

OpenCV_1

March 28, 2023

```
[1]: import cv2
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

```
[2]: plt.figure(figsize=(5,3))
img = plt.imread('CuteDog.jpg')
plt.imshow(img)
plt.grid(False)
plt.show()
```

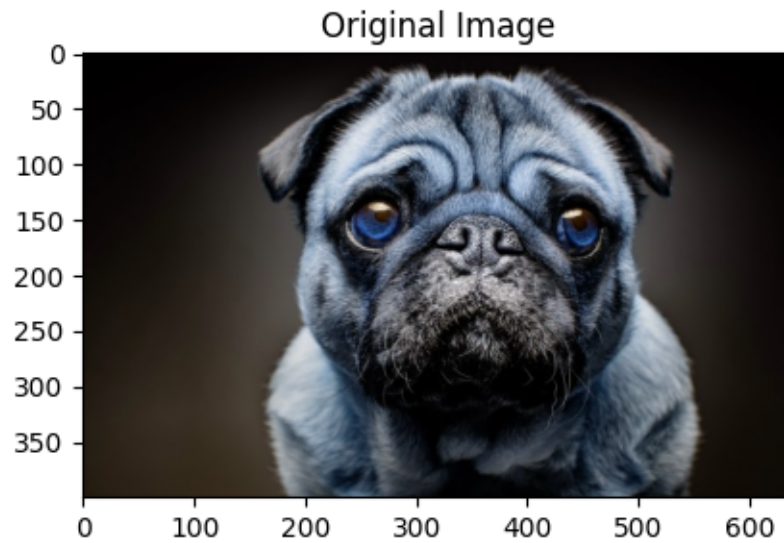


```
[3]: img.shape
```

```
[3]: (400, 640, 3)
```

```
[4]: def catImageshow(imageTitle,image):
    color_convert = cv2.cvtColor(image,cv2.COLOR_BGR2RGB)
    plt.figure(figsize=(5,3))
```

```
plt.imshow(color_convert)
plt.title(imageTitle)
plt.show()
catImageshow("Original Image",img)
```



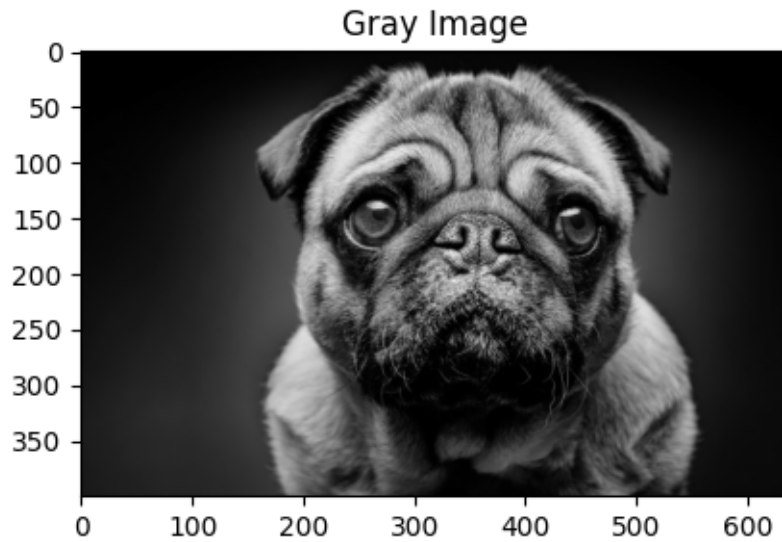
1 Gray

