IMAGE PROCESSING

Colors image processing

- ***Mohammed Mohammed Abdalhafez***
- ***Mohammed Elsayed Abdalsalam***
- ***Abdalrahman Mohammed Gad***

Headlines

- **Colors image processing it can be used to extract some objects from the image (object identification and extraction).**
- **color image smoothing and sharpening in a digital image.**
- **Color transformation.**
- **Using color in image segmentation.**

Object identification and extraction.

- **Color difference can represent Water, hills, land, sky and trees ….**
- **HUE & SATURATION & INTENSITY features used to describe any color (HUE->RGB, INTENSITY>256 level cubed, SATURATION->brightness).**
- **Standard HD screen has 1920 pixels horizontally and 1080 vertically.**

Color image smoothing and sharpening.

- **Smoothing** used for solving errors that noticed of number pixels in any photograph where it shouldn't be.
- **Average smoothing** each pixel takes an average of the eight pixels immediately around it.

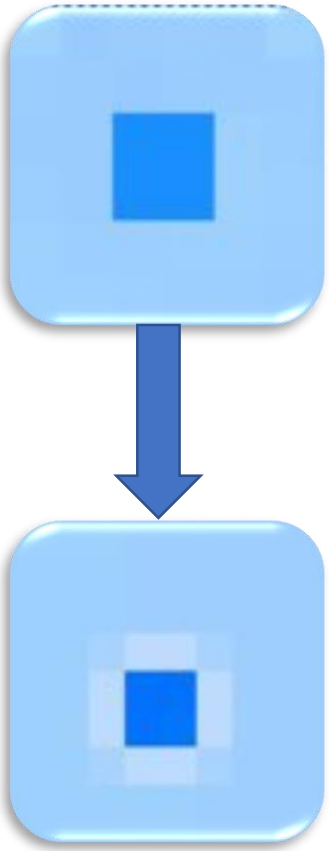
:Ex

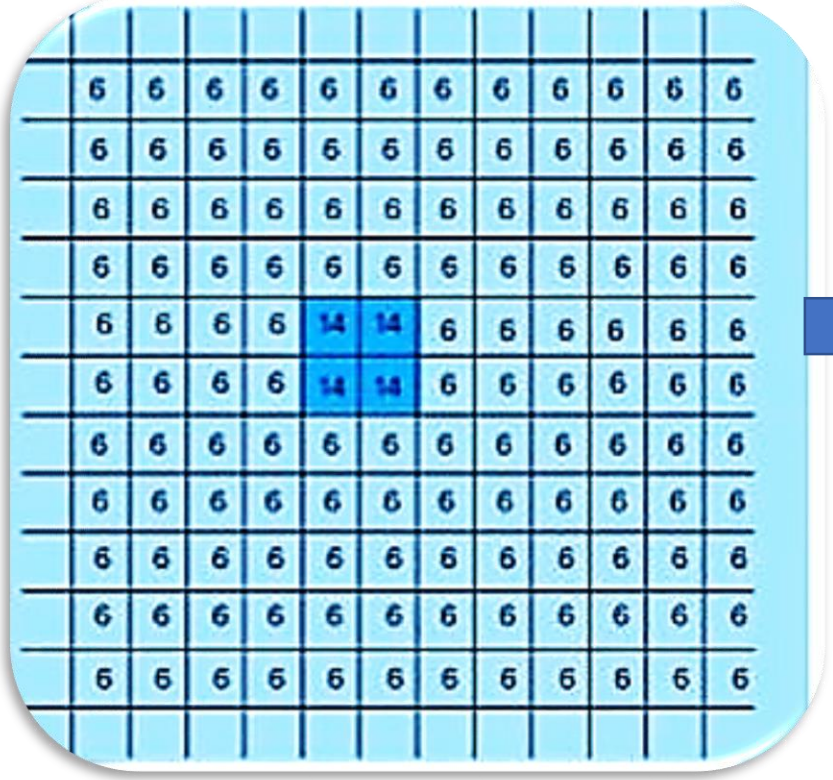
Color image smoothing and sharpening.

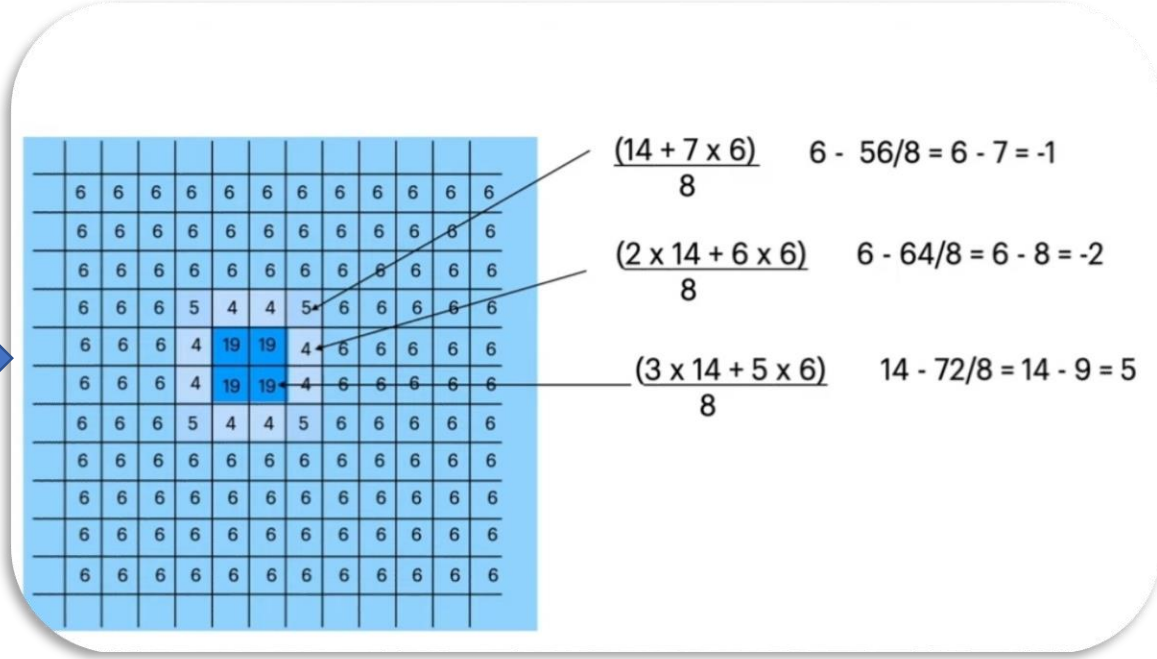
- **By applying Sharpening** for that noticed number of pixels in the last photograph where it shouldn't be.

- **Basic Sharpening** each pixel takes an average of the eight pixels immediately around it and then subtracted from the value of the pixel we are concentrating on.

