Image Processing for Computer Vision
Session 7

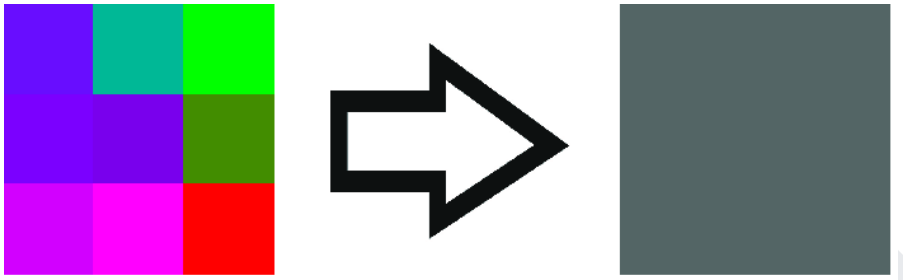
Colored Thresholding & Masking

Topics

- How grayscale
- Colored Image Thresholding
- Masking

MMZ

How grayscale conversion works?



(a) RGB pixels. (b) Grayscale pixels.

Researchegate

- Merging the three color channels (Red, Green, and Blue) into a single channel.
- Done by calculating a weighted sum of the RGB values for each pixel.
- The weights are chosen based on how the human eye perceives the intensity of each color.

Here's a common formula used for this conversion:

Gray = 0.299 * R + 0.587 * G + 0.114 * B

In OpenCV, you can convert an RGB image to grayscale while loading the image or by using the `cv2.cvtColor` function

While loading:
img = cv.imread('/berry-1.jpg', cv.IMREAD_GRAYSCALE)

Using function:
img = cv.cvtColor(img_bgr, cv.COLOR_BGR2GRAY)

Multicolored Image Thresholding

HSV Color Space

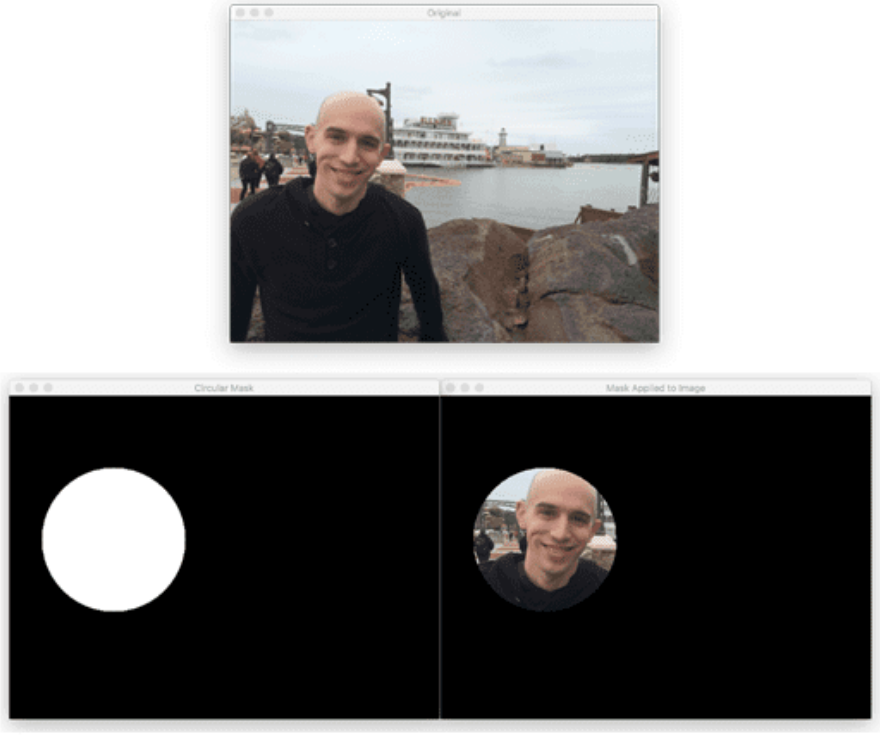
- **Hue (H)** component represents the type of color (e.g., red, yellow, green, blue),
- **Saturation (S)** represents the intensity of the color,
- **Value (V)** represents the brightness.

Steps:

1. Convert to HSV colorspace
2. Select lower and upper bound for a color
3. Create a binary mask

Masking

Mask in Image processing



PyImageSearch

A mask is a digital image that is used to hide or reveal portions of another image.

Steps of image masking:

1. **Load the image**
2. **Define a region of interest (ROI) in the Mask**
3. **Apply the mask on the original image**

AND Operation:

1*1 = 1
1*0 = 0
0*1 = 0
0*0 = 0

cv2.bitwise_and(src1, src2)

Src1 = input image; Src2 = mask

How it looks like under the hood?

```
10x10 Subset of the original image array Channel-1:
[[119 109  98  89  80  76  74  74  71  67]
 [117 103  95  89  84  83  75  69  70  78]
 [111 100  94  90  86  83  71  63  67  81]
 [104  96  93  90  85  77  67  61  66  79]
 [ 97  94  91  89  85  70  63  61  65  75]
 [102  98  92  81  70  62  62  68  77  85]
 [ 97  91  83  77  71  70  61  59  70  85]
 [ 94  87  76  69  68  78  65  57  66  88]
 [ 90  87  74  62  58  80  71  67  78  98]
 [ 78  79  71  60  57  72  75  80  92 108]]
```

```
10x10 Subset of the mask array:
[[  0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0 255 255 255 255 255]
 [  0   0   0   0   0 255 255 255 255 255]
 [  0   0   0   0   0 255 255 255 255 255]
 [  0   0   0   0   0 255 255 255 255 255]
 [  0   0   0   0   0 255 255 255 255 255]]
```

```
10x10 Subset of the resulted image array Channel-1:
[[  0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0   0   0   0   0   0]
 [  0   0   0   0   0  62  62  68  77  85]
 [  0   0   0   0   0  70  61  59  70  85]
 [  0   0   0   0   0  78  65  57  66  88]
 [  0   0   0   0   0  80  71  67  78  98]
 [  0   0   0   0   0  72  75  80  92 108]]
```