```
[5]: gray = cv2.cvtColor(img,cv2.COLOR_RGB2GRAY)
```

```
[6]: gray.shape
```

```
[6]: (400, 640)
```

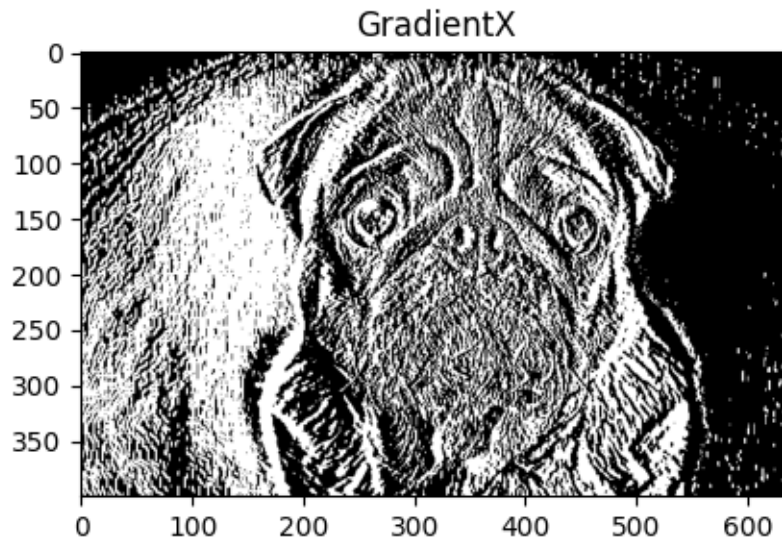
```
[7]: catImageshow("Gray Image",gray)
```



```
[8]: gradientX = cv2.Sobel(gray,ddepth = cv2.CV_32F,dx=1,dy=0,ksize = 3)
      gradientY = cv2.Sobel(gray,ddepth = cv2.CV_32F,dx=0,dy=1,ksize=3)
```

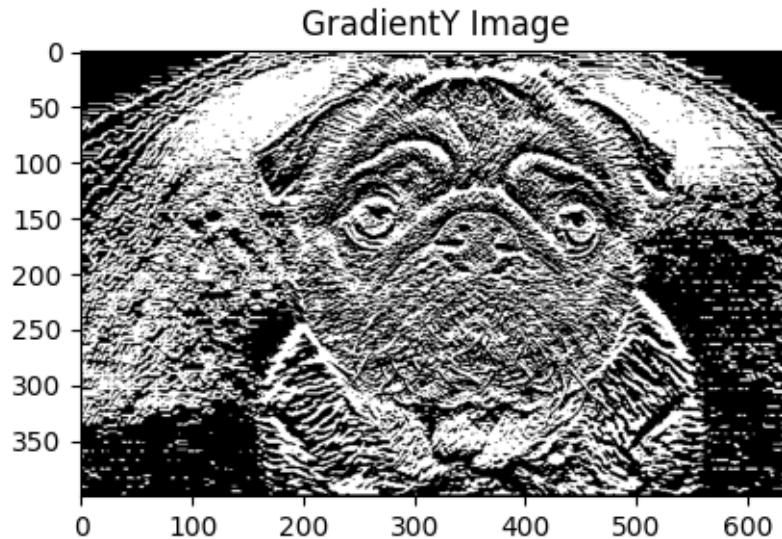
```
[9]: catImageshow("GradientX",gradientX)
```

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



```
[10]: catImageshow("GradientY Image",gradientY)
```

Clipping input data to the valid range for imshow with RGB data ([0..1] for floats or [0..255] for integers).



2 Masking

```
[11]: img.shape
```

```
[11]: (400, 640, 3)
```

```
[12]: img.shape[:2]
```

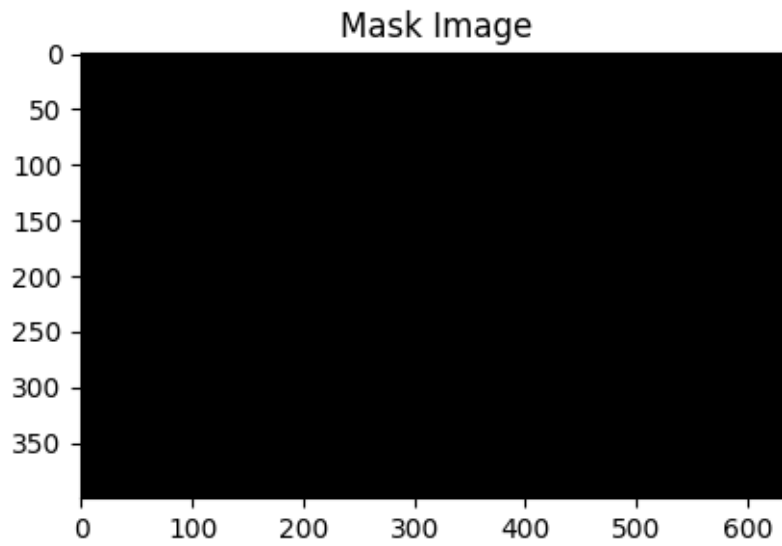
```
[12]: (400, 640)
```

