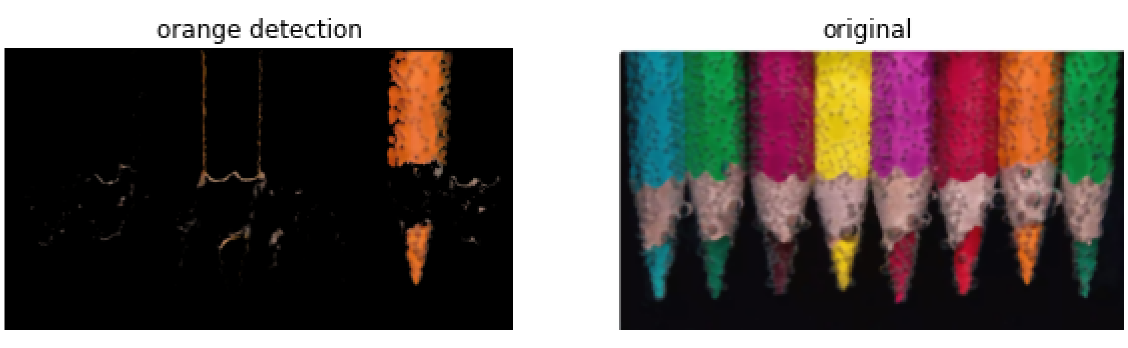
```
In [1]: import cv2
        import numpy as np
        import matplotlib.pyplot as plt
        from Image import my_image,Show,segmentation
        from skimage import color
```

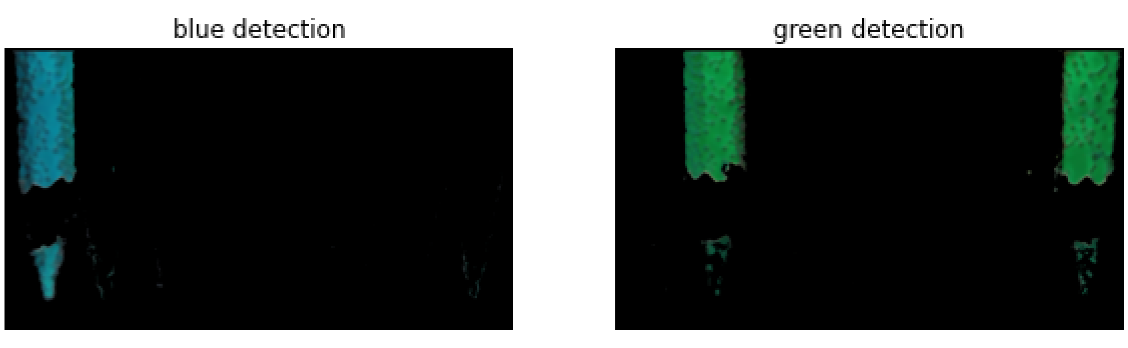
$\textbf{I. ColorSegmentator(image, min_color, max_color)}$

```
In [7]: pen = my_image.readimage('images/color2.jpg')
        pen = cv2.medianBlur(pen, ksize=11)

        result1_redpink=segmentation.ColorSegmentator(pen,120,179)
        result1_blue=segmentation.ColorSegmentator(pen,85,120)
        result1_yellow=segmentation.ColorSegmentator(pen,22,35)
        result1_orange=segmentation.ColorSegmentator(pen,8,17)
        result1_green=segmentation.ColorSegmentator(pen,36,80)

        Show.compareim(result1_orange,pen,'orange detection','original')
        Show.compareim(result1_yellow,result1_redpink,'yellow detection', 'red  & pink detection')
        Show.compareim(result1_blue,result1_green,'blue detection','green detection')
```

