Opencv Basics

1. Read Image

```
import cv2

img =cv2.imread("C:\\Users\\Asus-2020\\PycharmProjects\\OCV\\Resources\\curre.JPG")
cv2.imshow("Jasvee",img)
cv2.waitKey(2000)
```

2. Read Video file:

```
import cv2

#framewidth = 640
#frameheight = 360

cap = cv2.VideoCapture("C:\\Users\\Asus-
2020\\PycharmProjects\\OCV\\Resources\\Jas.mp4")

while True:
    sucess,img = cap.read()
    cv2.imshow("Jasvee Video",img)
    #picture
    #if cv2  .waitKey(0) & 0xFF == ord('q'):
    #video
    if cv2.waitKey(5) & 0xFF == ord('q'):
        break
```

3. Open webcam

```
import cv2

framewidth = 640
frameheight = 360

cap = cv2.VideoCapture(0)
#cap = cv2.VideoCapture("C:\\Users\\Asus-
2020\\PycharmProjects\\OCV\\Resources\\Jas.mp4")
cap.set(3,framewidth)
cap.set(4,frameheight)


while True:
    sucess,img = cap.read()
    #resize function
    img = cv2.resize(img,(framewidth,frameheight))
    cv2.imshow("Jasvee Webcam",img)
    #picture
    #if cv2  .waitKey(0) & 0xFF == ord('q'):
    #video
```

```
    if cv2.waitKey(5) & 0xFF == ord('q'):
        break
```

2. Basic functions:

RGB to Grey scale image:

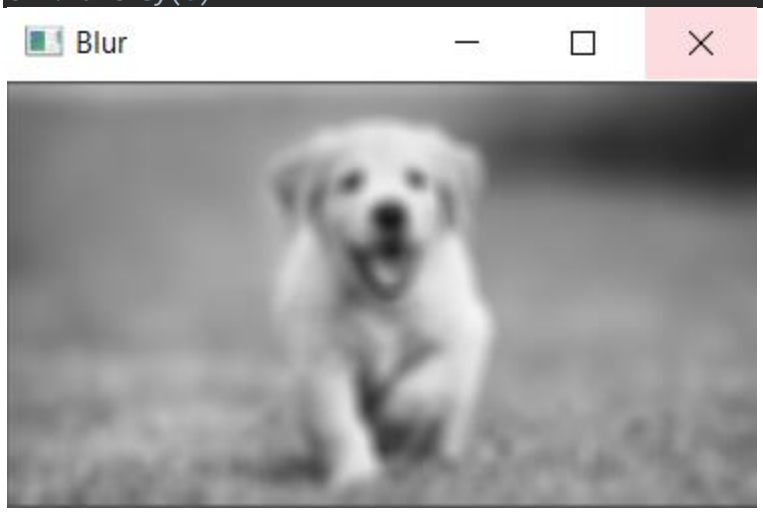
Function : cvtColor

```
import cv2

#RGB to greysccale image
path = "C:\\Users\\Asus-2020\\PycharmProjects\\OCV\\Resources\\dog.jfif"
#Rgb image
#img =cv2.imread(path)

#Grey scale image
#Method 1
img =cv2.imread(path,0)
#Method 2
imggrey = cv2.cvtColor(img,cv2.COLOR_BAYER_BG2GRAY)
cv2.imshow('Jasvee Greyscale',img)
cv2.imshow('Jasvee Greyscale 1',imggrey)
cv2.waitKey(0)
```

Blur image

Function : blur

```
import cv2

#RGB to greysccale image
path = "C:\\Users\\Asus-2020\\PycharmProjects\\OCV\\Resources\\dog.jfif"
#Rgb image
#img =cv2.imread(path)

#Grey scale image
#Method 1
img =cv2.imread(path,0)
#Method 2
imggrey = cv2.cvtColor(img,cv2.COLOR_BAYER_BG2GRAY)

#Image Blur
#Method 1
imgblur = cv2.blur(imggrey,(7,7))
#Method2
#imgblur = cv2.GaussianBlur(imggrey,(7,7),0)

cv2.imshow('Jasvee Greyscale',img)
cv2.imshow('Jasvee Greyscale 1',imggrey)
cv2.imshow('Blur',imgblur)
cv2.waitKey(0)
```



Edge Detectors:

