

# 用课程所学内容实现各类图像增广

常用图像增广方法主要有：左右翻转(上下翻转对于许多目标并不常用)，随机裁剪，变换颜色(亮度，对比度，饱和度和色调)等等，我们拟用opencv-python实现部分数据增强方法。  
结构如下：

```
class FunctionClass:
    def __init__(self, parameter):
        self.parameter=parameter

    def __call__(self, img):
```

## 要求

- 1.补全代码
- 2.验证增强效果
- 3.可自选实现其他增强效果

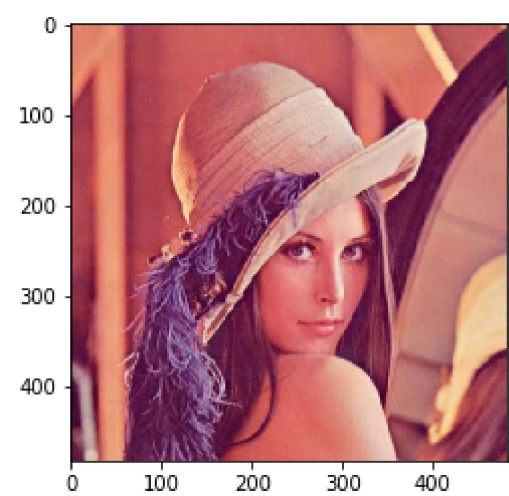
```
!pip install pillow
```

```
Looking in indexes: https://mirror.baidu.com/pypi/simple/
Requirement already satisfied: pillow in /opt/conda/envs/python35-paddle120-env/lib/python3.7/site-packages
```

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
from PIL import Image

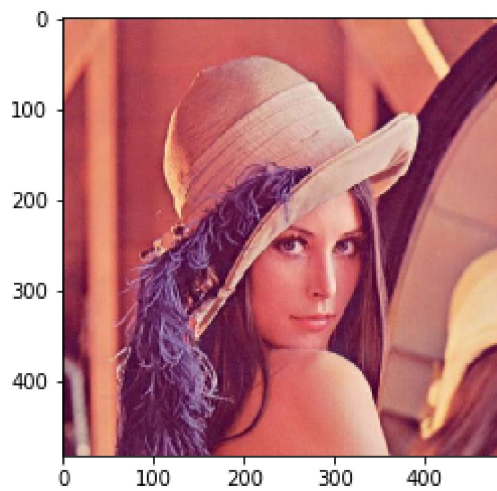
%matplotlib inline
```

```
img = Image.open('work/1.jpg')
plt.imshow(img)
plt.show()
```



```
#img = Image.open('1.jpg')
filename = 'work/1.jpg'
## [Load an image from a file]
img = cv2.imread(filename)
img = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
plt.imshow(img)
```

```
<matplotlib.image.AxesImage at 0x7f39c434f610>
```



```
print(img.shape)
```

```
(483, 482, 3)
```

## 1.图片缩放

```
class Resize:
    def __init__(self, size):
        self.size=size

    def __call__(self, img):
        return cv2.resize(img, self.size)
```

```
resize=Resize((600, 600))
img2=resize(img)
#print(img2.shape)
plt.imshow(img2)
```

```
<matplotlib.image.AxesImage at 0x7f69cc4d7a10>
```



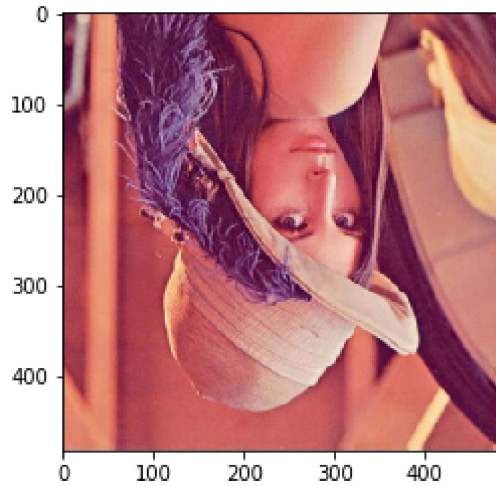
## 2.图片翻转

```
class Flip:
    def __init__(self, mode):
        self.mode=mode

    def __call__(self, img):
        return cv2.flip(img, self.mode)
```

```
flip=Flip(mode=0)
img2=flip(img)
plt.imshow(img2)
```

<matplotlib.image.AxesImage at 0x7f69b81cc790>



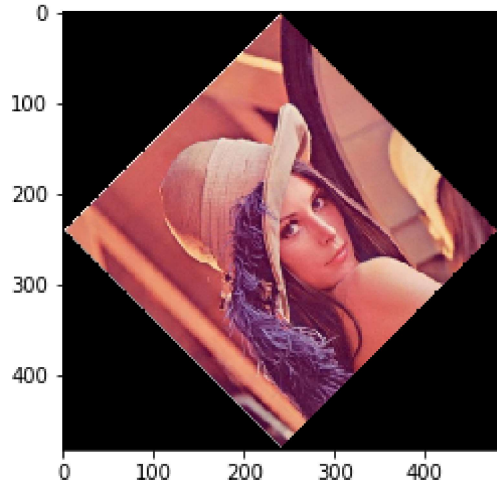
??cv2.rotate

## 3图片旋转

```
class Rotate:
    def __init__(self, degree,size):
        self.degree=degree
        self.size=size
    def __call__(self, img):
        (rows, cols,x) = img.shape
        M = cv2.getRotationMatrix2D(((cols - 1) / 2.0, (rows - 1) / 2.0), self.degree, self.size)
        return cv2.warpAffine(img, M, (cols, rows))
```

```
rotate=Rotate(45, 0.7)
img2=rotate(img)
plt.imshow(img2)
```

<matplotlib.image.AxesImage at 0x7f39c42f6110>

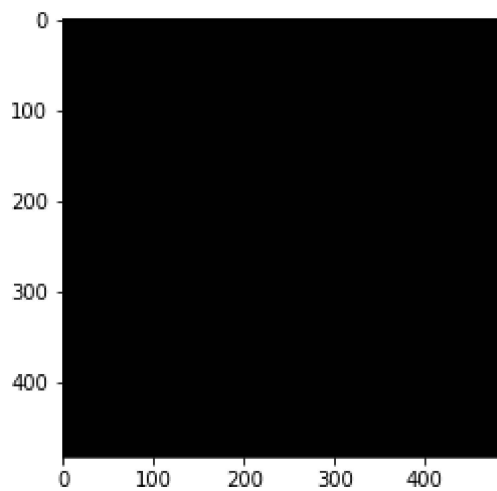


## 4.图片亮度调节

```
??cv2.CAP_PROP_BRIGHTNESS
```

```
rows, cols, x = img.shape
blank = np.zeros([rows, cols, x], img.dtype)
img2=brightness(blank)
plt.imshow(img2)
```

```
<matplotlib.image.AxesImage at 0x7f39c422b290>
```

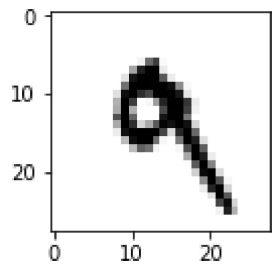


```
class Brightness:
    def __init__(self,brightness_factor):
        self.brightness_factor=brightness_factor

    def __call__(self, img):
        rows, cols, x = img.shape
        a = self.brightness_factor
        blank = np.zeros([rows, cols, x], img.dtype)
        return cv2.addWeighted(img, a, blank, 1-a, 1)

brightness=Brightness(0.6)
img2=brightness(img)
plt.imshow(img2)
```

```
<matplotlib.image.AxesImage at 0x7f39c4250890>
```



## 5.图片随机裁剪

```
import random
??random.uniform
```

```
import random
import math
```

```
class RandomErasing(object):
    def __init__(self, EPSILON=0.5, sl=0.02, sh=0.4, rl=0.3,
                 mean=[0., 0., 0.]):
        self.EPSILON = EPSILON
        self.mean = mean
        self.sl = sl
        self.sh = sh
        self.rl = rl

    def __call__(self, img):
        if random.uniform(0, 1) > self.EPSILON:
            return img

        for attempt in range(100):
            area = img.shape[0] * img.shape[1]

            target_area = random.uniform(self.sl, self.sh) * area
            aspect_ratio = random.uniform(self.rl, 1 / self.rl)

            h = int(round(math.sqrt(target_area * aspect_ratio)))
            w = int(round(math.sqrt(target_area / aspect_ratio)))

            if w < img.shape[0] and h < img.shape[1]:
                x1 = random.randint(0, img.shape[1] - h)
                y1 = random.randint(0, img.shape[0] - w)
                if img.shape[2] == 3:
                    img[ x1:x1 + h, y1:y1 + w, 0] = self.mean[0]
                    img[ x1:x1 + h, y1:y1 + w, 1] = self.mean[1]
                    img[ x1:x1 + h, y1:y1 + w, 2] = self.mean[2]
                else:
                    img[x1:x1 + h, y1:y1 + w, 0] = self.mean[0]
            return img
        return img
```

```
erase = RandomErasing()
img2=erase(img)
plt.imshow(img2)
```

```
<matplotlib.image.AxesImage at 0x7f6964168110>
```

