

PRODDEC PYTHON DAY 9

Resizing an image

```
In [2]: import cv2  
img=cv2.imread("parrot.jpg",1)  
print(img.shape)  
  
# Resizing image to 300x200 pixel.  
resized_img=cv2.resize(img,(300,200))  
cv2.imwrite("resized_image.jpg",resized_img)
```

Out[2]: True

Image to grey and resize

```
In [3]: import cv2  
img=cv2.imread("parrot.jpg",0)  
  
# Resizing image to 10x10 pixel.  
resized_img=cv2.resize(img,(10,10))  
cv2.imwrite("resized_image_10.jpg",resized_img)  
print(resized_img) # These are pixel values in the grey image.
```

Creating a new image using the list

```
In [6]: import cv2  
import numpy as np  
  
# This will create a Four.  
img=[[255, 0, 0, 0, 255],  
     [255, 0, 0, 0, 255],  
     [255, 255, 255, 255, 255],  
     [0, 0, 0, 0, 255]]  
img=np.array(img)  
cv2.imwrite("4.jpg",img)
```

Out[6]: True



Creating new image using the numpy

```
In [4]: import cv2  
import numpy as np  
# np.zeros create a List with elements as zero.  
# Creating a 3D List for pixels in the black image.  
black_img=np.zeros((512,512,3),np.uint8)  
cv2.imwrite("black_image.jpg",black_img)
```

Out[4]: True



Drawing the shape on an image

Fonts in Opencv

```
In [ ]: """  
cv2.FONT_HERSHEY_SIMPLEX  
cv2.FONT_HERSHEY_PLAIN  
cv2.FONT_HERSHEY_DUPLEX  
cv2.FONT_HERSHEY_COMPLEX  
cv2.FONT_HERSHEY_TRIPLEX  
cv2.FONT_HERSHEY_COMPLEX_SMALL  
cv2.FONT_HERSHEY_SCRIPT_SIMPLEX  
cv2.FONT_HERSHEY_SCRIPT_COMPLEX  
"""
```

Drawing line on the image

```
In [8]: """  
In opencv coordinate system starts from top left  
Top-Left is (0,0)  
X->Moving-Right  
Y->Moving-Down  
"""  
import cv2  
import numpy as np  
img=np.zeros((512,512,3),np.uint8)  
# Arguments-> img,startng-coordinate,ending-coordinate,color,thickness  
cv2.line(img,(0,0),(511,511),(0,255,0),3)  
cv2.line(img,(511,0),(0,511),(0,0,255),3)  
cv2.imshow("line",img)  
cv2.waitKey(0)  
cv2.destroyAllWindows()  
cv2.imwrite("lines.jpg",img)
```

Out[8]: True



Drawing rectangle on the image

```
In [12]: import cv2  
import numpy as np  
img=np.zeros((512,512,3),np.uint8)  
# Arguments:  
# img,top-left-coordinate,bottom-below-coordinate,color,thickness/fill(-1)  
cv2.rectangle(img,(100,100),(400,400),(0,255,0),3)  
cv2.rectangle(img,(150,150),(350,350),(0,255,255),3)  
cv2.waitKey(0)  
cv2.destroyAllWindows()  
cv2.imwrite("rectangle.jpg",img)
```

Out[12]: True



Drawing circle on image

```
In [17]: import cv2  
import numpy as np  
img=np.zeros((512,512,3),np.uint8)  
# Arguments-> img,center-coordinate,radius,color,thickness/fill(-1)  
cv2.circle(img,(256,256),200,(0,255,0),-1)  
cv2.imwrite("circle.jpg",img)
```

Out[17]: True

Example 1

```
In [4]: import cv2  
import numpy as np  
  
img=np.zeros((600,600,3),np.uint8)  
for i in range(600,600,40):  
    for j in range(20,600,40):  
        cv2.circle(img,(i,j),20,(0,255,0),1)  
  
cv2.imwrite("pattern-1.jpg",img)
```

Out[4]: True

Example 2

```
In [12]: import cv2  
import numpy as np  
  
img=np.zeros((600,600,3),np.uint8)  
for i in range(20,600,40):  
    for j in range(20,600,40):  
        cv2.circle(img,(i,j),20,(0,0,255),-1)  
  
cv2.imwrite("pattern-2.jpg",img)
```

Out[12]: True

Example 3

```
In [33]: # Code for creating number (1-100) inside images  
import cv2  
import numpy as np  
  
for i in range(1,101):  
    img=np.zeros((500,500,3),np.uint8)  
    text=str(i).zfill(3)  
    cv2.putText(img,text,(10,250),cv2.FONT_HERSHEY_SIMPLEX,4,(0,255,0),15)  
    cv2.imwrite("text"+str(i)+".jpg",img)
```

Out[33]: True

Getting frames from video

```
In [2]: import cv2  
video_name="video.mp4"  
  
# Creating video capture object.  
cap=cv2.VideoCapture(video_name)  
  
if cap.isOpened()==False:  
    print("Error reading video")
```

```
while(cap.isOpened()): # Trying for reading frame.  
    ret,img=cap.read() # Reading the frame  
    if ret==True:  
        grey = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # Converting to grey.  
        cv2.imshow("Video frame",img)  
        key=cv2.waitKey(1)  
        if key==100: # Press d button for stop playing video.  
            break  
    else:  
        break
```

```
cap.release() # Releasing the frame.  
cv2.destroyAllWindows()
```

Out[2]: True

Getting a image from a webcam

```
In [6]: import cv2  
# Video capture object  
cap=cv2.VideoCapture(0) # 0->primary camera 1->secondary camera
```

```
if cap.isOpened()==False:  
    print("Error reading frame")
```

```
while(cap.isOpened()): # Trying for reading frame.  
    ret,img=cap.read() # Reading the frame  
    if ret==True:  
        grey = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # Converting to grey.
```

```
        cv2.imshow("Video frame",img)  
        key=cv2.waitKey(1)  
        if key==100: # Press d button for stop playing video.  
            break  
    else:  
        break
```

```
cap.release() # Releasing the frame.  
cv2.destroyAllWindows()
```

Out[6]: True

Getting frames from Webcam/USB-camera

```
In [4]: import cv2  
video_name="video.mp4"  
  
# Creating video capture object.  
cap=cv2.VideoCapture(0) # 0->primary camera 1->secondary camera
```

```
if cap.isOpened()==False:  
    print("Error reading video")
```

```
while(cap.isOpened()): # Trying for reading frame.  
    ret,img=cap.read() # Reading the frame  
    if ret==True:  
        grey = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY) # Converting to grey.
```

```
        cv2.imshow("Video frame",img)  
        key=cv2.waitKey(1)  
        if key==100: # Press d button for stop playing video.  
            break  
    else:  
        break
```

```
cap.release() # Releasing the frame.  
cv2.destroyAllWindows()
```

Out[4]: True