Function : Canny

```
import cv2

#RGB to greysccale image
path = "C:\\Users\\Asus-2020\\PycharmProjects\\OCV\\Resources\\dog.jfif"
#Rgb image
#img =cv2.imread(path)
```

```
#Grey scale image
#Method 1
img =cv2.imread(path,0)
#Method 2
imggrey = cv2.cvtColor(img,cv2.COLOR_BAYER_BG2GRAY)

#Image Blur
#Method 1
imgblur = cv2.blur(imggrey,(7,7))
#Method2
#imgblur = cv2.GaussianBlur(imggrey,(7,7),0)

#Edge Detectors
imgdet = cv2.Canny(imgblur,40,50)

cv2.imshow('Jasvee Greyscale',img)
cv2.imshow('Jasvee Greyscale 1',imggrey)
cv2.imshow('Blur',imgblur)
cv2.imshow('Edge',imgdet)
cv2.waitKey(0)
```

Dilation:

Function : dilate

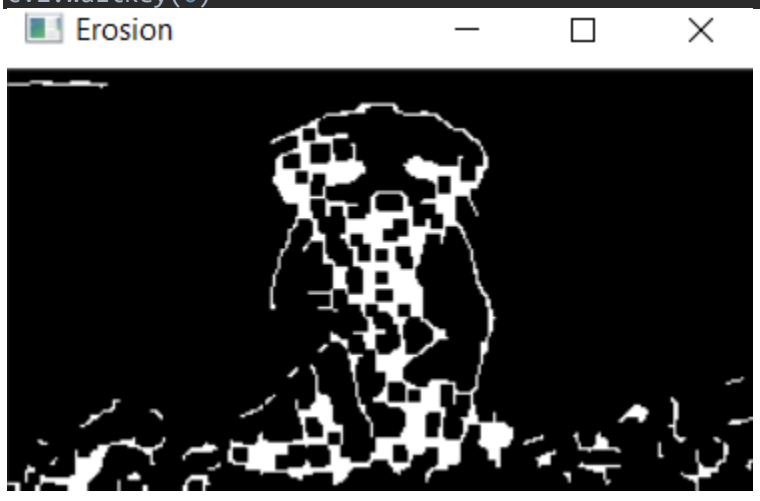
```
import cv2
import numpy as np

kernel = np.ones((5,5),np.int)
print(kernel)
#RGB to greysccale image
path = "C:\\Users\\Asus-2020\\PycharmProjects\\OCV\\Resources\\dog.jfif"
#Rgb image
#img =cv2.imread(path)
```

```
#Grey scale image
#Method 1
img =cv2.imread(path,0)
#Method 2
imggrey = cv2.cvtColor(img,cv2.COLOR_BAYER_BG2GRAY)

#Image Blur
#Method 1
imgblur = cv2.blur(imggrey,(7,7))
#Method2
#imgblur = cv2.GaussianBlur(imggrey,(7,7),0)

#Edge Detectors
#imgdet = cv2.Canny(imgblur,100,100)
imgdet = cv2.Canny(imgblur,40,50)

#image Dilation
imgdilation = cv2.dilate(imgdet,kernel,iterations=1)

cv2.imshow('Jasvee Greyscale',img)
cv2.imshow('Jasvee Greyscale 1',imggrey)
cv2.imshow('Blur',imgblur)
cv2.imshow('Edge',imgdet)
cv2.imshow('Dilation',imgdilation)
cv2.waitKey(0)
```

Erosion:

```
import cv2
import numpy as np

kernel = np.ones((5,5),np.int)
print(kernel)
#RGB to greysccale image
```

