**Draw geometric shapes with OpenCV**

Drawing a line:

cv2.line(image, start, end, color, thickness) function is used to draw a line on the image at desired location. It has five arguments:

1. image: It specifies the image on which we want to draw the line.
2. start: It specifies the starting position of line.
3. end: It specifies the ending position of the line.
4. color: It is used to give the color to line. BGR values are passed as a tuple to specify the desired color.
5. thickness: It decides the thickness of the line in px.

It returns an image with the line drawn on it.

**Python code:**

#import openCV library

import cv2

#imread() is a function in cv2 to read images

img = cv2.imread('test.jpg',1);

#draw a line

img = cv2.line(img,(0,0),(100,100),(0,255,0),20);

#imshow() is a function in cv2 to display images

cv2.imshow('image',img);

#waitKey to wait for 10 sec

cv2.waitKey(10000);

#close all windows

cv2.destroyAllWindows();

Output:



Drawing an arrowed line:

cv2.arrowedLine(image, start, end, color, thickness) function is used to draw an arrowed line on the image (arrow points toward the end point).

It also returns an image with an arrowed line drawn on it.

**Python code:**

#import openCV library

import cv2

#imread() is a function in cv2 to read images

img = cv2.imread('test.jpg',1);

#draw an arrowed line

img = cv2.arrowedLine(img,(100,0),(200,100),(255,200,0),5);

#imshow() is a function in cv2 to display images

cv2.imshow('image',img);

#waitKey to wait for 10 sec

cv2.waitKey(10000);

#close all windows

cv2.destroyAllWindows();

Output:



Drawing a rectangle:

cv2.rectangle( image, left\_top, right\_bottom, color, thickness) function is used to draw a rectangle on an image. It has five arguments:

1. image: It specifies the image on which we want to draw the rectangle.
2. left\_top: It specifies the starting of rectangle by a tuple( x-coordinate, y-coordinate) which is the coordinate point of left top corner of the rectangle.
3. right\_bottom: It specifies the ending of rectangle by a tuple( x-coordinate, y-coordinate) which is the coordinate point of right bottom corner of the rectangle.
4. color: It is used to give the color to line. BGR values are passed as a tuple to specify the desired color.
5. thickness: It decides the thickness of the rectangle in px.

It returns an image with the rectangle drawn on it.

**Python code:**

#import openCV library

import cv2

#imread() is a function in cv2 to read images

img = cv2.imread('test.jpg',1);

#draw a rectangle

img = cv2.rectangle(img,(0,0),(275,183),(255,200,100),7);

#imshow() is a function in cv2 to display images

cv2.imshow('image',img);

#waitKey to wait for 10 sec

cv2.waitKey(10000);

#close all windows

cv2.destroyAllWindows();

Output:



Drawing a circle:

cv2.circle(image, center, radius, color, thickness) function is used to draw a circle at desired location on the image. It has five arguments:

1. image: It specifies the image on which we want to draw the rectangle.
2. center: It specifies the center coordinates of the circle using a tuple (x,y).
3. radius: It specifies the radius of the circle in px.
4. color: It is used to give the color to line. BGR values are passed as a tuple to specify the desired color.
5. thickness: It decides the thickness of the circle in px.

It returns an image with the circle drawn on it.

**Python code:**

#import openCV library

import cv2

#imread() is a function in cv2 to read images

img = cv2.imread('test.jpg',1);

#draw a circle

img = cv2.circle(img,(137,91),50,(255,0,100),7);

#imshow() is a function in cv2 to display images

cv2.imshow('image',img);

#waitKey to wait for 10 sec

cv2.waitKey(10000);

#close all windows

cv2.destroyAllWindows();

Output:

