscale before processing;
5. Run the code and see the detection results. It takes some time to run the algorithm.

An example of the template matching result is shown as follows:

Template Image

Source Image

Template Matching Result