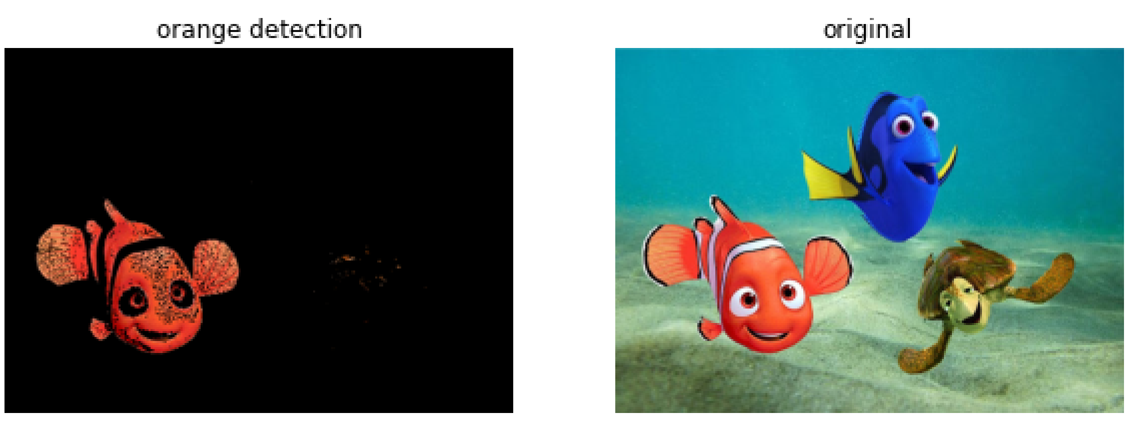


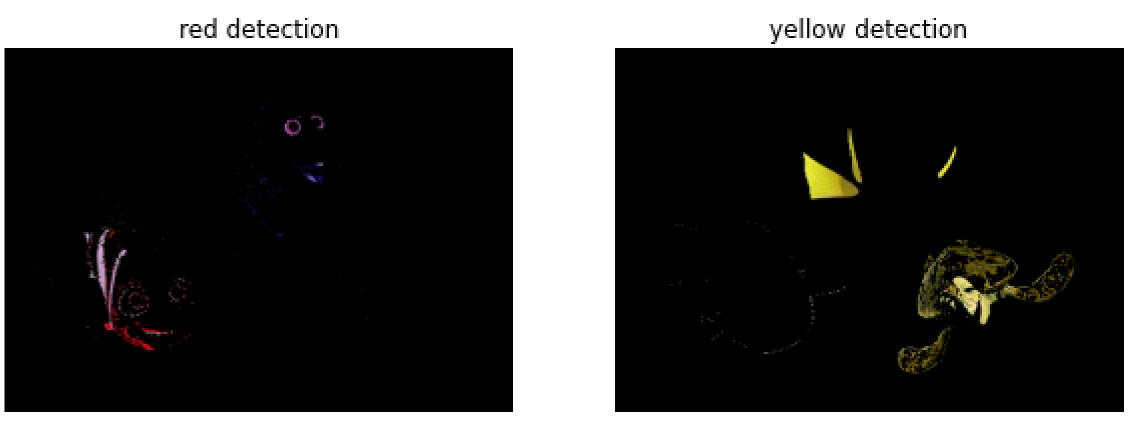


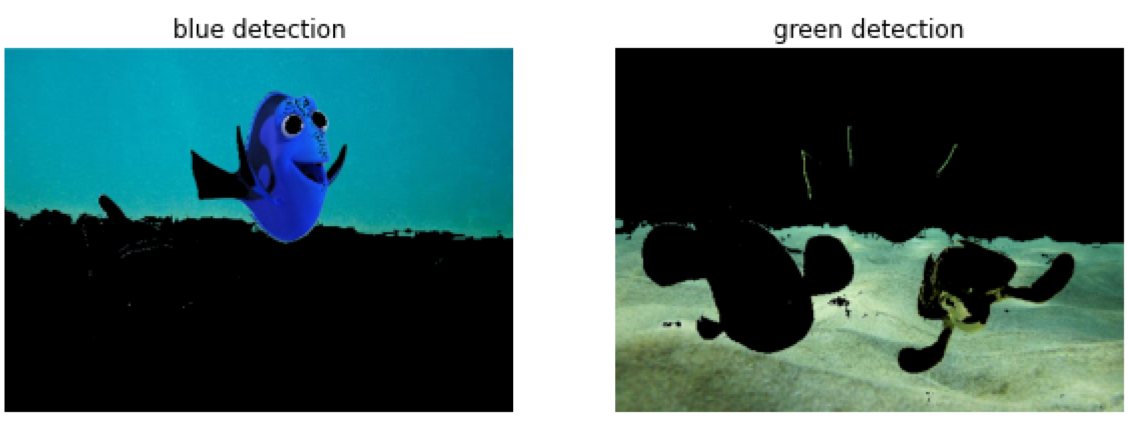

```
In [6]: nemo = my_image.readimage('images/color1.jpg')

        result1_redpink=segmentation.ColorSegmentator(nemo,119,179)
        result1_blue=segmentation.ColorSegmentator(nemo,85,120)
        result1_yellow=segmentation.ColorSegmentator(nemo,22,35)
        result1_orange=segmentation.ColorSegmentator(nemo,0,13)
        result1_green=segmentation.ColorSegmentator(nemo,36,80)

        Show.compareim(result1_orange,nemo,'orange detection','original')
        Show.compareim(result1_redpink,result1_yellow,'red detection','yellow detection')
        Show.compareim(result1_blue,result1_green,'blue detection','green detection')
```