:Ex

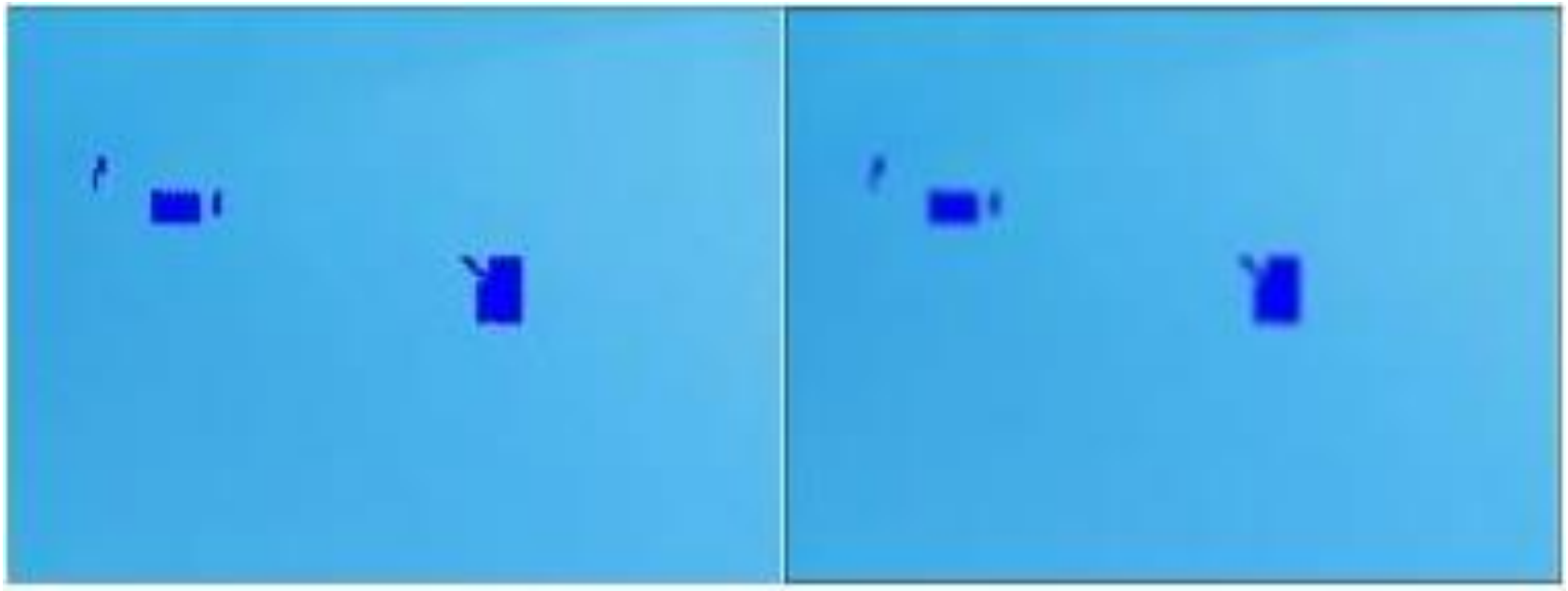



6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 14 14 6 6 6 6 6 6
6 6 6 6 14 14 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6


6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 5 4 4 5 6 6 6 6 6
6 6 6 4 19 19 4 6 6 6 6 6
6 6 6 4 19 19 4 6 6 6 6 6
6 6 6 5 4 4 5 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
6 6 6 6 6 6 6 6 6 6 6 6
(14 + 7 x 6) / 8
6 - 56/8 = 6 - 7 = -1
(2 x 14 + 6 x 6) / 8
6 - 64/8 = 6 - 8 = -2
(3 x 14 + 5 x 6) / 8
14 - 72/8 = 14 - 9 = 5

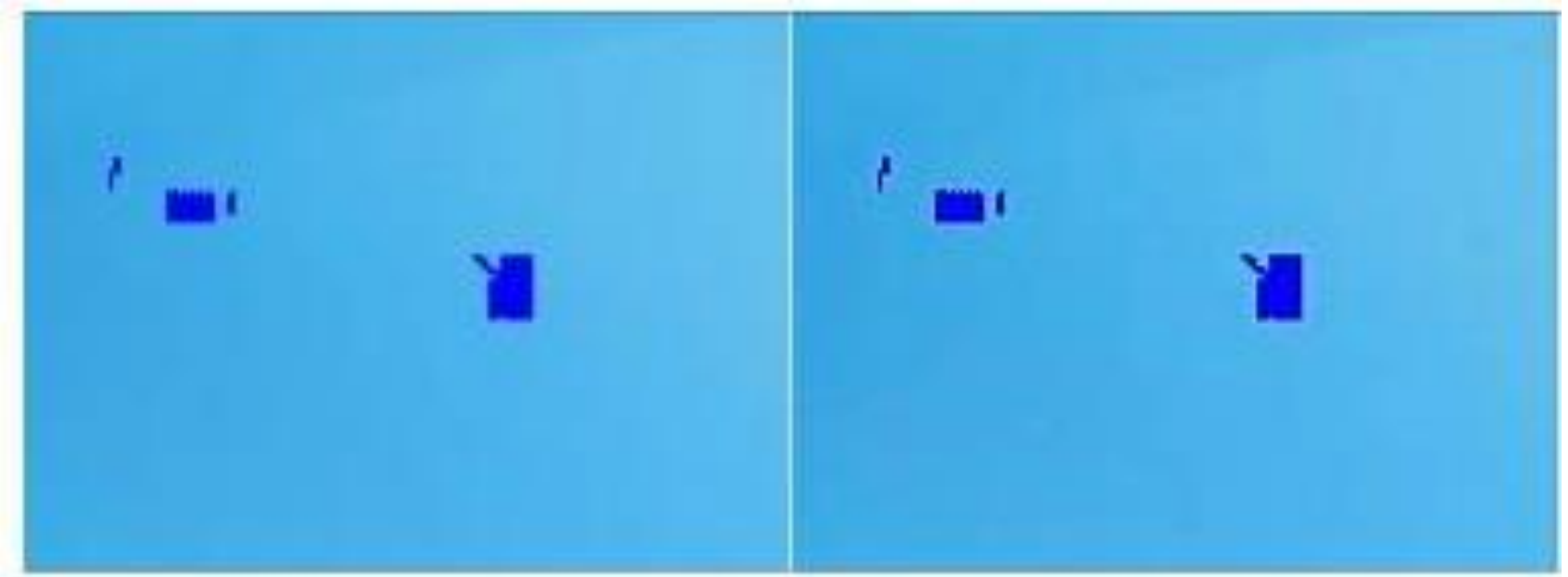
Color image smoothing using Matlab.

```
a=imread('pr.jpg');
f=fspecial('average',3);
m=imfilter(a,f);
figure,imshowpair(a,m,'montage')
```



Color image sharpening using Matlab.

```
a=imread('pr.jpg');
s=imsharpen(a);
figure,imshowpair(a,s,'montage')
```

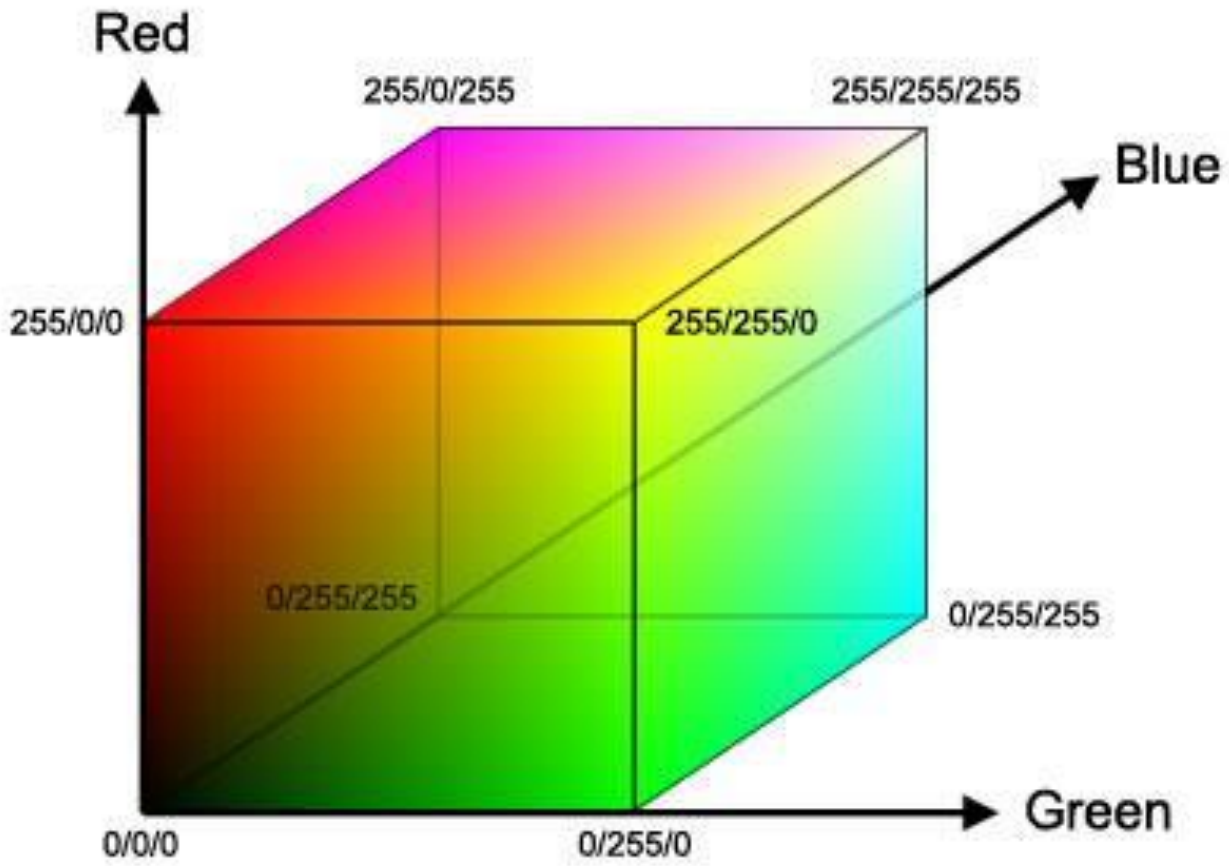


using color spaces in image segmentation

- Color spaces
- Image segmentation

Color spaces:

- system that uses a specific color model to translate color into numbers.
- The Image Processing software typically represents colors as red, green, and blue (RGB) numeric values.

