**Steps to store histogram:**

1] Convert RGB into BGRHSV

2] Calculate Histogram

So now we use [**cv2.calcHist()**](https://docs.opencv.org/3.1.0/d6/dc7/group__imgproc__hist.html#ga4b2b5fd75503ff9e6844cc4dcdaed35d) function to find the histogram. Let's familiarize with the function and its parameters :

*cv2.calcHist(images, channels, mask, histSize, ranges[, hist[, accumulate]])*

1. images : it is the source image of type uint8 or float32. it should be given in square brackets, ie, "[img]".
2. channels : it is also given in square brackets. It is the index of channel for which we calculate histogram. For example, if input is grayscale image, its value is [0]. For color image, you can pass [0], [1] or [2] to calculate histogram of blue, green or red channel respectively.
3. mask : mask image. To find histogram of full image, it is given as "None". But if you want to find histogram of particular region of image, you have to create a mask image for that and give it as mask. (I will show an example later.)
4. histSize : this represents our BIN count. Need to be given in square brackets. For full scale, we pass [256].
5. ranges : this is our RANGE. Normally, it is [0,256].

3] Normalize image using cv2.NORM\_MINMAX

4] If press c then follow this

1] The functions calcBackProject calculate the back project of the histogram. That is, similarly to calcHist , at each location (x, y) the function collects the values from the selected channels in the input images and finds the corresponding histogram bin. But instead of incrementing it, the function reads the bin value, scales it by scale , and stores in backProject(x,y) . In terms of statistics, the function computes probability of each element value in respect with the empirical probability distribution represented by the histogram. See how, for example, you can find and track a bright-colored object in a scene:

2]