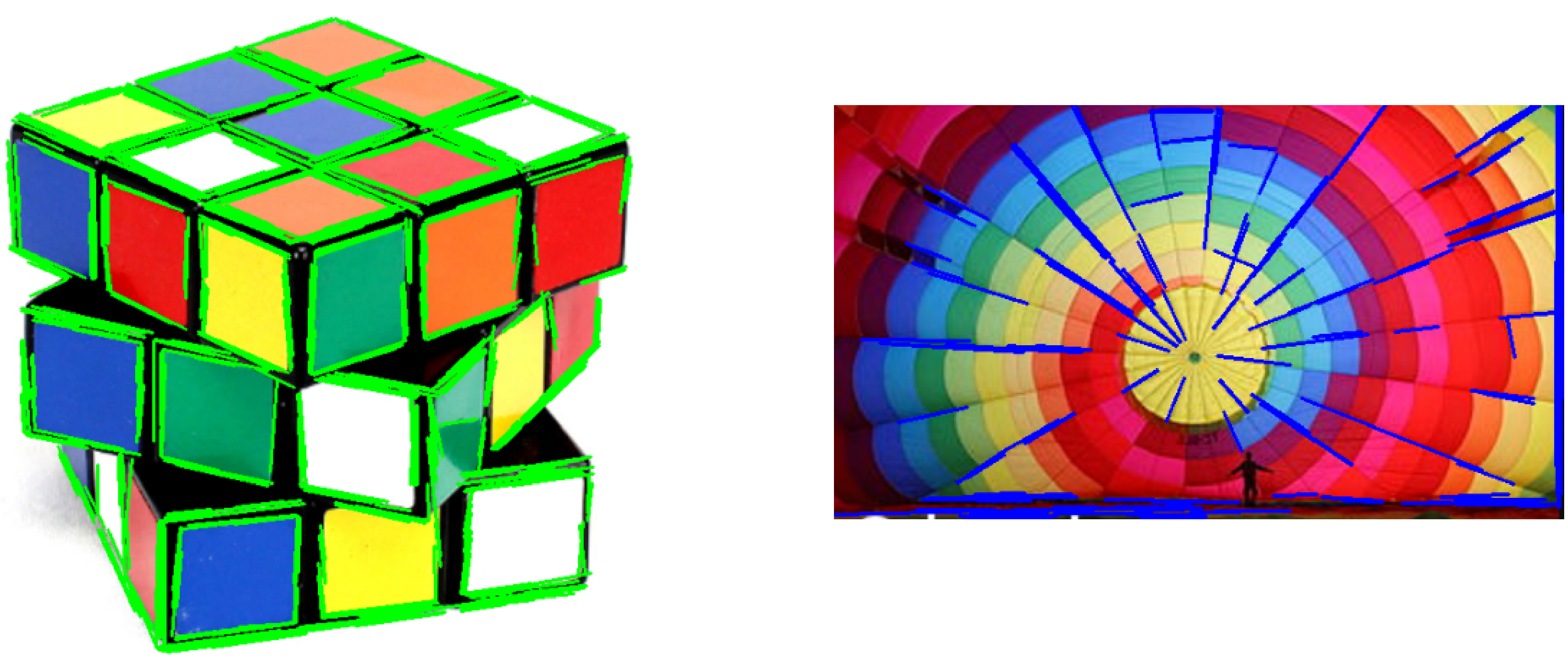
$\textbf{II. LinesDetector(image, minlenght)}$