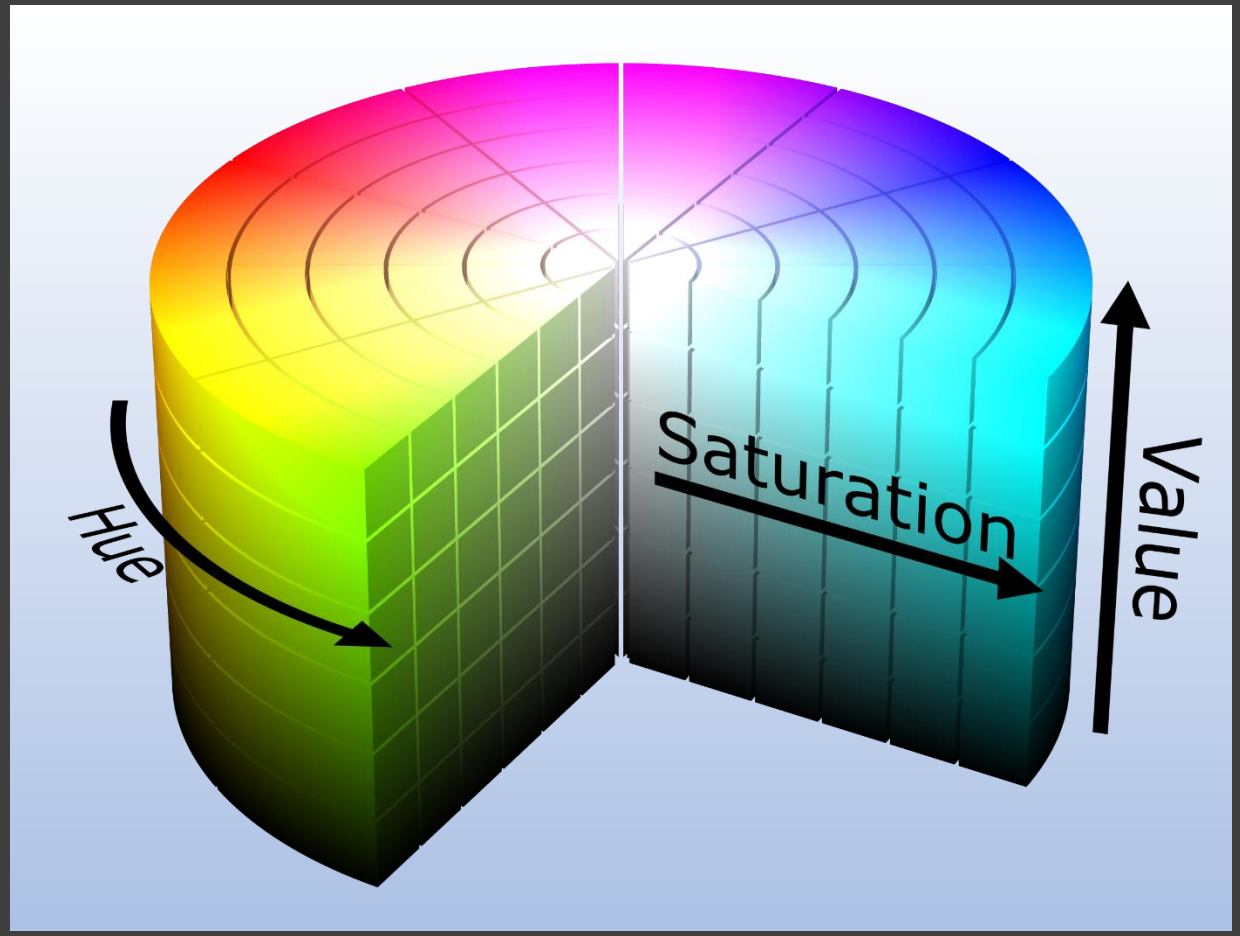


Examples of color spaces:

- RGB → intensity values of the red, green, blue color
- HSV → Hue, Saturation, Value
- CIE L*A*B → Luminance, Amount of red or green, Amount of yellow or blue

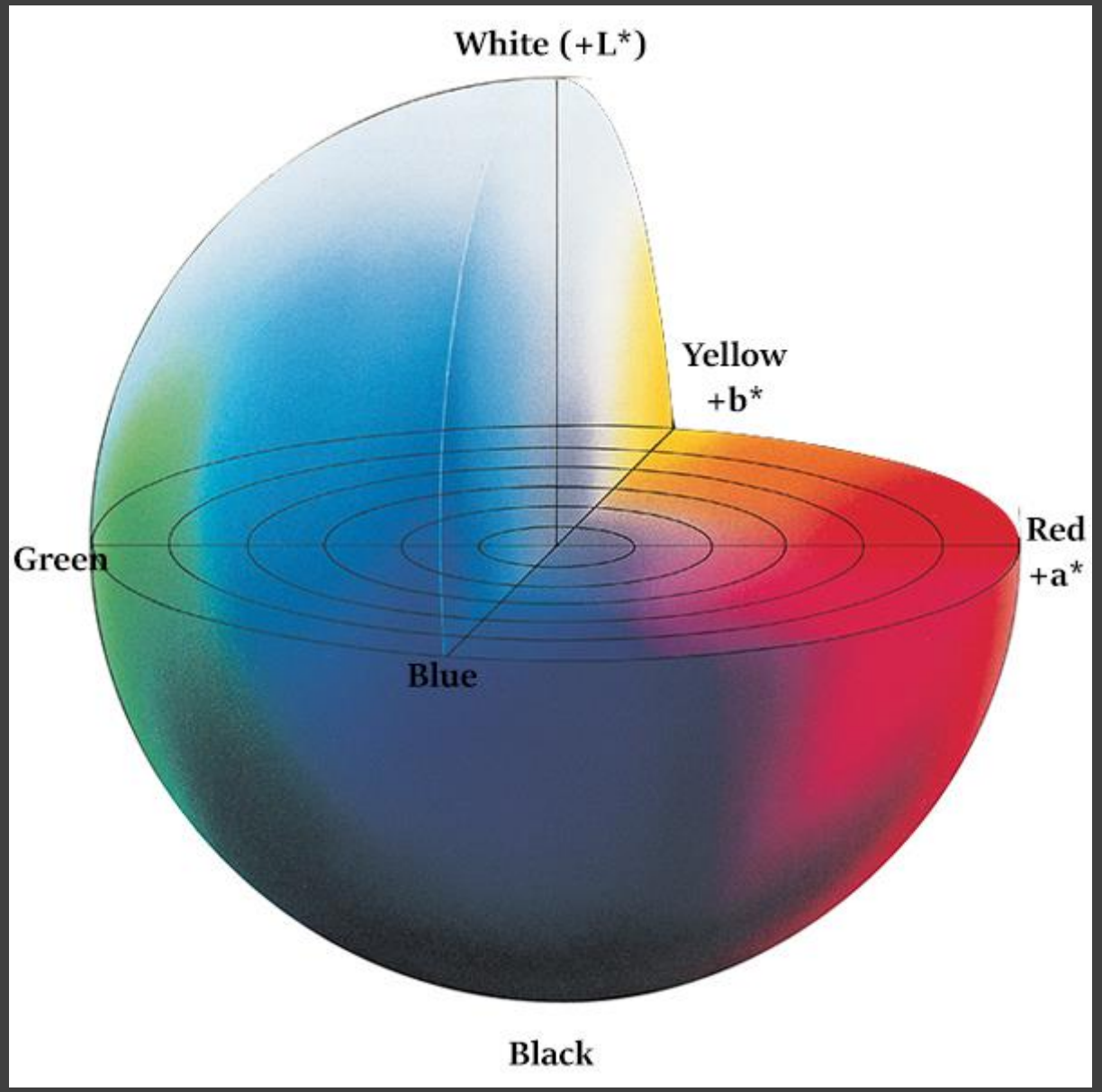
HSV:

Attribute	Description
H	Hue, which corresponds to the color's position on a color wheel. H is in the range [0, 1]. As H increases, colors transition from red to orange, yellow, green, cyan, blue, magenta, and finally back to red. Both 0 and 1 indicate red.
S	Saturation, which is the amount of hue or departure from neutral. S is in the range [0, 1]. As S increases, colors vary from unsaturated (shades of gray) to fully saturated (no white component).
V	Value, which is the maximum value among the red, green, and blue components of a specific color. V is in the range [0, 1]. As V increases, the corresponding colors become increasingly brighter.



CIE L*a*b* :

Attribute	Description
L*	Luminance or brightness of the image. Values are in the range [0, 100], where 0 specifies black and 100 specifies white. As L* increases, colors become brighter.
a*	Amount of red or green tones in the image. A large positive a* value corresponds to red. A large negative a* value corresponds to green. Values commonly fall in the range [-100, 100].
b*	Amount of yellow or blue tones in the image. A large positive b* value corresponds to yellow. A large negative b* value corresponds to blue. Values commonly fall in the range [-100, 100].



However, there are other models besides RGB for representing colors numerically.

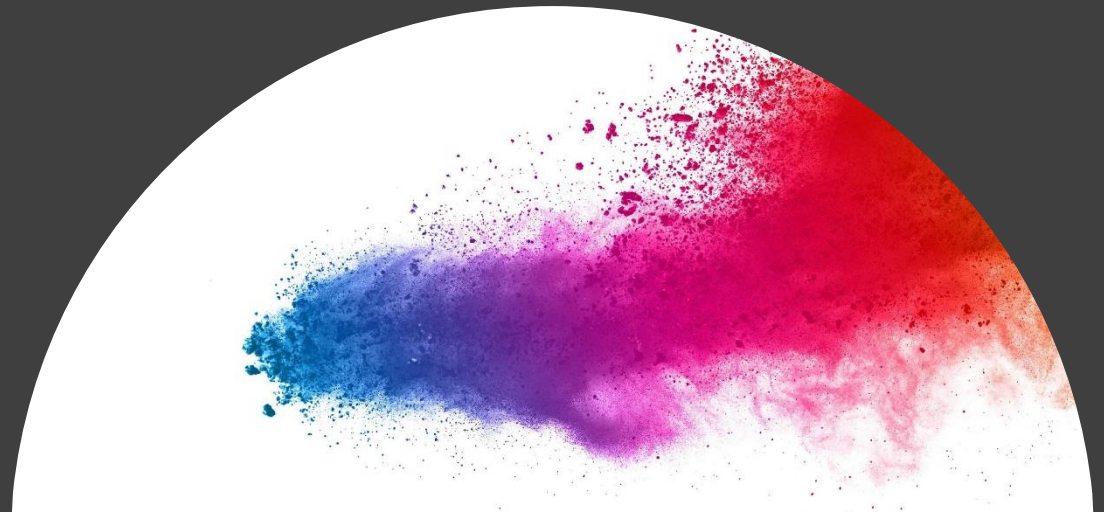
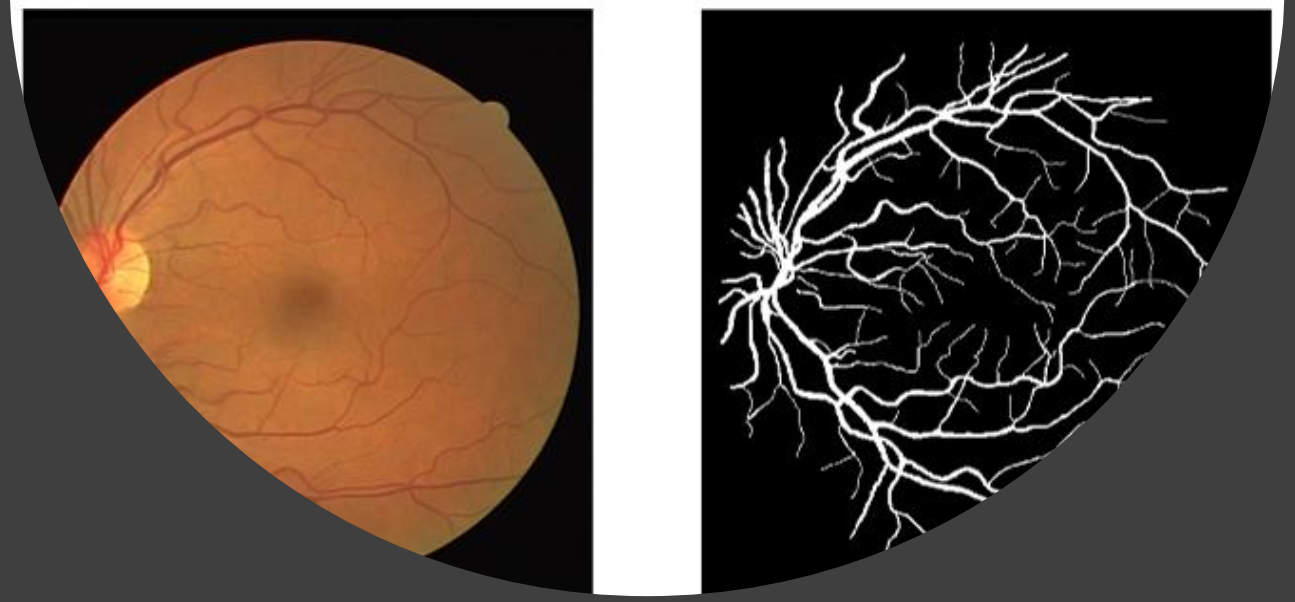
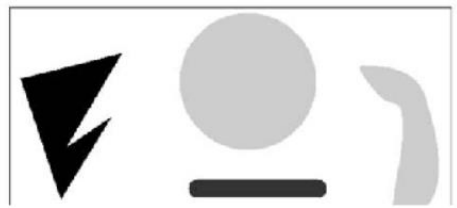


Image segmentation:



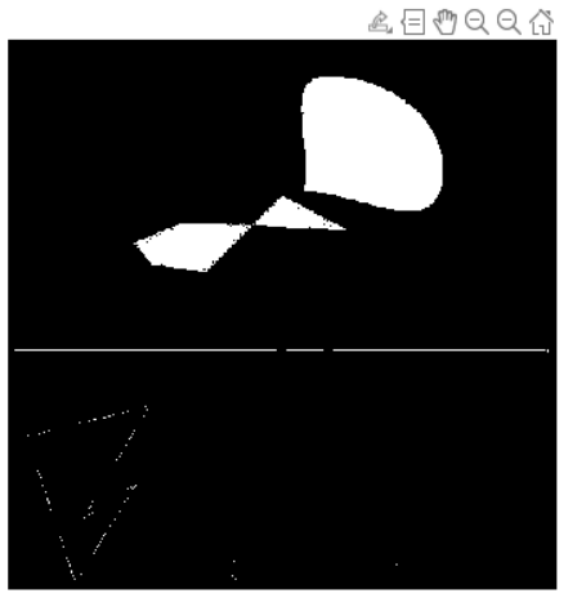
- Image segmentation is a commonly used technique in digital image processing and analysis to partition an image into multiple parts or regions

```
clc
clear
close all
warning off
image=imread('project.jpg');
gray_image=im2gray(image);
imshow(gray_image)
impixelinfo;
pause;
min=input("Enter the min_threshold value: ");
max=input("Enter the max_threshold value: ");
[row col]=size(gray_image);
mask=zeros(row,col);
for i=1:row
    for j=1:col
        if([gray_image(i,j)>=min gray_image(i,j)<=max])

            mask(i,j)=1;
        else
            mask(i,j)=0;
        end

    end
end
figure,imshow(mask)
for i=1:row
    for j=1:col
        if(mask(i,j)==0)
            gray_image(i,j)=255;

       end
    end
end
figure,imshow(gray_image)

