Image Processing by CV2

Shuwei Li, shwli@bu.edu

In this report, I will focus on the image processing part of DSP (Digital Signal Processing) in Python, especially in the package called py-opencv. Opencv could be the most popular library that scientists are using in daily research. I went through how to display a image, adaptive thresholding, edge detection, counting objects and face detection, which are among the most useful tools in digital signal processing.

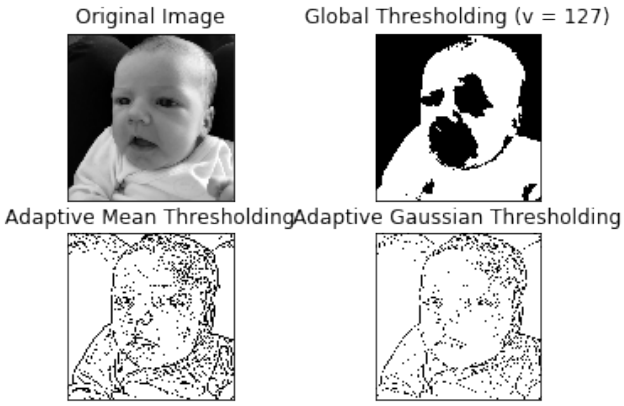
digital signal processing | image processing | python | opencv

Display

The function to read from an image into OpenCv is imread(). imshow() is the function that displays the image on the screen. The first value is the title of the window, the second is the image file we have previously read. cv2.waitKey(0) is required so that the image window won't close immediately. It will wait for a key press before closing the image. For example, cv2.waitKey(30000) represents that the window will last 30,000 ms then close.

Adaptive Thresholding

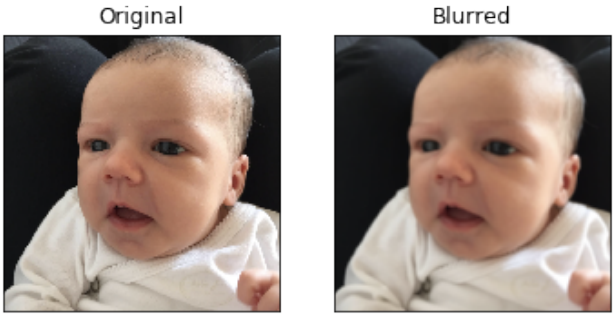
Some simple thresholding functions, like cv2.threshold, they can only do the global thresholding, which may cause the image to be blur. It is because when we use a global value as threshold value, it may not be good in all the conditions where image has different lighting conditions in different areas. In that case, we go for adaptive thresholding. In this, the algorithm calculate the threshold for a small regions of the image. So we get different thresholds for different regions of the same image and it gives us better results for images with varying illumination. The adaptive thresholding functions include cv2.ADAPTIVETHRESHMEANC and cv2.ADAPTIVETHRESHGAUSSIANC.



As the pictures shows, Global Thresholding cannot lose a lot of details of the image, while Adaptive Thresholding can overcome the unbalanced lighting conditions.

Image Smoothing

In image smoothing, cv2 also has plenty of tools. This is done by convolving the image with a normalized box filter. It simply takes the average of all the pixels under kernel area and replaces the central element with this average. This is done by the function cv2.blur() or cv2.boxFilter(). Check the docs for more details about the kernel. We should specify the width and height of kernel. A 3x3 normalized box filter would look like this:



As the pictures shows, the image is smoothed by a 5*5 kernel using Averaging algrithm.

Edge Detection

OpenCV puts all the edge detection algorithm in single function, cv2.Canny(). It is the size of Sobel kernel used for find image gradients. By default it is 3. Last argument is L2gradient which specifies the equation for finding gradient magnitude.

The parameters is set to be 10-30, which represents relatively more edges will be detected.

The parameters is set to be 50-100, which means only the very obvious edges can be detected.

Color Space

There are more than 150 color-space conversion methods available in OpenCV, including the most widely used ones, BGR to Gray and BGR to HSV.

For color conversion, we use the function cv2.cvtColor(inpuimage, flag) where flag determines the type of conversion.

For BGR to Gray conversion we use the flags cv2.COLORBGR2GRAY. Similarly for BGR to HSV, we use the flag cv2.COLORBGR2HSV.

There is no image to show the effect. The code should looks like: flags = [i for i in dir(cv2) if i.startswith('COLOR')]