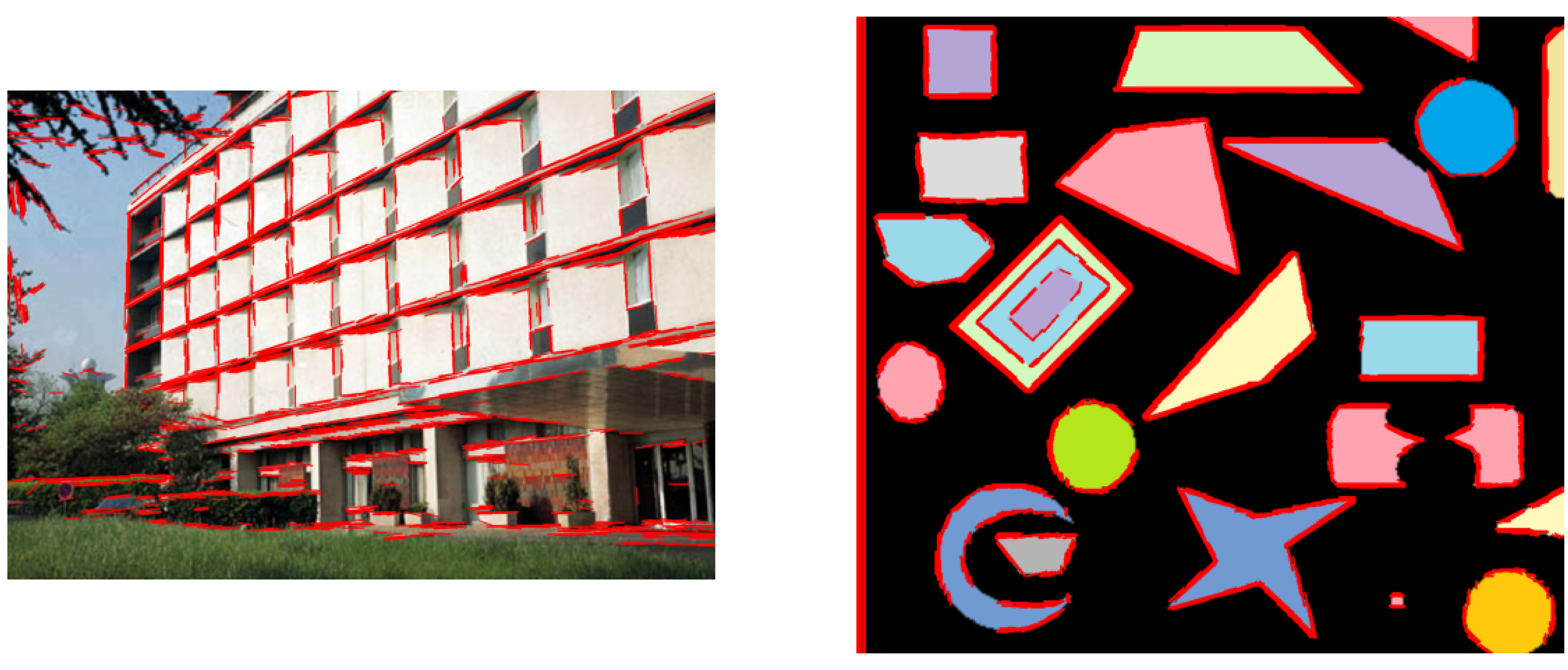
```
In [5]: img1=my_image.readimage('images/edgedetection/test1.png')
        img2=my_image.readimage('images/test5.png')
        img3=my_image.readimage('images/test8.png')
        img4=my_image.readimage('images/t3.png')



        line1=segmentation.LinesDetector(img1,minlenght=30,r=0,g=255,b=0)
        line2=segmentation.LinesDetector(img2,minlenght=30,r=0,g=0,b=255)
        line3=segmentation.LinesDetector(img3,minlenght=30)
        line4=segmentation.LinesDetector(img4,minlenght=10)

        Show.compareim(line1,line2,'','',2)
        Show.compareim(line3,line4,'','',2)
```