```

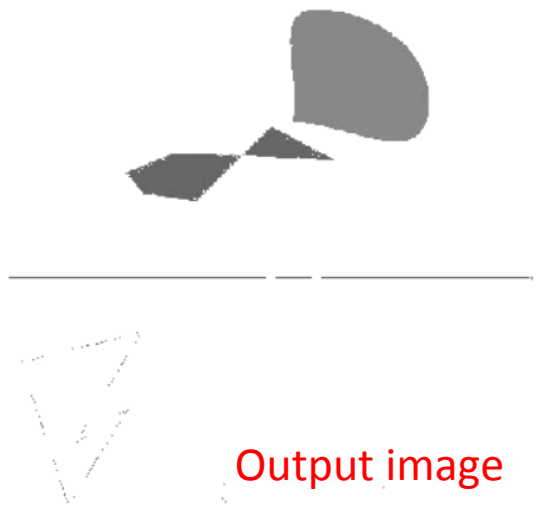


Pixel info: (306, 92) 136

Original image



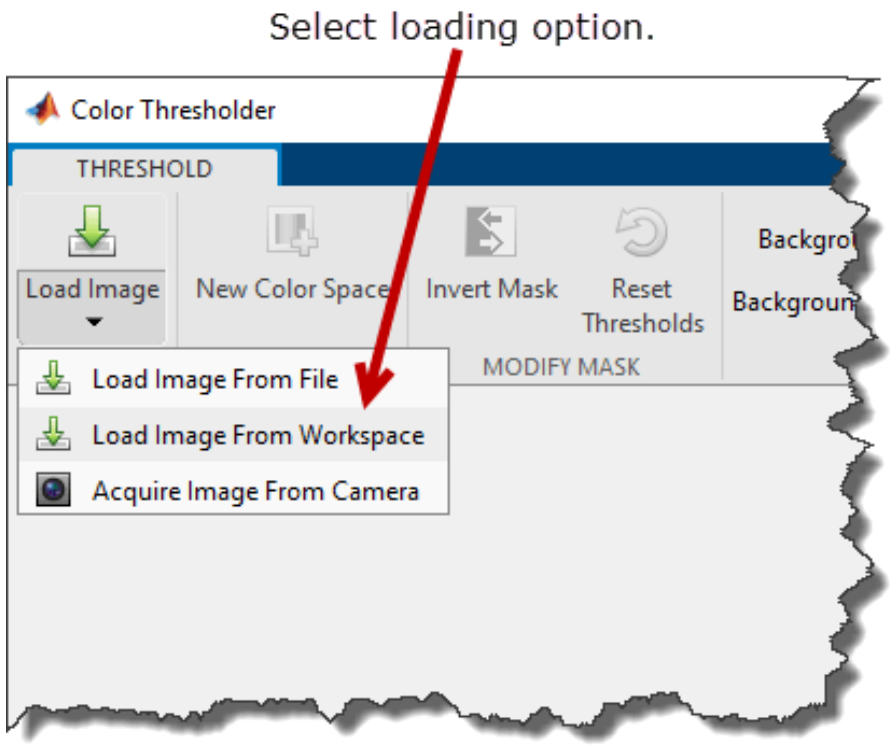
mask

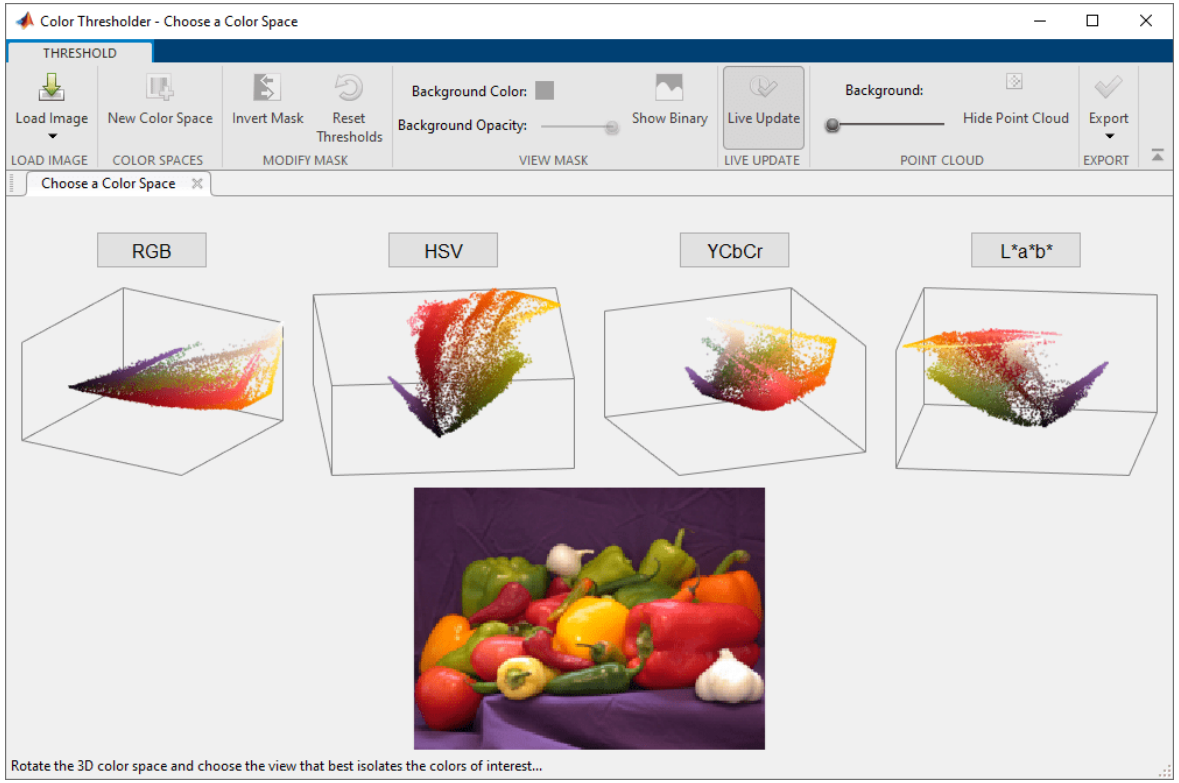


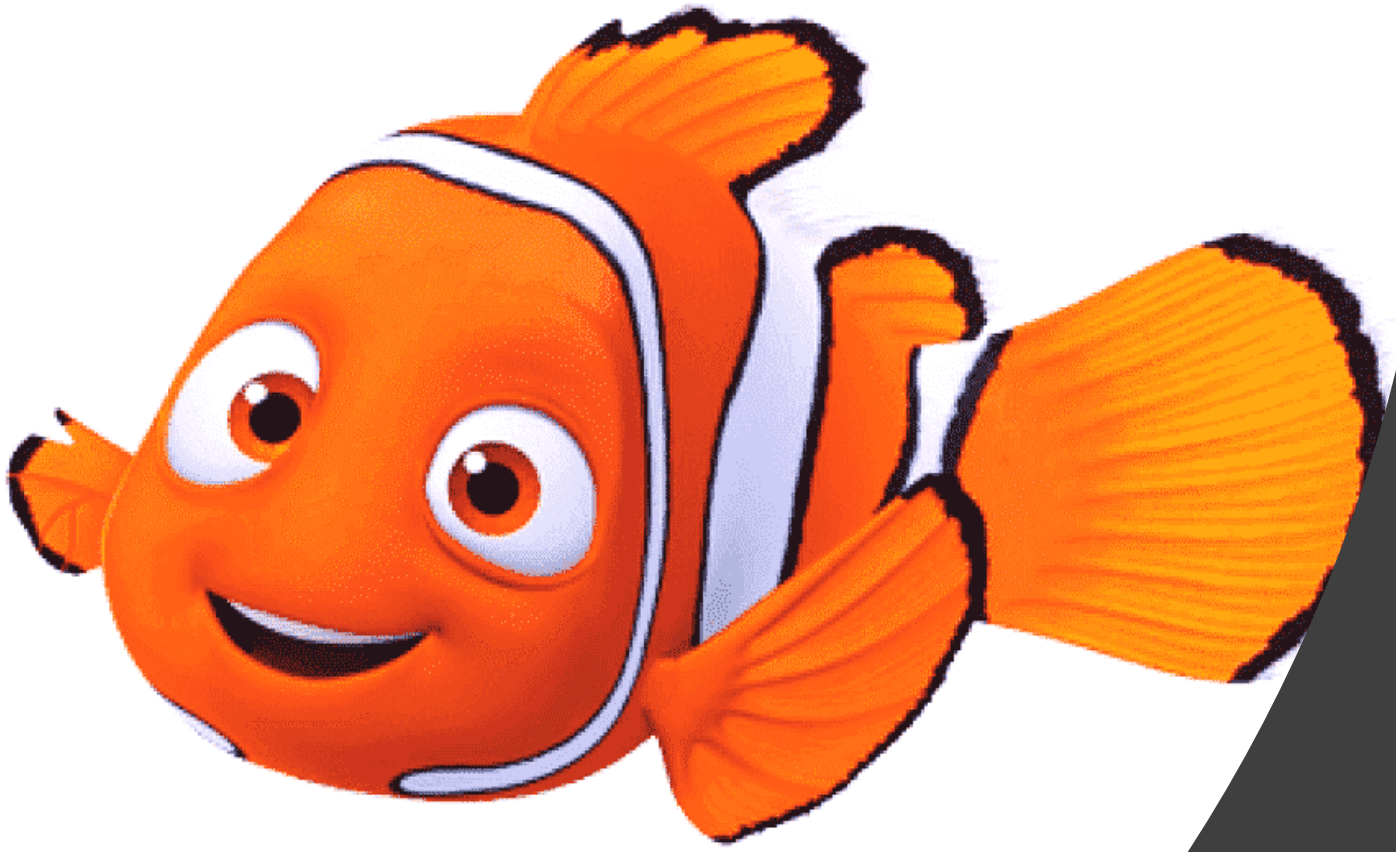
Output image

What about colored images?!

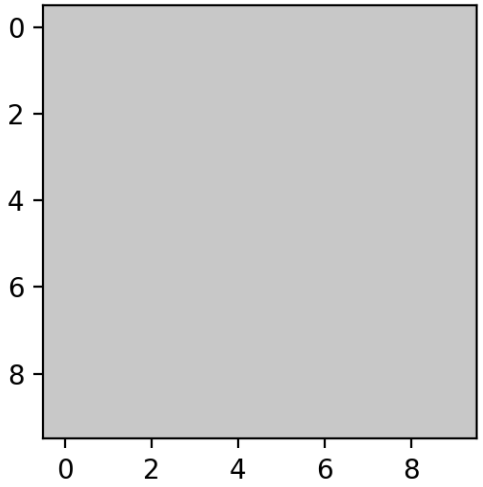
Color Thresholder

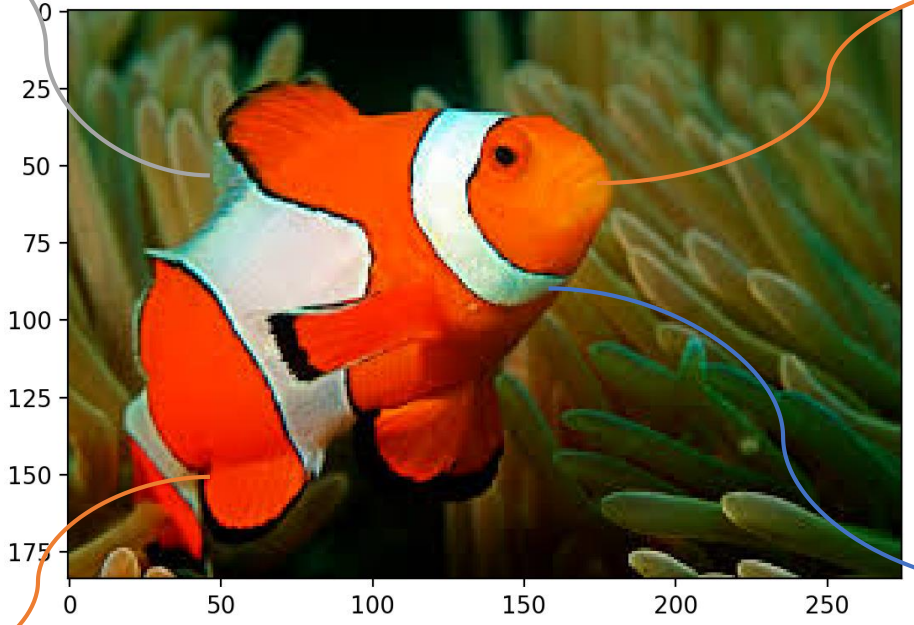


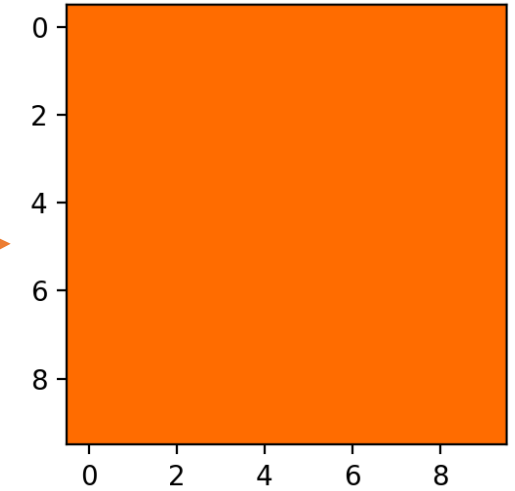


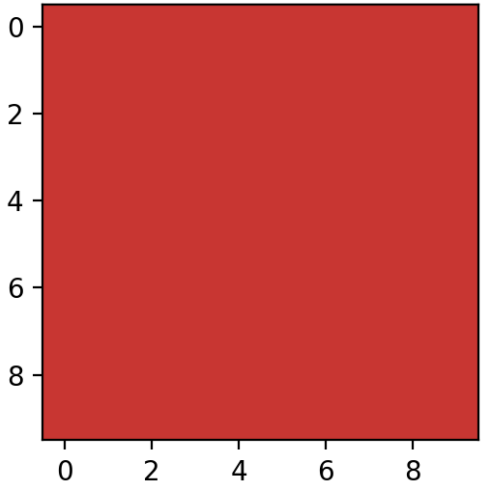


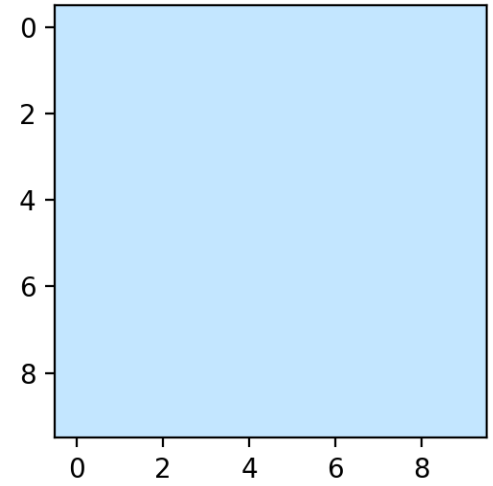
Let’s threshold Nemo


0
2
4
6
8
0
2
4
6
8

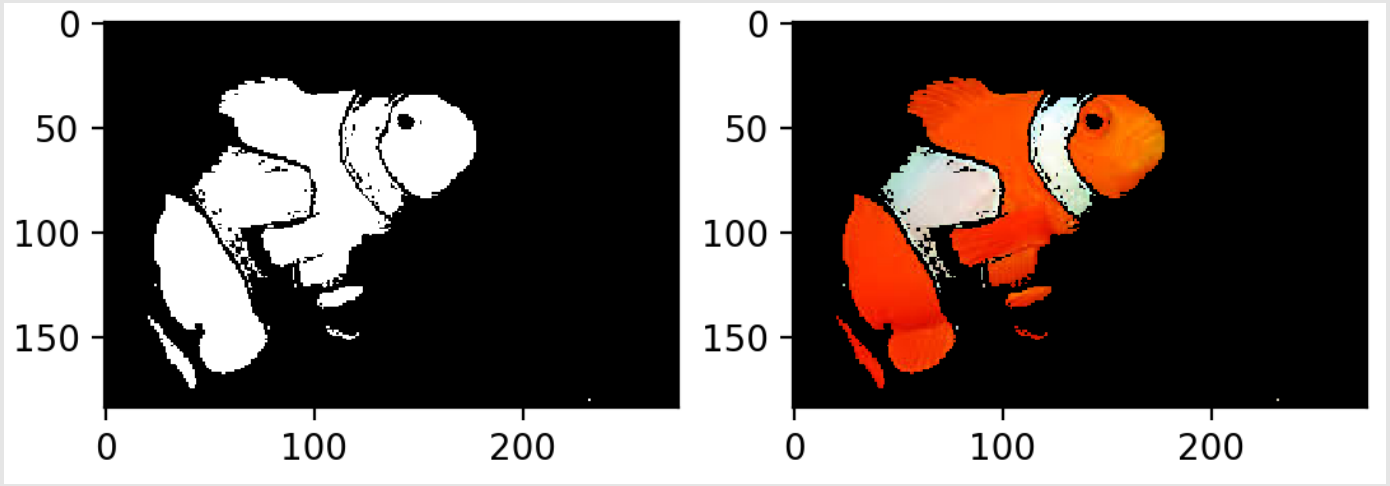

0
25
50
75
100
125
150
175
0
50
100
150
200
250


0
2
4
6
8
0
2
4
6
8

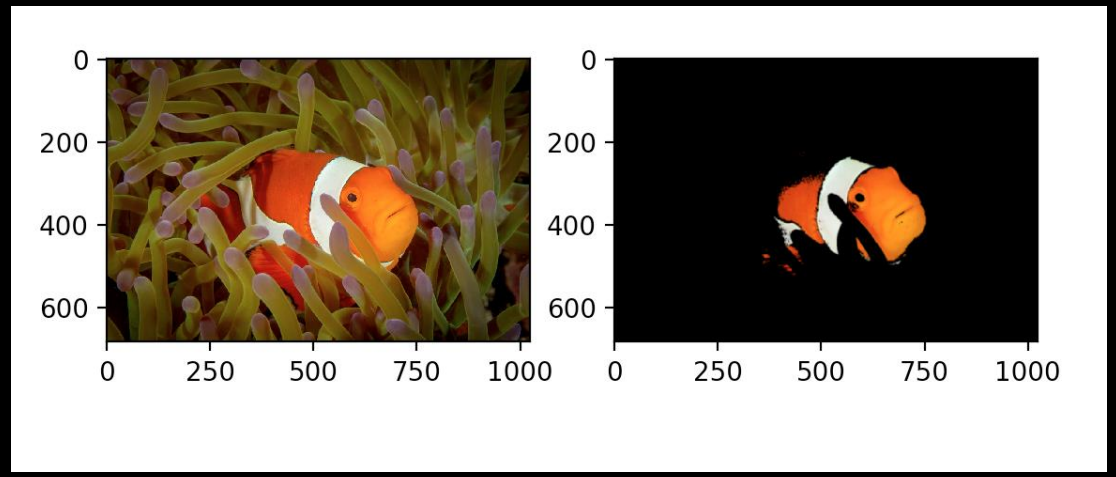