```
[13]: mask = np.zeros(img.shape[:2],dtype="uint8")
```

```
[14]: mask
```

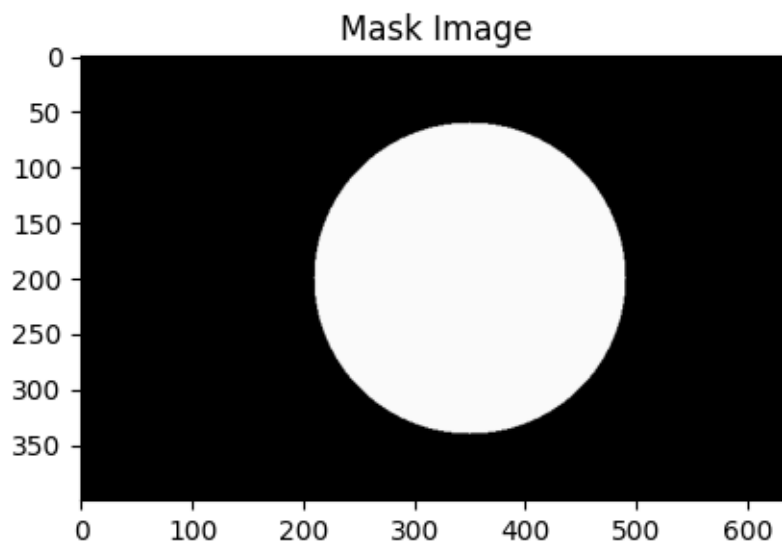
```
[14]: array([[0, 0, 0, ..., 0, 0, 0],
          [0, 0, 0, ..., 0, 0, 0],
          [0, 0, 0, ..., 0, 0, 0],
          ...,
          [0, 0, 0, ..., 0, 0, 0],
          [0, 0, 0, ..., 0, 0, 0],
          [0, 0, 0, ..., 0, 0, 0]], dtype=uint8)
```

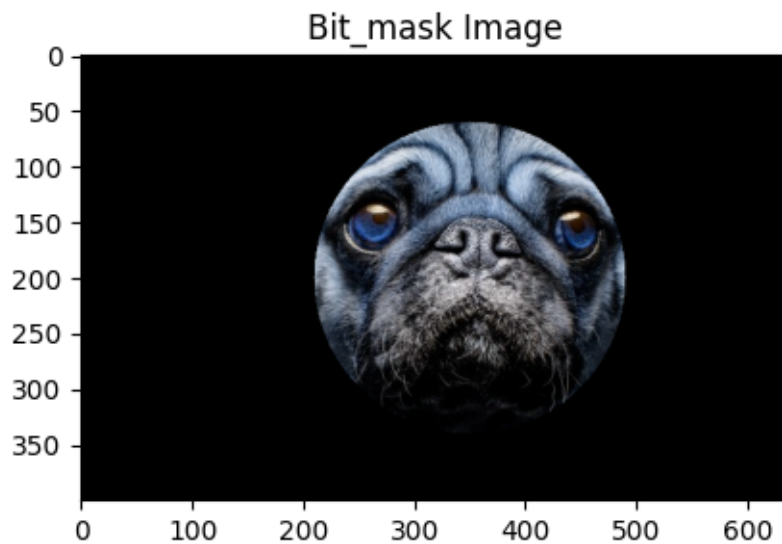
```
[15]: cv2.rectangle(mask,(100, 500), (150, 100), 0)
      catImageShow("Mask Image",mask)
```



3 Bit_Mask

```
[16]: cv2.circle(mask, (350, 200), 140, 250, -1)
      bit_mask = cv2.bitwise_and(img,img,mask = mask)
      catImageshow("Mask Image",mask)
      catImageshow("Bit_mask Image",bit_mask)
```





4 Image Scalling

```
[17]: img.shape
```

```
[17]: (400, 640, 3)
```

```
[18]: img/255
```

```
[18]: array([[[0.      , 0.      , 0.      ],
             [0.      , 0.      , 0.      ],
             [0.      , 0.      , 0.      ],
             ...,
             [0.      , 0.      , 0.      ],
             [0.      , 0.      , 0.      ],
             [0.      , 0.      , 0.      ]],

            [[0.      , 0.      , 0.      ],
             [0.      , 0.      , 0.      ],
             [0.      , 0.      , 0.      ],
             ...,
             [0.      , 0.      , 0.      ],
             [0.      , 0.      , 0.      ],
             [0.      , 0.      , 0.      ]],

            [[0.      , 0.      , 0.      ],
             [0.      , 0.      , 0.      ],
             [0.      , 0.      , 0.      ]])
```

```

...,
[0.          , 0.          , 0.          ],
[0.          , 0.          , 0.          ],
[0.          , 0.          , 0.          ]],

...,

[[0.0627451 , 0.10196078, 0.1372549 ],
 [0.0627451 , 0.10196078, 0.1372549 ],
 [0.0627451 , 0.10196078, 0.1372549 ]],

...,
[0.02352941, 0.04313725, 0.05882353],
[0.02352941, 0.04313725, 0.05882353],
[0.02352941, 0.04313725, 0.05882353]],

[[0.0627451 , 0.10196078, 0.1372549 ],
 [0.0627451 , 0.10196078, 0.1372549 ],
 [0.0627451 , 0.10196078, 0.1372549 ]],

...,
[0.01960784, 0.03921569, 0.05490196],
[0.01960784, 0.03921569, 0.05490196],
[0.01960784, 0.03921569, 0.05490196]],

[[0.0627451 , 0.10196078, 0.1372549 ],
 [0.0627451 , 0.10196078, 0.1372549 ],
 [0.0627451 , 0.10196078, 0.1372549 ]],

...,
[0.01960784, 0.03921569, 0.05490196],
[0.01960784, 0.03921569, 0.05490196],
[0.01960784, 0.03921569, 0.05490196]]])

```

```
[19]: img.shape
```

```
[19]: (400, 640, 3)
```

5 Resize Image

```
[20]: customsizeH = 120/img.shape[0]
```

```
[21]: customsizeW = 120/img.shape[1]
```

```
[22]: customsizeW
```

```
[22]: 0.1875
```

```
[23]: customsizeH
```

[23]: 0.3

```
[24]: imgDimension = (120,int(img.shape[0]*customsizeW))
```

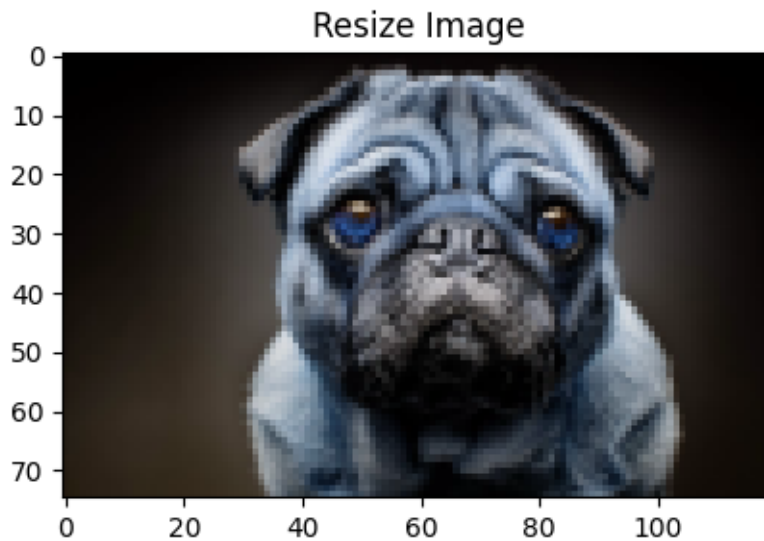
```
[25]: imgDimension
```

[25]: (120, 75)

```
[26]: img.shape
```

[26]: (400, 640, 3)

```
[27]: resizeImage = cv2.resize(img,imgDimension,interpolation = cv2.INTER_AREA)  
catImageShow("Resize Image",resizeImage)
```



```
[28]: resizeImage.shape
```

[28]: (75, 120, 3)

6 Rotate Image

```
[29]: (imageH,imageW) = img.shape[:2]
```

```
[30]: imageH
```

[30]: 400

```
[31]: imageW
```


[31]: 640

```
[32]: (centerX,centerY) = (imageH//2,imageW//2)
```

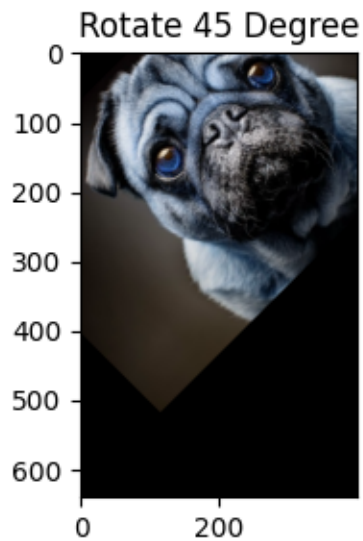
```
[33]: centerX
```

[33]: 200

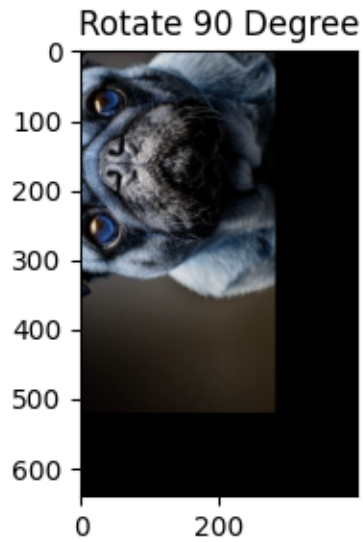
```
[34]: centerY
```

[34]: 320

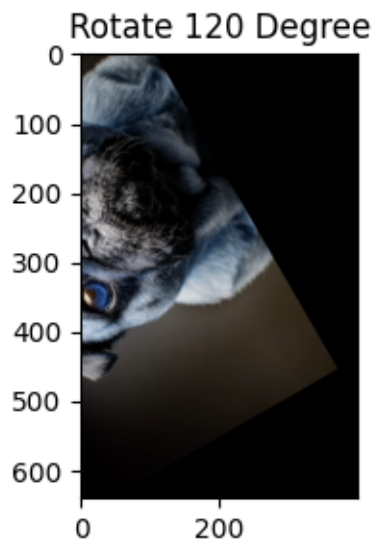
```
[35]: imageRorate = cv2.getRotationMatrix2D((centerX,centerY),45,1.0)
      rotateNow = cv2.warpAffine(img,imageRorate,(imageH,imageW))
      catImageshow("Rotate 45 Degree",rotateNow)
```



```
[36]: imageRorate = cv2.getRotationMatrix2D((centerX,centerY),90,1.0)
      rotateNow = cv2.warpAffine(img,imageRorate,(imageH,imageW))
      catImageshow("Rotate 90 Degree",rotateNow)
```



```
[37]: imageRorate = cv2.getRotationMatrix2D((centerX,centerY),120,1.0)  
rotateNow = cv2.warpAffine(img,imageRorate,(imageH,imageW))  
catImageshow("Rotate 120 Degree",rotateNow)
```



```
[38]: imageRorate = cv2.getRotationMatrix2D((centerX,centerY),180,1.0)  
rotateNow = cv2.warpAffine(img,imageRorate,(imageH,imageW))  
catImageshow("Rotate 180 Degree",rotateNow)
```

