**目标：**

* 用OpenCV绘制图形
* 掌握函数：cv.line(), cv.circle() , cv.rectangle(), cv.ellipse(), cv.putText()

**代码**

上述函数的参数： - img：用于设置图像 - color：用于设置颜色。是一个RGR值，将用元组传递，例如：（255,0,0）为蓝色。对于灰度图像，只需传递标量值。 - thickness：用于设置线条或圆形等的厚度。如果是- 1则在图案内生成闭合图案并填充颜色。这个参数的默认厚度为1。 - lineType：用于设置线条的类型，有8型连接，抗锯齿等。默认情况是8型连接。cv2.LINE\_AA为抗锯齿，这样看起来会非常平滑。

**绘制直线**

要绘制线条，你需要设置起点和终点坐标。我们将创建一个黑色图像，并在其上从左上角到右下角绘制一条蓝线。 参考代码：

**import** numpy **as** np

**import** cv2 **as** cv

*# Create a black image*

img = np.zeros((512,512,3), np.uint8)

*# Draw a diagonal blue line with thickness of 5 px*

cv.line(img,(0,0),(511,511),(255,0,0),5)

**绘制矩形**

要绘制矩形，你需要设置矩形的左上角和右下角的坐标。这次我们将在图像的右上角绘制一个绿色矩形。 参考一下代码：

cv.rectangle(img,(384,0),(510,128),(0,255,0),3)

**绘制圆形**

要绘制圆形，你需要设置其圆点坐标和半径，我们将在上面绘制的矩形内绘制一个圆。 参考一下代码：

cv.circle(img,(447,63), 63, (0,0,255), -1)

**绘制椭圆**

要绘制椭圆，我们需要传递几个参数，一个参数是椭圆中心位置（x，y），下一个是长轴长度和断轴长度（a，b），角度是椭圆在逆时针方向上的旋转角度，startAngle和endAngle表示从主轴顺时针方向测量的椭圆弧的起点和终点，即给出值0和360给出完整的椭圆，给出值180则画出半个椭圆。 参考一下代码：

cv.ellipse(img,(256,256),(100,50),0,0,180,255,-1)

**绘制多边形**

要绘制多边形，首先需要顶点坐标。 将这些点转换为ROWSx1x2的数组，其中ROWS是顶点数，它应该是int32类型。在这里，我们绘制一个带有四个黄色顶点的小多边形。 参考一下代码：

pts = np.array([[10,5],[20,30],[70,20],[50,10]], np.int32)

pts = pts.reshape((-1,1,2))

cv.polylines(img,[pts],**True**,(0,255,255))

注意：如果第三个参数为False，则绘制所有点的相连图形而不是闭合图形。cv.polylines（）可用于绘制多条线。 只需创建要绘制的所有行的列表并将其传递给函数。 所有线条都将单独绘制。 绘制一组直线比每行调用cv.line（）要好得多，速度更快。

**向图像中添加文字**

要将文本放入图像中，你需要传递以下几个参数： - 要写入的文本数据， - 要放置的位置（即文本数据的左下角） - 字体类型（检查cv.putText（）文档以获取支持的字体）， - 字体大小 - 常规的参数，比如颜色、粗细、线型等，为了更好看，建议使用lineType = cv.LINE\_AA作为线型参数的值。

我们将在图像上编写白色的OpenCV字样。 参考以下代码：

font = cv.FONT\_HERSHEY\_SIMPLEX

cv.putText(img,'OpenCV',(10,500), font, 4,(255,255,255),2,cv.LINE\_AA)

**结果**

是时候看看我们所绘制图案的结果了，正如文章之间所讲述的那样，通过显示图像将上面六个绘制结果显示出来。 窗口将如下图所示： 