

Drawing images and shapes using OpenCv

Let's start off by making a black square

to draw an image we need to create a black or white image

by

`np.zeros((dimensions), np.uint8) ----># we can create black box`

`n.ones((dimensions), np.uint8) -----># we can create white image`

```
In [5]: 1 import cv2
        2 import numpy as np
        3 #Create a black image
        4 # 3 means three layers for RGB color
        5 image = np.zeros((512, 512, 3), np.uint8)
        6
        7 #Can we make this in black and white?
        8 image_bw = np.zeros((512, 512), np.uint8)
        9
       10 cv2.imshow('Black Rectangel(color)', image)
       11 cv2.imshow('Black Retangle (B & W)', image_bw)
       12 cv2.waitKey()
       13 cv2.destroyAllWindows()
```

Line

`cv2.line(image, starting cordinates, ending cordinates, color, thickness)`

Draw a line over our black square

```
In [1]: 1 import numpy as np
        2 import cv2
        3 image = np.zeros((512, 512, 3), np.uint8)
        4 cv2.line(image, (0, 0), (512, 512), (255, 127, 0), 5)
        5 cv2.imshow('Blue Line', image)
        6 cv2.waitKey()
        7 cv2.destroyAllWindows()
```

Rectangle

rectangle(image, starting vertex, opposite vertex, color, thickness)

```
In [3]: 1 image = np.ones((512, 512, 3), np.uint8)
        2 cv2.rectangle(image, (100, 100), (300, 250), (127, 50, 127), 3)
        3 cv2.imshow('Rectangle', image)
        4 cv2.waitKey()
        5 cv2.destroyAllWindows()
```

^if you want to fill the rectangle with passed color the pass the thickness = -1

Circle

cv2.circle(image, center, radius, color, thickness)

```
In [9]: 1 image = np.ones((512, 512, 3), np.uint8)
        2 cv2.circle(image, (256, 256), (50), (255, 2, 255), -3)
        3 cv2.imshow('Circle', image)
        4 cv2.waitKey()
        5 cv2.destroyAllWindows()
```

polylines

by this we can create triange, polygon, hexagon, ocatve etc...

polylines(img, pts, isClosed, color, thickness)

Polygons

```
In [2]: 1 import numpy as np, cv2
2 image = np.zeros((512, 512, 3), np.uint8)
3 #Let's define for point
4 pts = np.array([[10, 50], [400, 50], [90, 200], [50, 500]], np.int32)
5 print(f'Before reshape {pts}')
6 #Let's nw reshape our pints in form requred by polylines
7 pts = pts.reshape((-1, 1, 2))
8 print(f'After reshape {pts}')
9 cv2.polylines(image, [pts], True, (0, 0, 255), 3)
10 cv2.imshow('Polygon', image)
11 cv2.waitKey()
12 cv2.destroyAllWindows()
```

Before reshape [[10 50]

[400 50]

[90 200]

[50 500]]

After reshape [[[10 50]]

[[400 50]]

[[90 200]]

[[50 500]]]

Trinangle

EXERCISE

```
In [8]: 1 image = np.zeros((512, 512, 3), np.uint8)
2 #Let's define for pints
3 pts = np.array([[1, 509], [254, 1], [509, 509]], np.int32)
4 #Let's nw reshape our pints in form required by polylines
5 pts = pts.reshape((-1, 1, 2))
6 cv2.polylines(image, [pts], True, (0, 0, 255), 3)
7 cv2.rectangle(image, (127, 255), (381, 509), (0, 255, 0), -3)
8 cv2.circle(image, (254, 382), (127), (255, 0, 0), -4)
9 cv2.putText(image, 'Noushad Khan', (134, 389), cv2.FONT_HERSHEY_COMPLEX, 1, (255, 255, 255), 3)
10 cv2.imshow('Project', image)
11 cv2.waitKey()
12 cv2.destroyAllWindows()
```

Text

cv2.putText(image, 'Text to display', bottom left starting point, Font Type, Font Size, Color, thickness) ¶

Font style

FONT_HERSHEY_SIMPLEX, FONT_HERSHEY_PLAIN FONT_HERSHEY_DUPLEX, FONT_HERSHEY_COMPLEX FONT_HERSHEY_TRIPLEX,
FONT_HERSHEY_COMPLEX_SMALL FONT_HERSHEY_SCRIPT_SIMPLEX FONT_HERSHEY_SCRIPT_COMPLEX

```
In [99]: 1 image = np.ones((512, 512, 3), np.uint8)
2 cv2.putText(image, 'Noushad Khan', (134, 389), cv2.FONT_HERSHEY_SCRIPT_SIMPLEX, 1, (255, 255, 255), 3)
3 cv2.imshow('Text', image)
4 cv2.waitKey()
5 cv2.destroyAllWindows()
```