Getting and Setting Pixels

In the Previous chapter we seen that, Images are Simply Numpy arrays.

Normally, We express the image as RGB (Red, Green, Blue)

But in opencv we represent BGR (Blue Green Red)

The Reason being it was set as Standard when opencv was developed. So we are following the same Standard.

1) Now lets say that we want to know the Pixel value at a Position

For example → Given an input image, what is the RGB Value on a particular Co-ordinate.

```
(b, g, r) = image[0,0]
Print("Pixel Value at (0,0) - Red:{}, Green:{}, Blue{}".format(r, g, b))
```

Similarly, we can access at different position

```
image[20, 50]
image[30, 40]
```

2) Now lets say that we want to update the pixel values

Eg:- update the pixel at (50, 20) and set it to Red
(w = width, h = height)

Height → Rows
Width → Columns.

image[20, 50] = (0, 0, 255)

Typically RGB will be like → (255, 0, 0)
But since we follow BGR format → (0, 0, 255)

(b, g, r) = image[20, 50]

Print("pixel value at (20,50) - Red: {}, Green: {}, Blue {}".format(r, g, b))

3) Now instead of updating pixel by pixel, we can crop a particular area and update a particular color.

(h, w, c) = image.shape[:3]

Here are the Steps

1) Compute the Center of the image.

(ie) width and height divided by 2

$$(cX, cY) = (w//2, h//2)$$

2) Now Since we are using Numpy arrays, we can apply array slicing to grab particular Region from the image.

For example :- Lets take Top left corner of image

AS we know already that (0,0) is top left corner. Also we have the width, height. Now we can use all the information to get the Top left corner of image.

```
top_left = image[0:cY, 0:cX]
                  ↓      ↓
               Height  Width
```

```
top_right = image[0:cY, cX:w]
Bottom_right = image[cY:h, cX:w]
Bottom_left = image[cY:h, 0:cX]
```

3) Now we have 4 regions instead of just one pixel and we can update the Region to any desired Color

For example - Lets Set top left corner to Green

image[0:cy, 0:cx] = (0, 255, 0)

B G R

Captures Top left corner Area

4) Once we do that, then we can View the image with updated Color on the particular Region

```
cv2.imshow("updated Color in Top left", image)
cv2.waitKey(0)
```