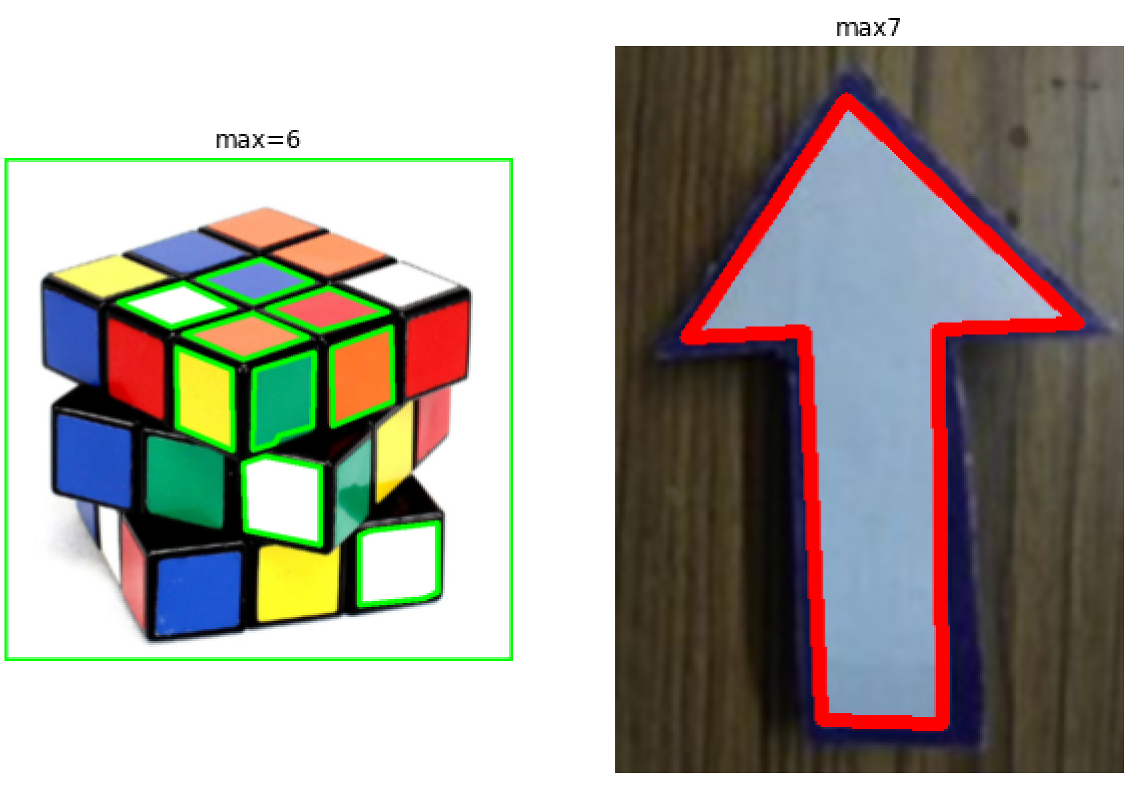


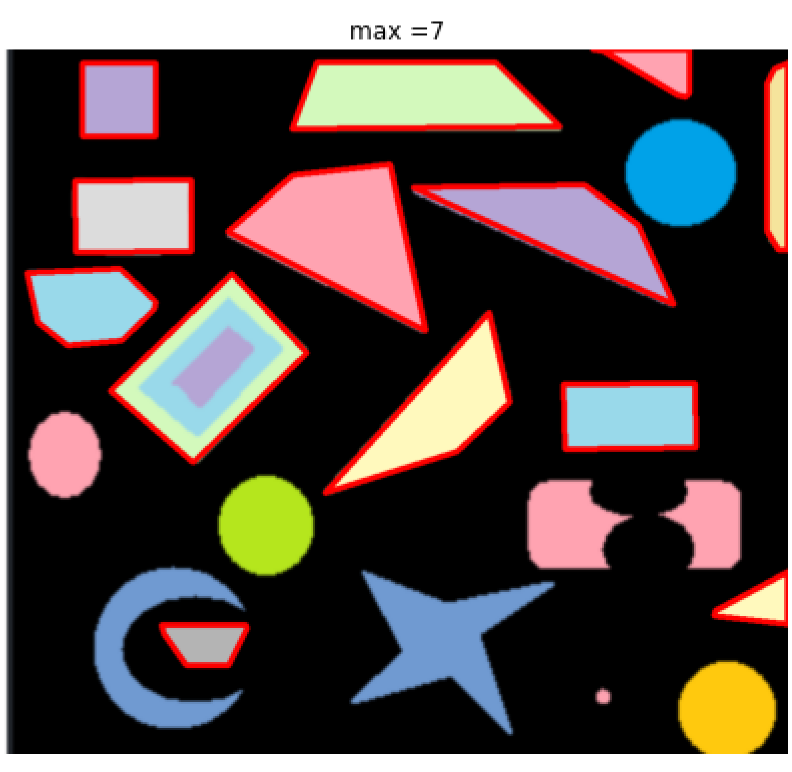
$\textbf{III. PolygonDetector(image, maxside)}$

```
In [4]: img1=cv2.imread('images/edgedetection/test1.png')
        img2=cv2.imread('images/t4.jpg')
        img4=cv2.imread('images/t3.png')

        s1=segmentation.PolygonDetector(img1,6,r=0,g=255,b=0)
        s2=segmentation.PolygonDetector(img2,7)
        s4=segmentation.PolygonDetector(img4,7)

        s1 = cv2.cvtColor(s1, cv2.COLOR_BGR2RGB)
        s2 = cv2.cvtColor(s2, cv2.COLOR_BGR2RGB)
        s4 = cv2.cvtColor(s4, cv2.COLOR_BGR2RGB)
        Show.compareim(s1,s2,'max=6','max7',1)
        Show.show_me(s4,'max =7')
```





$\star \star$

$\textbf{color segmentation with WEBCAM .}$

$ \text{exit with Esc}$

```
In [2]: segmentation.camera_color()
```

```
In [3]: segmentation.camera_Line(minlenght=30)
```

```
In [2]: segmentation.camera_Polygon(maxside=30)
```

```
In [ ]:
```

```
In [ ]:
```

```
In [ ]:
```

In [ ]: