

Read an image

In []:

```
import numpy as np
import cv2
img=cv2.imread('zelda.png') # cv2.imread_color(), cv2.imread_grayscale(), cv2.imread
print(img.shape)
cv2.imshow('Original',img)
cv2.waitKey(0) ##The function waits for specified milliseconds for any keyboard event
cv2.destroyAllWindows() # destroys all the windows we created
#to destroy any specific window, use the function cv2.destroyWindow() where you pass
```

(512, 512, 3)

Save an image in different format

In []:

```
import numpy as np
import cv2
img=cv2.imread('zelda.png')
print(img.shape)
img2=cv2.imwrite('zelda.jpg',img)
print(img2)
cv2.imshow('Original',img2)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Drawing a rectangle

In []:

```
import numpy as np
import cv2
pic=np.zeros((500,500,3),dtype='uint8')#500*500 image, 3 channels red green and blue
cv2.rectangle(pic,(0,0),(500,150),(123,200,98),3,lineType=cv2.LINE_AA,shift=0)
cv2.imshow('dark',pic)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Drawing a line

In []:

```
import numpy as np
import cv2
pic=np.zeros((500,500,3),dtype='uint8')
cv2.line(pic,(350,350),(500,350),(0,0,255))#starting point of line, ending point of
cv2.imshow('dark',pic)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Drawing a circle

In []:

```
import numpy as np
import cv2
pic=np.zeros((500,500,3),dtype='uint8')
cv2.circle(pic,(250,250),100,(255,0,255))#coordinates of circle, radius of circle,
cv2.imshow('dark',pic)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Writing text

In []:

```
import numpy as np
import cv2
pic=np.zeros((500,500,3),dtype='uint8')
cv2.putText(pic,'Opencv in python',(100,100),cv2.FONT_HERSHEY_DUPLEX,3,(255,255,255))
cv2.imshow('dark',pic)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Image Transformation

In []:

```
import cv2
import numpy as np
pic=cv2.imread('zelda.jpg')
cols=pic.shape[1]
rows=pic.shape[0]
M=np.float32([[1,0,150],[0,1,70]])#transition matrix as floating point array, shift
shifted=cv2.warpAffine(pic,M,(cols,rows))#transition function warpAffine
cv2.imshow('shifted',shifted)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Image Rotation

In []:

```
import cv2
import numpy as np
pic=cv2.imread('zelda.jpg')
cols=pic.shape[1]
rows=pic.shape[0]
center=(cols/2,rows/2)
angle=90
M=cv2.getRotationMatrix2D(center,angle,1)#1 for scaling(changing size) image
rotated=cv2.warpAffine(pic,M,(cols,rows))#transition function warpAffine
cv2.imshow('rotated',rotated)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Image Thresholding

In []:

```
import cv2
import numpy
pic=cv2.imread('zelda.png',0)
threshold_value=100
(T_value,binary_threshold)=cv2.threshold(pic,threshold_value,255,cv2.THRESH_BINARY)
(T_value,binary_threshold)=cv2.threshold(pic,threshold_value,255,cv2.THRESH_BINARY)
cv2.imshow('binary',binary_threshold)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Gaussian Blur(Image Filtering)

In []:

```
import cv2
import numpy
pic=cv2.imread('zelda.png')
matrix=(7,7)
blur=cv2.GaussianBlur(pic,matrix,0)
cv2.imshow('blurred',blur)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Median Blur(Image Filtering)

In []:

```
import cv2
import numpy
pic=cv2.imread('zelda.png')
kernel=3
median=cv2.medianBlur(pic,kernel)
cv2.imshow('median',median)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Bilateral filtering

In []:

```
import cv2
import numpy
pic=cv2.imread('zelda.png')
dimpixel=7
color=100
space=100
filter=cv2.bilateralFilter(pic,dimpixel,color,space)
cv2.imshow('filtered',filter)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Canny Edge Detector

In []:

```
import cv2
import numpy
pic=cv2.imread('zelda.png')
threshold_value1=50
threshold_value2=100
canny=cv2.Canny(pic,threshold_value1,threshold_value2)#values<50 is 0 and values>100 is 255
cv2.imshow('canny',canny)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Load video

In [2]:

```

import cv2
import numpy
cap=cv2.VideoCapture('sample.mp4')
while(cap.isOpened()): #check whether cap is initialized or not by the method cap.is
    ret,frame=cap.read()#cap.read() returns a bool (True/False)
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    cv2.imshow('vid',gray)
    if cv2.waitKey(1) & 0xFF==ord('q'):
        break
cap.release()
cv2.destroyAllWindows()

```

```

-----
-----
error                                Traceback (most recent call
last)
<ipython-input-2-00a349346e5f> in <module>
      4 while(cap.isOpened()): #check whether cap is initialized or no
t by the method cap.isOpened().
      5     ret,frame=cap.read()#cap.read() returns a bool (True/Fals
e)
----> 6     gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
      7     cv2.imshow('vid',gray)
      8     if cv2.waitKey(1) & 0xFF==ord('q'):

```

```

error: OpenCV(4.1.2) /io/opencv/modules/imgproc/src/color.cpp:182: err
or: (-215:Assertion failed) !_src.empty() in function 'cvtColor'

```

Save video in different format

In [2]:

```

import cv2
import numpy
cap=cv2.VideoCapture('sample.mp4')
fourcc=cv2.VideoWriter_fourcc(*'XVID') #FourCC is a 4-byte code used to specify the
fps=30
framesize=(720,480)
out=cv2.VideoWriter('sample.avi',fourcc,fps,framesize)
while(cap.isOpened()):
    ret,frame=cap.read()
    cv2.imshow('frame',frame)
    if cv2.waitKey(1) & 0xFF==ord('q'):
        break
cap.release()
cv2.destroyAllWindows()

```


 error Traceback (most recent call
 last)

```

<ipython-input-2-2723b0712700> in <module>
      8 while(cap.isOpened()):
      9     ret,frame=cap.read()
--> 10     cv2.imshow('frame',frame)
     11     if cv2.waitKey(1) & 0xFF==ord('q'):
     12         break

```

error: OpenCV(4.1.2) /io/opencv/modules/highgui/src/window.cpp:376: error: (-215:Assertion failed) size.width>0 && size.height>0 in function 'imshow'

Face Detection using Web Cam

In [1]:

```

import cv2
import numpy as np
face_cascade=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
pic=cv2.imread('zelda.jpg')
scale_factor=1.3
while 1:
    faces=face_cascade.detectMultiScale(pic,scale_factor,5)
    for (x,y,w,h) in faces:
        cv2.rectangle(pic,(x,y),(x+w,y+h),(255,0,0),2)
        font=cv2.FONT_HERSHEY_SIMPLEX
        cv2.putText(pic,'Zelda',(x,y),font,2,(255,255,255),2,cv2.LINE_AA)
        print("Number of Faces found{}",format(len(faces)))
        cv2.imshow('face',pic)
        k=cv2.waitKey(30) & 0xff
        if k ==2:
            break
cv2.destroyAllWindows()

```

KeyboardInterrupt

Traceback (most recent call
last)

```

<ipython-input-1-f8cbf45baaa2> in <module>
      5 scale_factor=1.3
      6 while 1:
----> 7     faces=face_cascade.detectMultiScale(pic,scale_factor,5)
      8     for (x,y,w,h) in faces:
      9         cv2.rectangle(pic,(x,y),(x+w,y+h),(255,0,0),2)

```

KeyboardInterrupt:

In []:

```

k = cv2.waitKey(0) & 0xFF
if k == 27:      # wait for ESC key to exit
    cv2.destroyAllWindows()
elif k == ord('s'): # wait for 's' key to save and exit

```