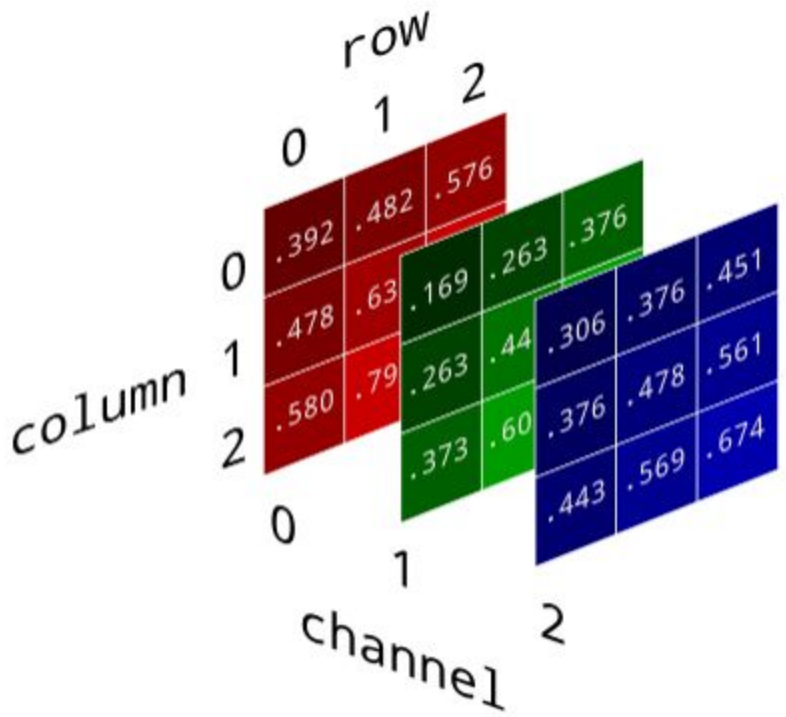
RGB Image to Grayscale Conversion And Edge detection

RGB image


row
0
1
2
column
0
1
2
.392
.482
.576
.478
.63
.580
.79
.169
.263
.376
.263
.44
.373
.60
.306
.376
.451
.376
.478
.561
.443
.569
.674
0
1
2
channel

RGB image

When reading a color image file, OpenCV `imread()` reads as a NumPy array `ndarray` of `row (height) x column (width) x color (3)`. The order of color is BGR (blue, green, red)

Grayscale image

`Gray' color is one in which the red, green and blue components all have equal intensity in RGB space, and so it is only necessary to specify a single intensity value for each pixel, as opposed to the three intensities needed to specify each pixel in a RGB image.

In open cv an image with 2 dimension (height x width) array can be view as RGB image in which R-plane , G-plane and B-plane all have equal value (gray color image).

RGB Image to Grayscale Conversion

Generally we can take average of R-value , G-value and B-value of an RGB image corresponding to each pixels.

Gray value = (R-value + G-value + B-value) / 3 [Eq. 1]

RGB Image to Grayscale Conversion

But

To our eyes green looks about ten times brighter than blue. Through many repetitions of carefully designed experiments, psychologists have figured out how different we perceive the luminance or red, green, and blue to be. They have provided us a different set of weights for our channel averaging to get total luminance.

$$\text{Gray value} = 0.2126R + 0.7152G + 0.0722B \quad \text{[Eq. 2]}$$

RGB Image to Grayscale Conversion

We are able to see small differences when luminance is low, but at high luminance levels, we are much less sensitive to them. In order to avoid wasting effort representing imperceptible differences at high luminance, the color scale is warped, so that it concentrates more values in the lower end of the range, and spreads them out more widely in the higher end. This is called gamma compression.

To undo the effects of gamma compression before calculating the grayscale luminance, it's necessary to apply the inverse operation, gamma expansion.

After gamma expansion grayscale image approximate to

$$\text{Gray value} = 0.299R + 0.587G + 0.114B \qquad \text{[Eq. 3]}$$

Edge detection

Find a note here for edge detection.

Find a tutorial here for canny edge detection.

Canny Edge Detection

The Canny edge detection algorithm is composed of 5 steps:

1. Noise reduction;
2. Gradient calculation; [Apply Sobel filter on each pixel of image]
3. Non-maximum suppression;
4. Double threshold;
5. Edge Tracking by Hysteresis.

The prerequisite is to convert the image to grayscale before following the above-mentioned steps.

Link for a grayscale image: Pigeon

Noise reduction

It decrease the possibility of generation of non-relevant edge during Gradient calculation (Ex. Edge for a point)

Apply gaussian filter of size 3 x 3 on each pixel of image. For preventing size of image to be reduce ,padding can be apply.

Gradient calculation

Apply Sobel filters to find Ix and Iy on each pixel of image

Output 1 : A matrix of image size which store Gradient value(itensity) |G| of corresponding pixels.

Output 2 :A matrix of image size which store Gradient direction $\theta(x, y)$on corresponding pixels

$$|G| = \sqrt{I_x^2 + I_y^2},$$

$$\theta(x, y) = arctan\left(\frac{I_y}{I_x}\right)$$

Non-Maximum Suppression

Previous step output image may have thick edges. So for making edges thin we apply Non-Maximum Suppression.

Each pixel (except border pixels) of image have 8 neighbour pixels. Each pixel have gradient . So in the direction of gradient and opposite direction of gradient there are two pixels. If any of these two pixels have their intensity |G| greater than the the given pixel , then value of given pixelset to ziro.

.

Double thresholding and Edge Tracking by Hysteresis

Output image of previous step may have pixel values b/n 0 to 255. We need only two values 0 and 255 for edge detection.

Double thresholding and Edge Tracking by Hysteresis step will make the image with only two pixels values (0 and 255).

Today’s task (17-02-2021)

Download Pigeon

Apply each step of Canny Edge detection here .

Padding of Image

In some Edge detection techniques a filter (Kernel) is applied. So for enble this filter on all the pixels of image ,we can add some extra rows and columns.

In given example image is padded with 0. For edge detection it can be padded with adjacent pixel values.

50	50	50	50	50
50	50	50	100	100
50	100	100	100	100
50	100	100	100	100
100	100	100	100	100

0	0	0	0	0	0	0
0	50	50	50	50	50	0
0	50	50	50	100	100	0
0	50	100	100	100	100	0
0	50	100	100	100	100	0
0	100	100	100	100	100	0
0	0	0	0	0	0	0

Assignment 3.1

Make a python code for RGB Image to Grayscale Conversion of a color image(download any small size (max height =50 , max width =50)) using above 3 weighting formula.

(Submit ipynb format) (23 feb)

Assignment 3.2

Apply canny edge detection on following grayscale image.Apply each step of canny edge detection. Apply padding if needed.

50	50	50	50	50
50	50	50	100	100
50	100	100	100	100
100	100	100	100	100
100	100	100	100	100

(Submit hand written pdf format) (23 feb)