0
2
4
6
8
0
2
4
6
8

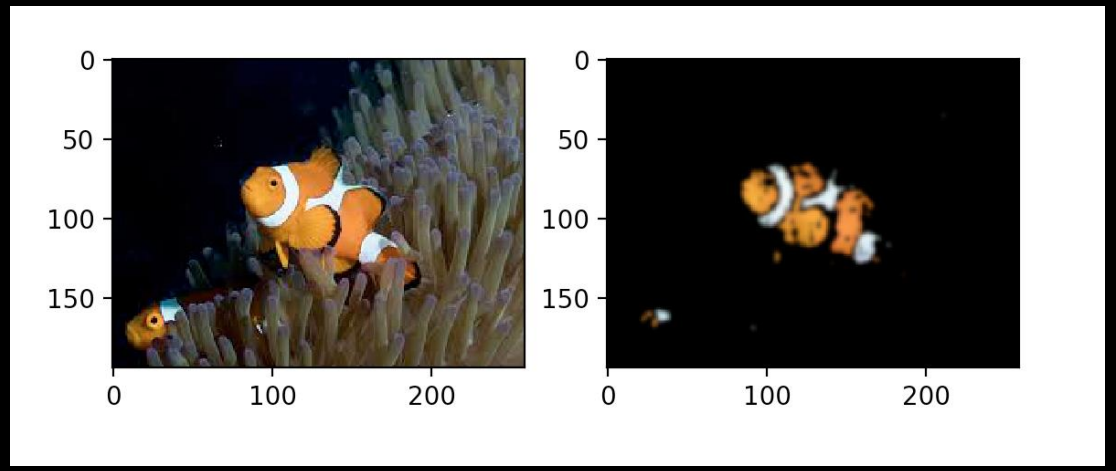

0
2
4
6
8
0
2
4
6
8

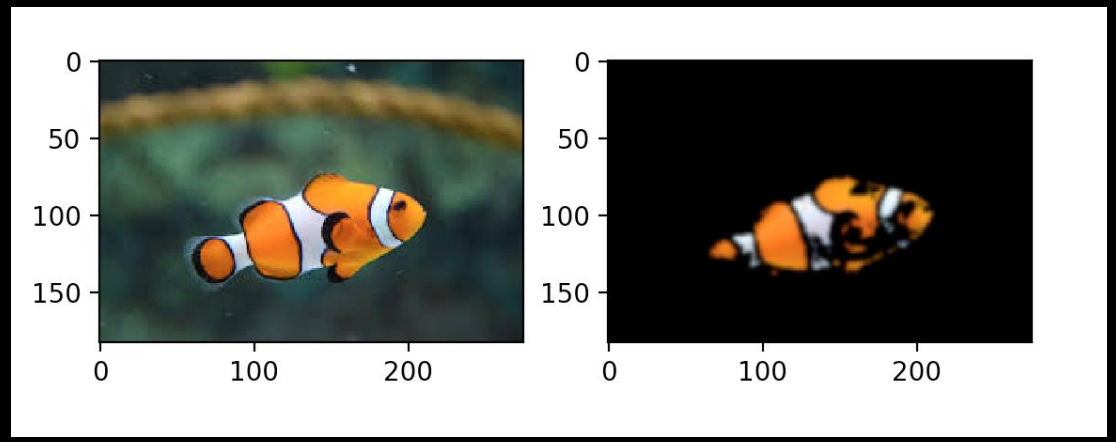
Now we have the mask

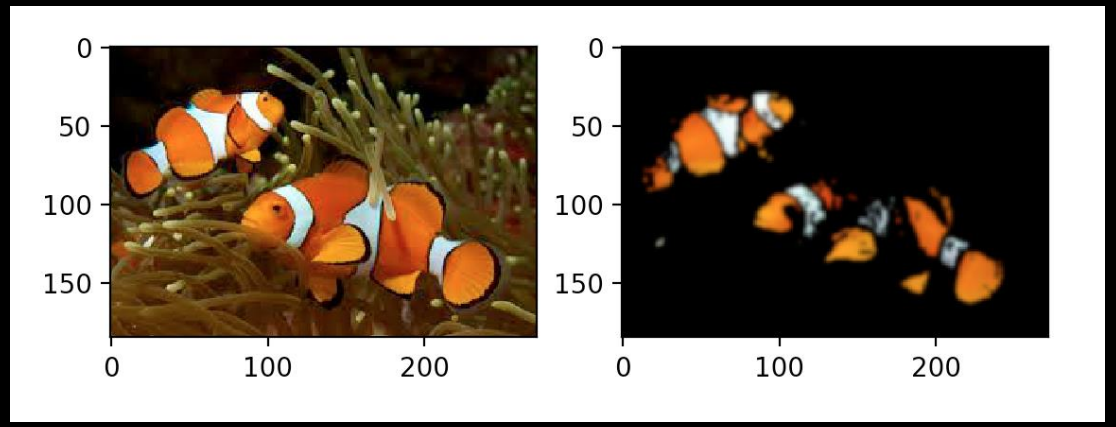


- We can use the same range to extract nemo from any image

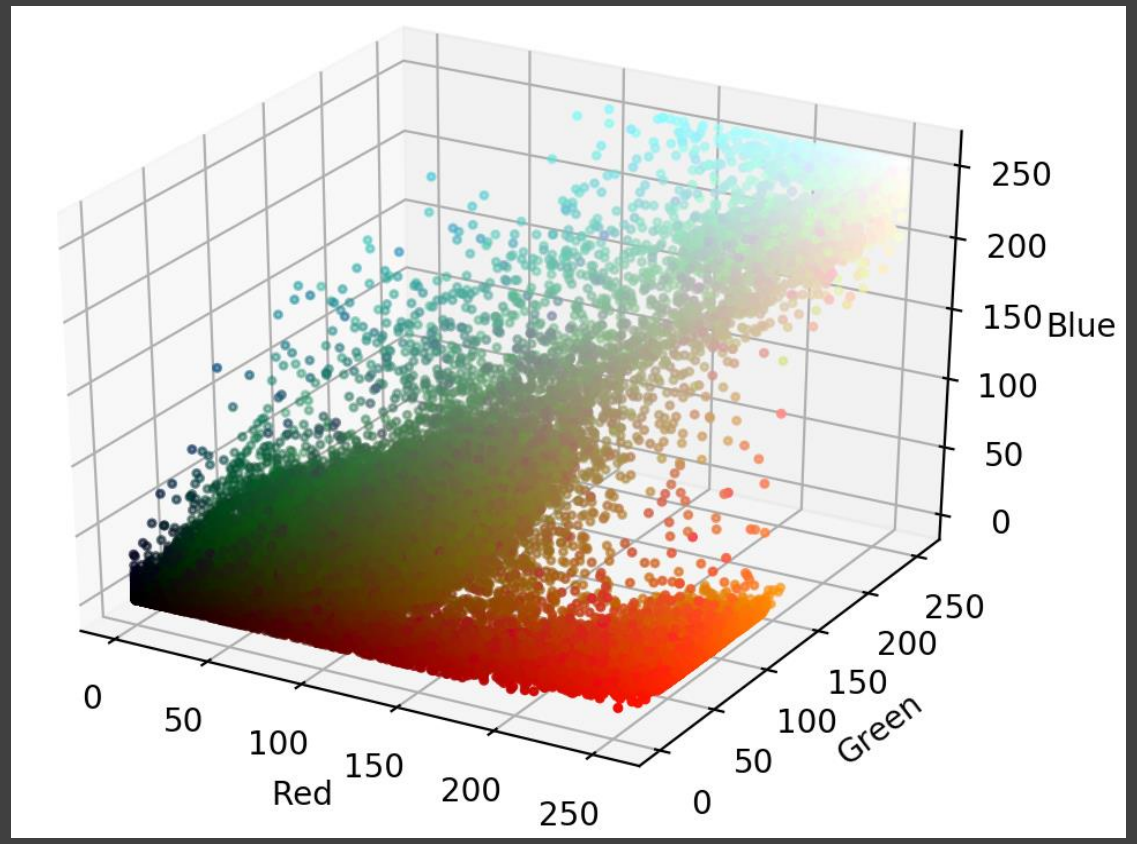

0
200
400
600
0
250
500
750
1000
0
200
400
600
0
250
500
750
1000


0
50
100
150
0
100
200
0
50
100
150
0
100
200

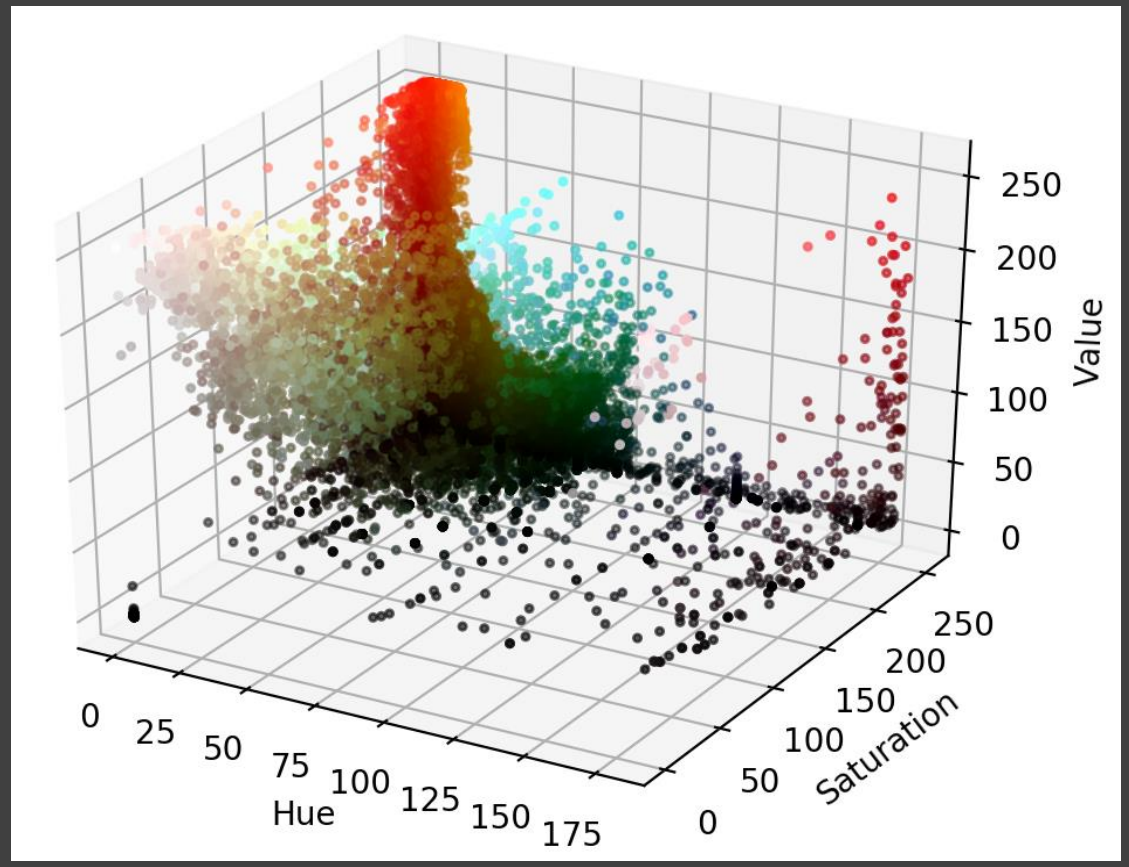

0
50
100
150
0
100
200
0
50
100
150
0
100
200


0
50
100
150
0
100
200
0
50
100
150
0
100
200

How to choose the best suitable color space?



the orange parts of the image span across almost the entire range of red, green, and blue values.



The saturation and value of the oranges do vary, but they are mostly located within a small range along the hue axis.