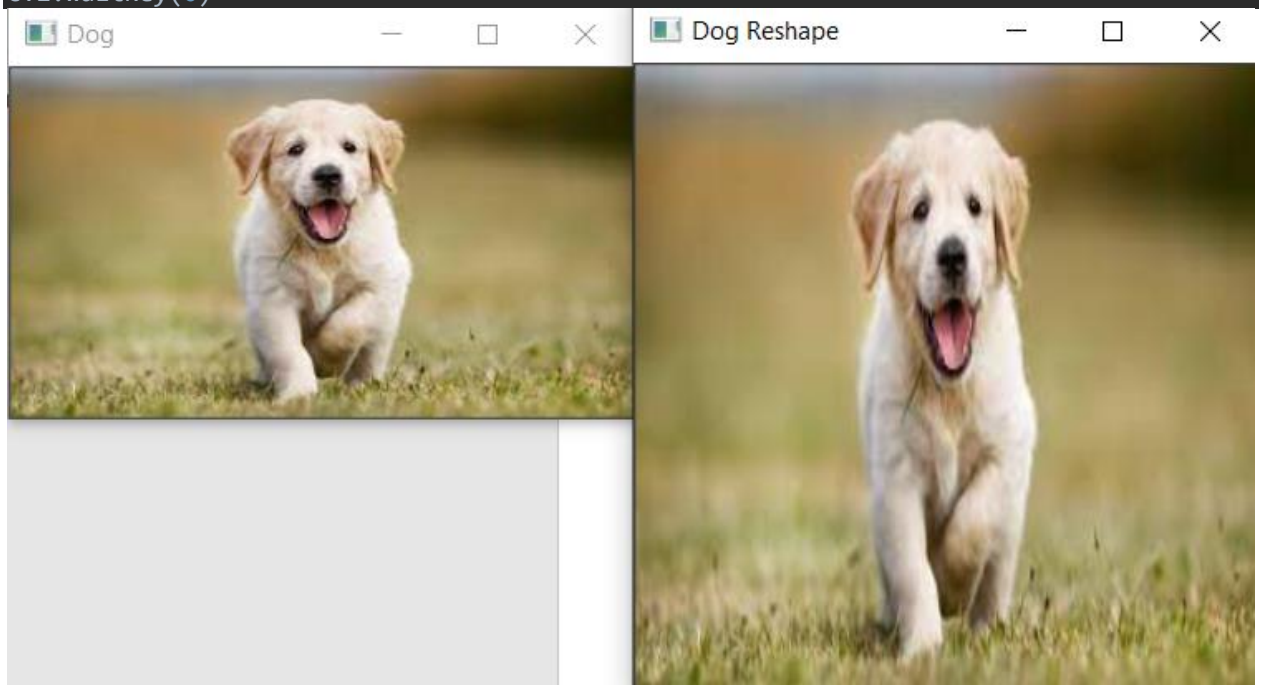
```
path = "C:\\Users\\Asus-2020\\PycharmProjects\\OCV\\Resources\\dog.jfif"
#Rgb image
#img =cv2.imread(path)

#Grey scale image
#Method 1
img =cv2.imread(path,0)
#Method 2
imggrey = cv2.cvtColor(img,cv2.COLOR_BAYER_BG2GRAY)

#Image Blur
#Method 1
imgblur = cv2.blur(imggrey,(7,7))
#Method2
#imgblur = cv2.GaussianBlur(imggrey,(7,7),0)

#Edge Detectors
#imgdet = cv2.Canny(imgblur,100,100)
imgdet = cv2.Canny(imgblur,40,50)

#image Dilation
imgdilation = cv2.dilate(imgdet,kernel,iterations=1)

#Erosion
imgerode = cv2.erode(imgdilation,kernel,iterations=1)
cv2.imshow('Jasvee Greyscale',img)
cv2.imshow('Jasvee Greyscale 1',imggrey)
cv2.imshow('Blur',imgblur)
cv2.imshow('Edge',imgdet)
cv2.imshow('Dilation',imgdilation)
cv2.imshow('Erosion',imgerode)
cv2.waitKey(0)
```



3. Resize and Crop images

Resize:

```
import cv2

path = "C:\\Users\\Asus-2020\\PycharmProjects\\OCV\\Resources\\dog.jfif"
img = cv2.imread(path)

#resize
print(img.shape)
wd,ht = 300,300
imgresize = cv2.resize(img,(wd,ht))
print(imgresize.shape)

cv2.imshow("Dog",img)
cv2.imshow('Dog Reshape',imgresize)
cv2.waitKey(0)
```



Crop:

```
import cv2

path = "C:\\Users\\Asus-2020\\PycharmProjects\\OCV\\Resources\\dog.jfif"
img = cv2.imread(path)

#resize
print(img.shape)
wd,ht = 300,300
imgresize = cv2.resize(img,(wd,ht))
print(imgresize.shape)

#crop
```

```
#height and width
imgcrp = img[10:165,90:200]

#cv2.imshow("Dog",img)
#cv2.imshow('Dog Reshape',imgresize)
cv2.imshow('Dog Crop',imgcrp)
cv2.waitKey(0)
```

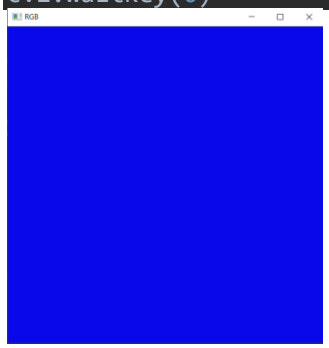
Cropped image into original picture size dimension:

```
import cv2

path = "C:\\Users\\Asus-2020\\PycharmProjects\\OCV\\Resources\\dog.jfif"
img = cv2.imread(path)

#resize
print(img.shape)
wd,ht = 300,300
imgresize = cv2.resize(img,(wd,ht))
print(imgresize.shape)

#crop
#height and width
imgcrp = img[10:165,90:200]

#original image size
imgcrpsize = cv2.resize(imgcrp,(img.shape[1],img.shape[0]))
cv2.imshow("Dog",img)
cv2.imshow('Dog Reshape',imgresize)
cv2.imshow('Dog Crop',imgcrp)
cv2.imshow('Org Img',imgcrpsize)
cv2.waitKey(0)
```

4. Draw shapes and text

Setting background color:

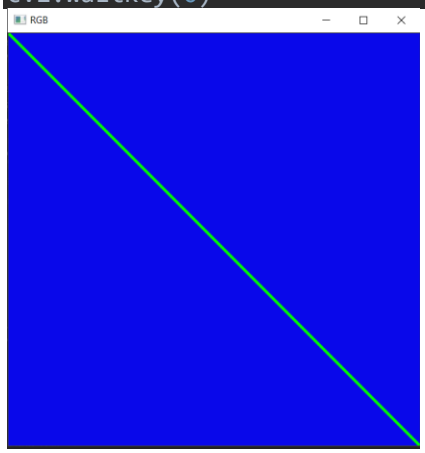
```
import cv2
import numpy as np

img = np.zeros((512,512,3),np.uint8)
print(img)
print(img.shape)
#setting color
img[:] = 234,8,9
#img[200:200,200:200] = 234,8,9

#Draw lines
#starting point, ending point, color, thickness
#cv2.line(img,(0,0),(100,100),(0,255,0),2)
#cv2.line(img,(0,0),(img.shape[1],img.shape[0]),(0,255,0),2)
cv2.imshow("RGB",img)
cv2.waitKey(0)
```



Draw line:

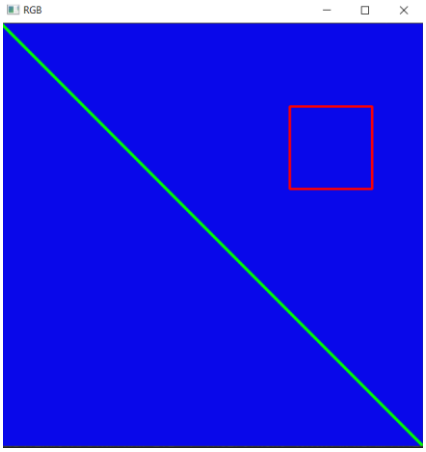
```
import cv2
import numpy as np
```

```
img = np.zeros((512,512,3),np.uint8)
print(img)
print(img.shape)
#setting color
img[:] = 234,8,9
#img[200:200,200:200] = 234,8,9

#Draw lines
#starting point, ending point, color, thickness
#cv2.line(img,(0,0),(100,100),(0,255,0),2)
cv2.line(img,(0,0),(img.shape[1],img.shape[0]),(0,255,0),2)
cv2.imshow("RGB",img)
cv2.waitKey(0)
```



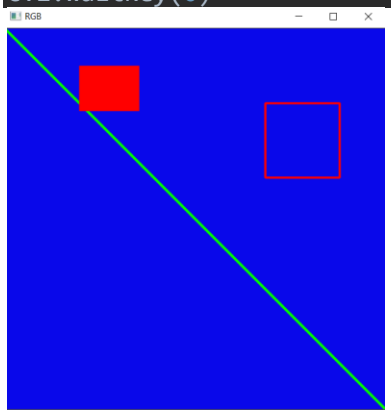
Draw Rectangle:

```
import cv2
import numpy as np

img = np.zeros((512,512,3),np.uint8)
print(img)
print(img.shape)
#setting color
img[:] = 234,8,9
#img[200:200,200:200] = 234,8,9

#Draw lines
#starting point, ending point, color, thickness
#cv2.line(img,(0,0),(100,100),(0,255,0),2)
cv2.line(img,(0,0),(img.shape[1],img.shape[0]),(0,255,0),2)
#draw rectangle
cv2.rectangle(img,(350,100),(450,200),(0,0,255),2)

cv2.imshow("RGB",img)
cv2.waitKey(0)
```



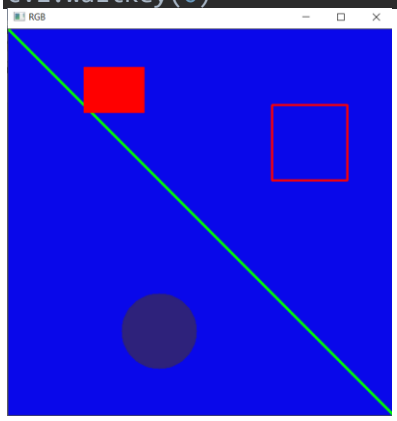
Fill color inside the rectangle

```
import cv2
import numpy as np

img = np.zeros((512,512,3),np.uint8)
#print(img)
print(img.shape)
#setting color
img[:] = 234,8,9
#img[200:200,200:200] = 234,8,9

#Draw lines
#starting point, ending point, color, thickness
#cv2.line(img,(0,0),(100,100),(0,255,0),2)
cv2.line(img,(0,0),(img.shape[1],img.shape[0]),(0,255,0),2)
#draw rectangle
cv2.rectangle(img,(350,100),(450,200),(0,0,255),2)
#draw rectangle and fill color
cv2.rectangle(img,(100,50),(180,110),(0,0,255),cv2.FILLED)

cv2.imshow("RGB",img)
cv2.waitKey(0)
```



