Displaying specific region of image:

- Obtain the size of the image

  ```
  Height, weight, colors = image_name.shape()
  ```

- Copy the specific region of the image in a new image. For example, to extract the top half of the image,

  ```
  New_image = image[0:(height/2), :, :]
  ```

Converting between color formats:

- Opencv supports many color formats like BGR, HSV, GRAY etc. To convert an image from one color format to another, use the function

  ```
  Cv2.cvtcolor(image_name, COLOR_FORMAT_12COLOR_FORMAT_2)
  ```

- Ex: To convert from BGR to GRAY color format, use the following line,

  ```
  Cv2.cvtcolor(image_name, BGR2GRAY)
  ```

Masking image:

- Sometimes, it becomes important to extract only that region of the image which lies within a given range of pixel intensities. Opencv provides a function inRange() that can be used to do the same.

  ```
  mask  = cv2.inRange(image_name, lower_range, upper_range)
  ```

  This function extracts only that region of the image that lies within the range bounded by lower_range and upper_range. Both these objects are three element arrays that contain the boundary intensities of the three parameters constituting the color format.

Resizing images:

- Opencv provides functions for resizing images. The following lines of code can be used

  ```
  Final_image = cv2.resize(image_name,(width, height), interpolation = cv2.INTER_CUBIC)
  ```

Here, width is the desired width of the final_image and height is the desired height of the image. Interpolation refers to the technique that is employed to calculate the values of pixels that are newly added or to select the pixels that are to be removed.

For example, to reduce the width and height of an image, the following line of code is used,

```
Final_image = cv2.resize(image_name,(width/2, height/2), interpolation = cv2.INTER_CUBIC)
```

Here, width and height are the original width and height of the image that is found using shape() function described above.