Draw Circle:

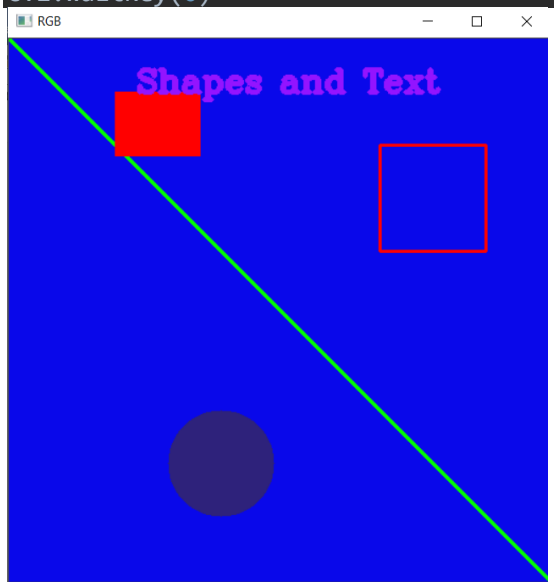
```
import cv2
import numpy as np

img = np.zeros((512,512,3),np.uint8)
#print(img)
print(img.shape)
#setting color
img[:] = 234,8,9
#img[200:200,200:200] = 234,8,9

#Draw lines
#starting point, ending point, color, thickness
#cv2.line(img,(0,0),(100,100),(0,255,0),2)
cv2.line(img,(0,0),(img.shape[1],img.shape[0]),(0,255,0),2)
#draw rectangle
cv2.rectangle(img,(350,100),(450,200),(0,0,255),2)
#draw rectangle and fill color
cv2.rectangle(img,(100,50),(180,110),(0,0,255),cv2.FILLED)
#Circle
cv2.circle(img,(200,400),50,(123,34,46),cv2.FILLED)

cv2.imshow("RGB",img)
cv2.waitKey(0)
```



Adding Text:

```
import cv2
import numpy as np

img = np.zeros((512,512,3),np.uint8)
#print(img)
print(img.shape)
#setting color
img[:] = 234,8,9
#img[200:200,200:200] = 234,8,9

#Draw lines
#starting point, ending point, color, thickness
#cv2.line(img,(0,0),(100,100),(0,255,0),2)
```

```
cv2.line(img,(0,0),(img.shape[1],img.shape[0]),(0,255,0),2)
#draw rectangle
cv2.rectangle(img,(350,100),(450,200),(0,0,255),2)
#draw rectangle and fill color
cv2.rectangle(img,(100,50),(180,110),(0,0,255),cv2.FILLED)
#Circle
cv2.circle(img,(200,400),50,(123,34,46),cv2.FILLED)
#Text
#text, position,font type,scale of the font,color,thinkness
cv2.putText(img,'Shapes and Text',(120,50),cv2.FONT_HERSHEY_COMPLEX,1,(255,20,147),3)
cv2.imshow("RGB",img)
cv2.waitKey(0)
```



5. Merge two images into a single image(Stacking)

```
import cv2
import numpy as np

img1 = cv2.imread("C:\\Users\\Asus-2020\\PycharmProjects\\OCV\\Resources\\dog.jfif")
img2 = cv2.imread("C:\\Users\\Asus-2020\\PycharmProjects\\OCV\\Resources\\dog1.jfif")
print(img1.shape)
print(img2.shape)
wd,ht = 300,168
img2resize = cv2.resize(img2,(wd,ht))
print(img2resize.shape)
#reshape the two images with same size
imga = cv2.resize(img1, (0,0), None, 0.5, 0.5)
imgb = cv2.resize(img2resize, (0,0), None, 0.5, 0.5)
print(imga.shape)
print(imgb.shape)
hor = np.hstack((imga,imgb))
ver = np.vstack((imga,imgb))

cv2.imshow('Vertical',ver)
```

```
cv2.imshow('Horizontal',hor)
cv2.waitKey(0)
```