

# alumentations.HorizontalFlip、Rotate、RandomBrightnessContrast、ShiftScaleRotate、ToTensorV2等增强方法测试

原创

版权

jjw\_zyfx



已于 2022-04-16 11:54:06 修改



276



收藏

1

分类专栏：

pytorch

python

matplotlib

文章标签：

python

计算机视觉

pytorch



pytorch 同时被 3 个专栏收录 ▼

订阅专栏

专栏

0 订阅

5 篇文章

1 订阅

113 篇文章

2 订阅

64 篇文章

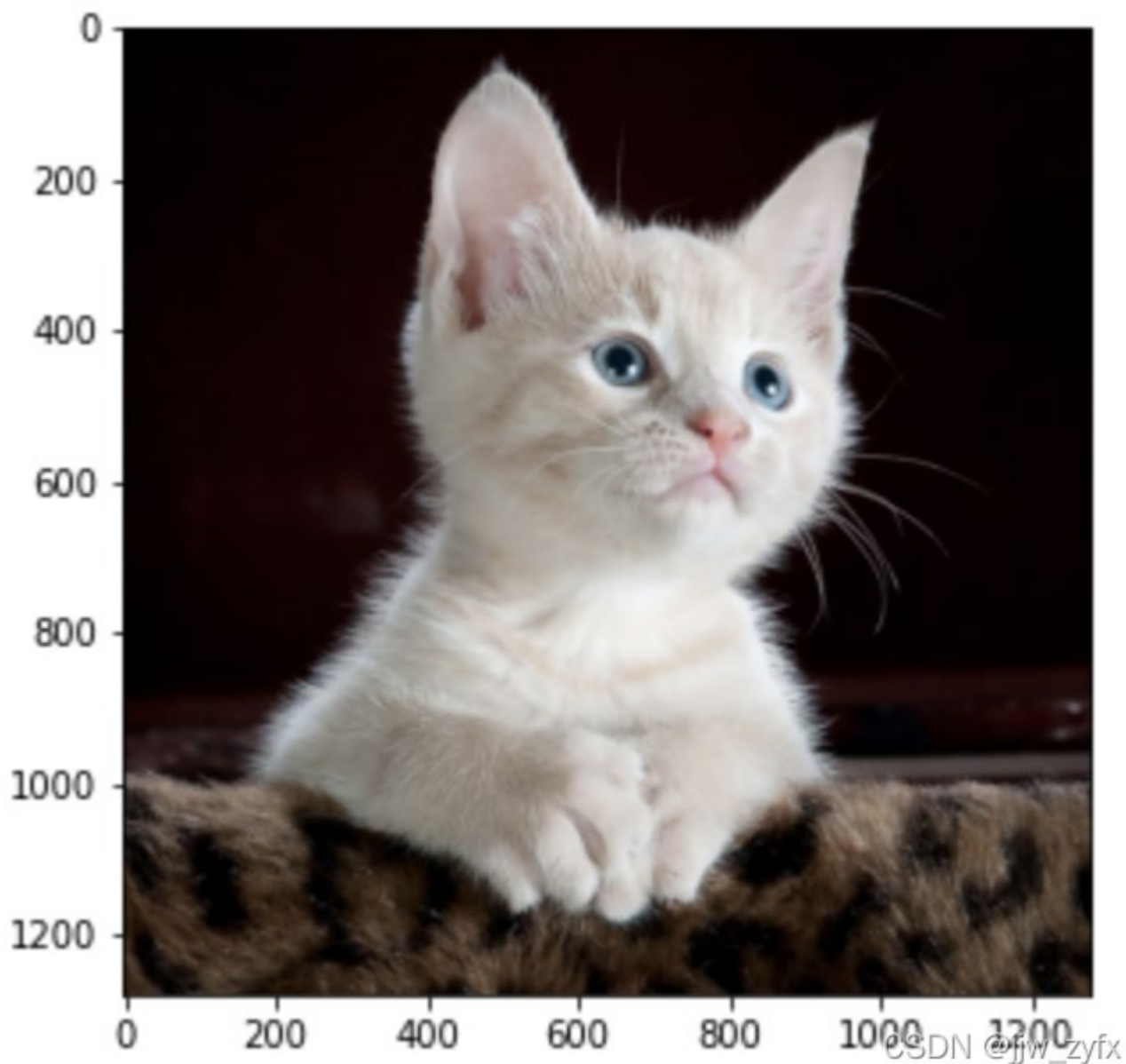
代码及效果如下：

先导包

```
1 import cv2
2 import alumentations
3 from alumentations.pytorch.transforms import ToTensorV2
4 import matplotlib.pyplot as plt
```

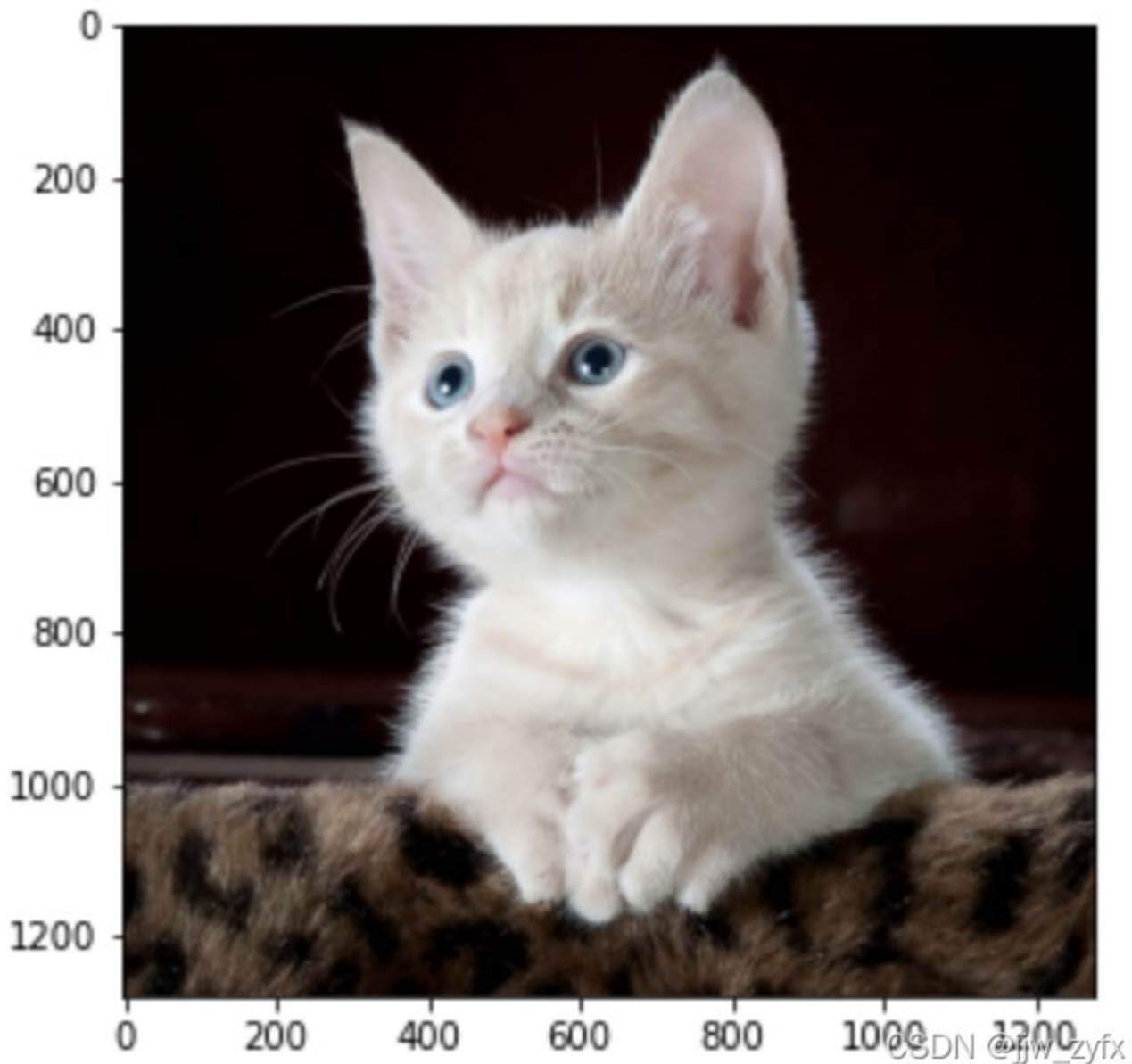
原图：

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 plt.figure(figsize=(5, 5))
4 plt.imshow(image) # 原始图片
```



## 水平翻转

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 # 水平翻转
4 image1 = albumentations.HorizontalFlip(always_apply=True, p=1.0)(image=image)['image']
5 plt.figure(figsize=(5, 5))
6 plt.imshow(image1)
```

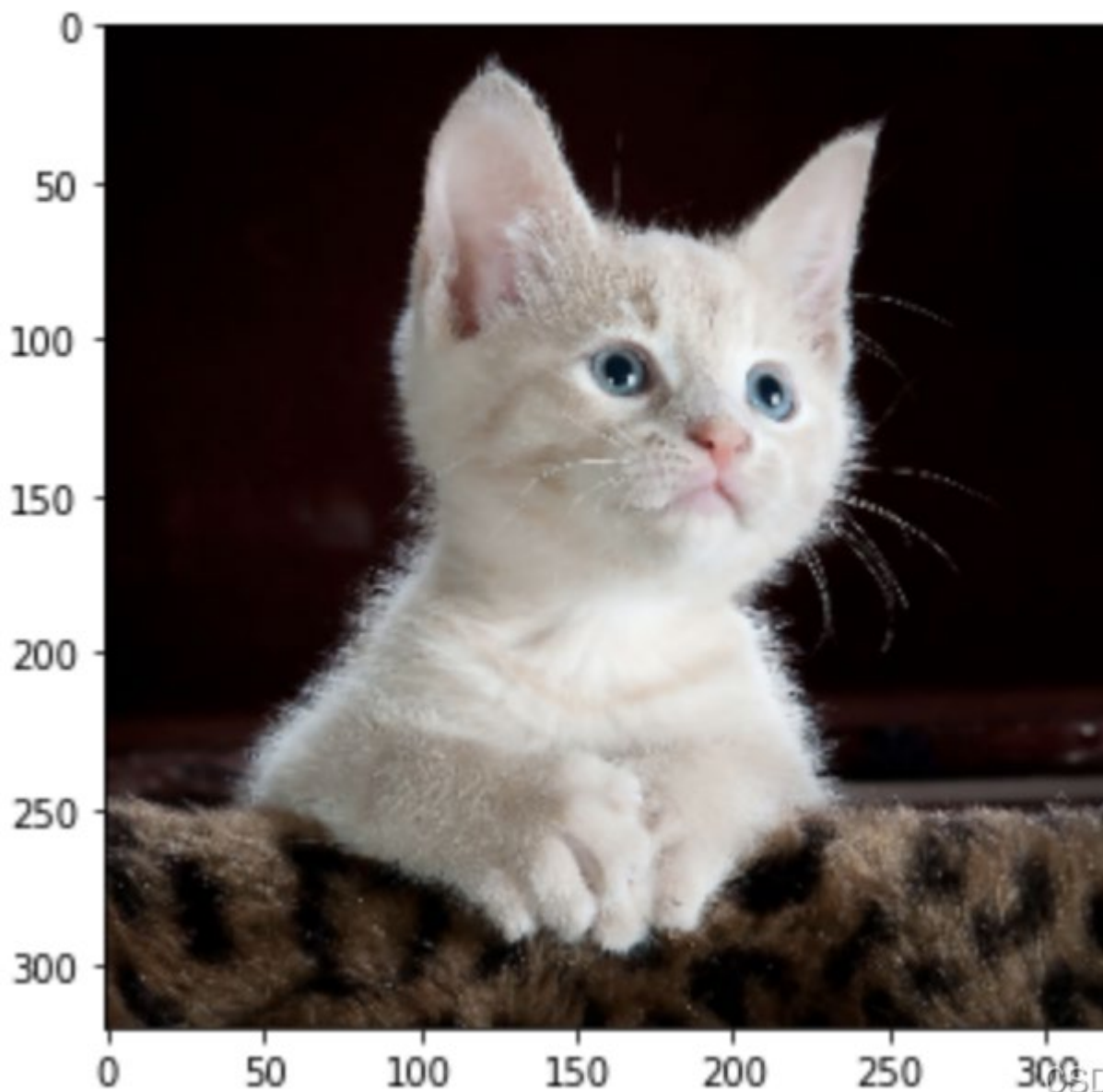


## 变换尺寸

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 print(image.shape)
4 # 变换尺寸, 变换后的尺寸为320, 320
5 image = albumentations.Resize(320, 320)(image=image)["image"]
6 print(image.shape)
7 plt.figure(figsize=(5, 5))
8 plt.imshow(image)
```

```
(1279, 1280, 3)
(320, 320, 3)
```

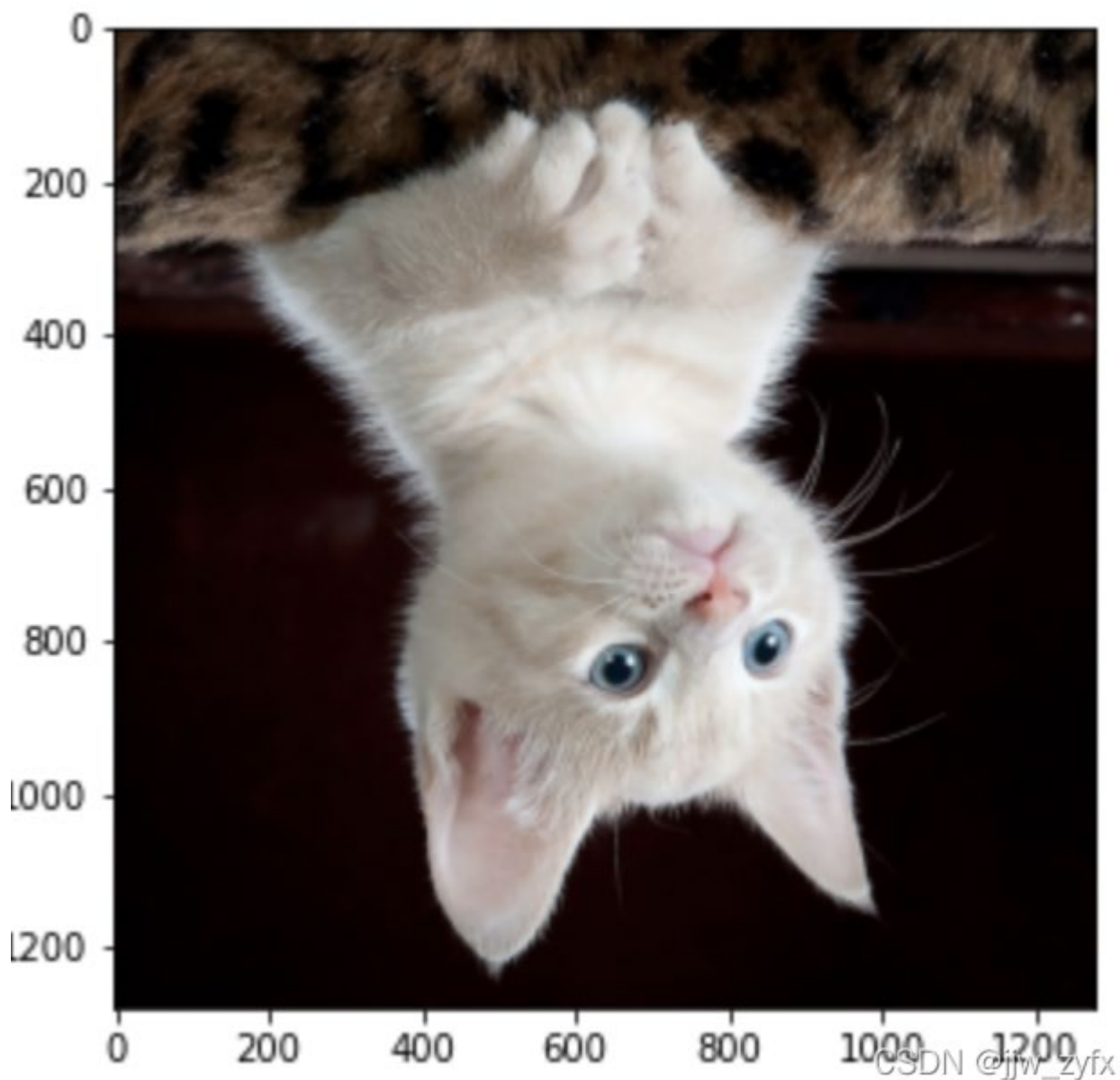
```
<matplotlib.image.AxesImage at 0x130c9cfd0>
```



OSDN @jjw\_zyfx

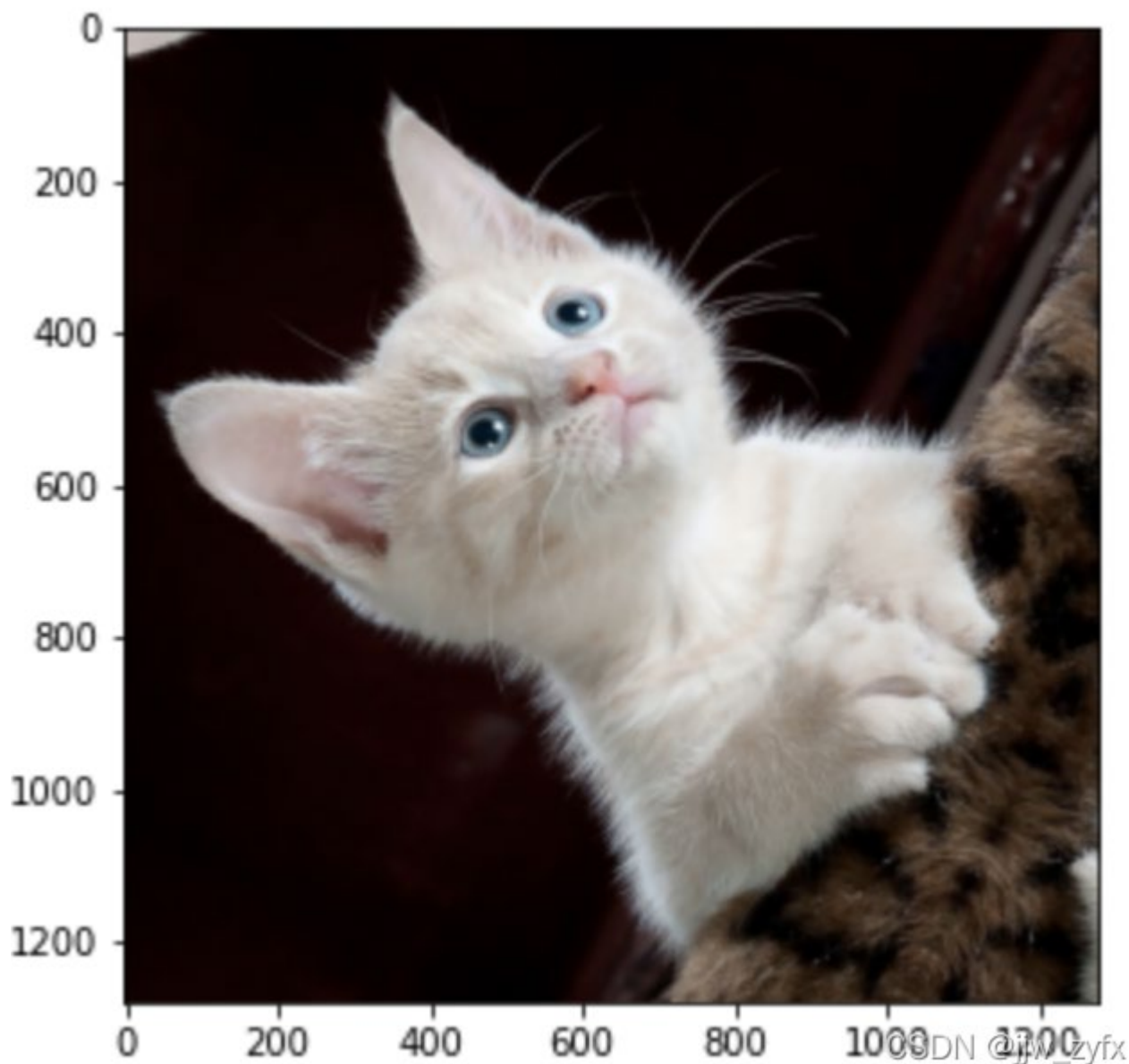
## 垂直翻转

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 # 垂直翻转
4 image = albumentations.VerticalFlip(p=1.0)(image=image)["image"]
5 plt.figure(figsize=(5, 5))
6 plt.imshow(image)
```



## 旋转

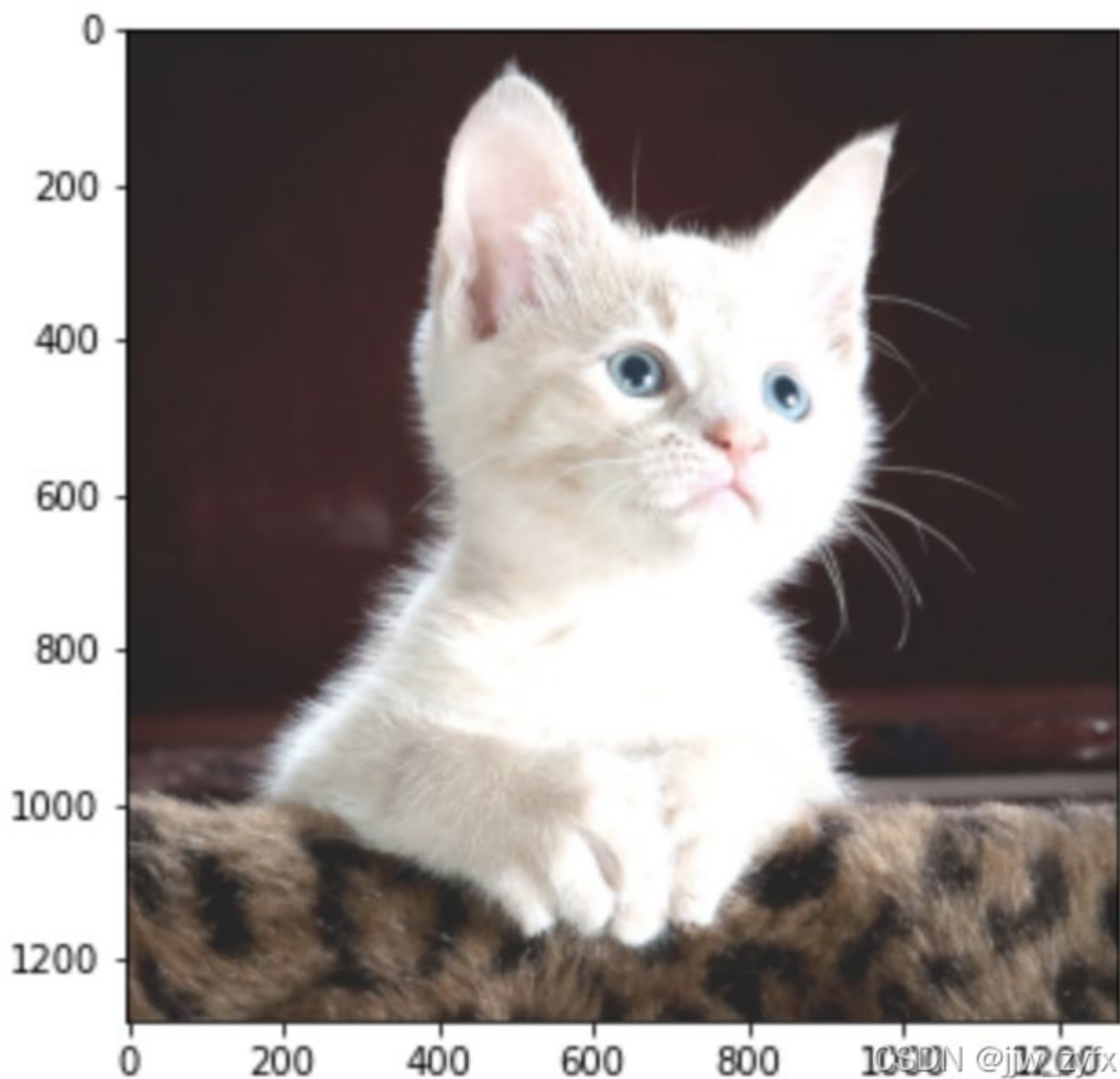
```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 image = albumentations.Rotate(limit=180, p=1)(image=image)["image"] # 在(-180, 180)之
4 间随机旋转
5 plt.figure(figsize=(5, 5))
  plt.imshow(image)
```



## RandomBrightnessContrast

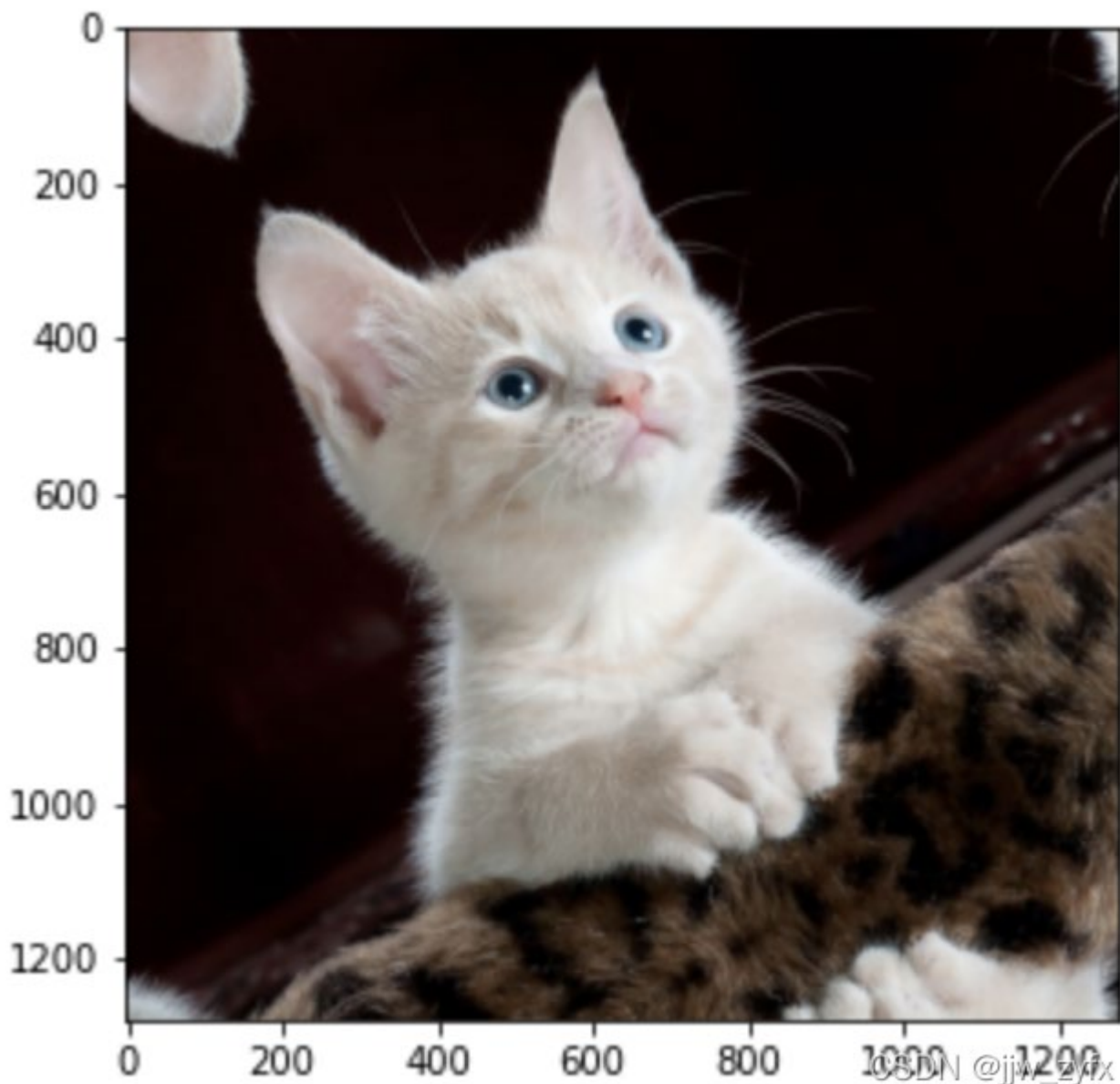
```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 image = albumentations.RandomBrightnessContrast(brightness_limit=0.8, contrast_limit=.
4 2, brightness_by_max=True, p=1)(image=image)["image"]
5 # RandomBrightnessContrast默认参数亮度和对比度都是0.2，将亮度调到0.8后发现猫明显变亮
6 plt.figure(figsize=(5, 5))
  plt.imshow(image)
```





## ShiftScaleRotate

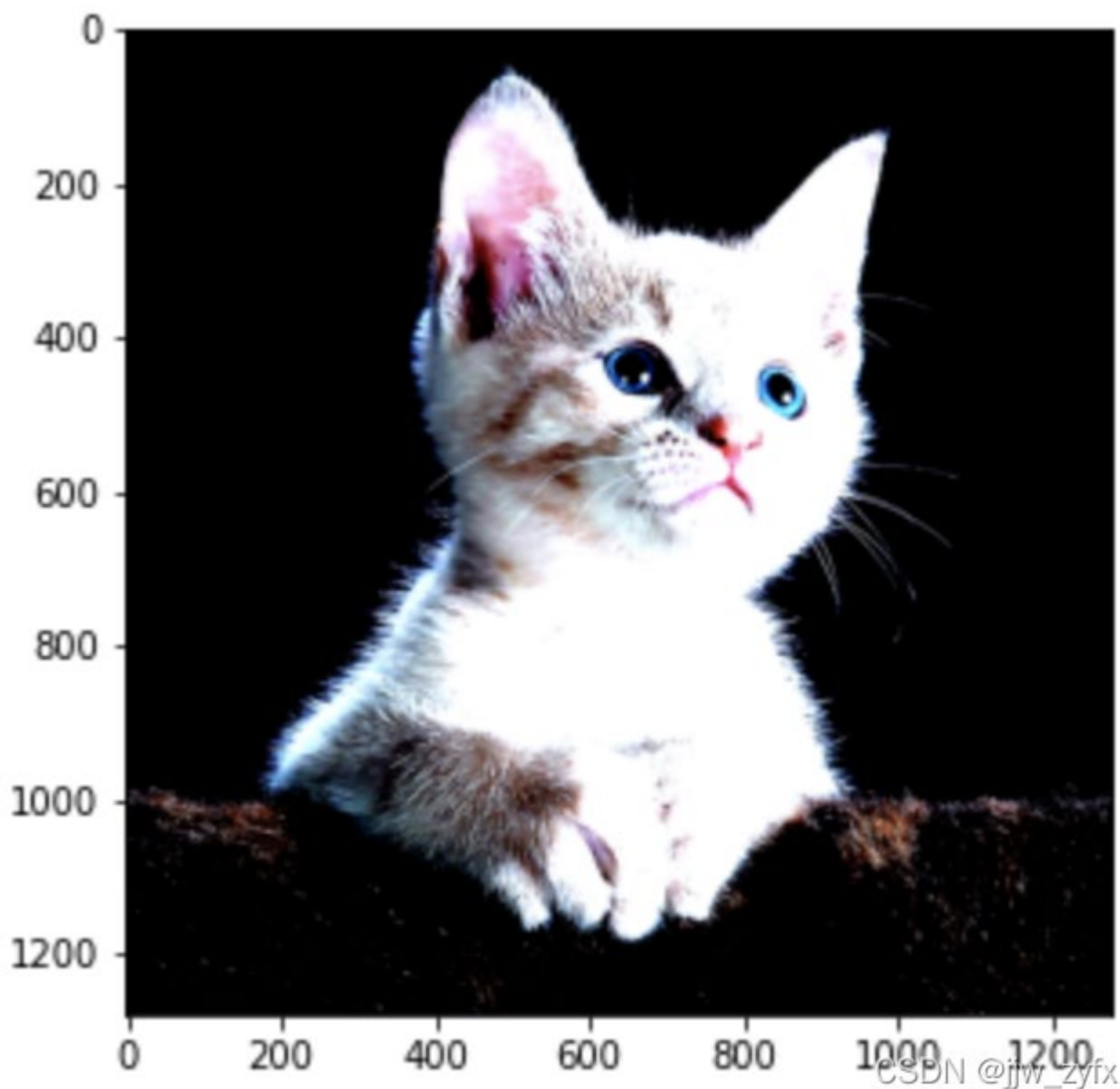
```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 # shift_limit 移动因子(我看着不像是平移倒像是镜像即以某个点为中心, 中心对称), 设置高和宽的移
4 动因子取值范围, 如果shift_limit是一个浮点数,
5 # 则移动因子的取值范围为(-shift_limit, shift_limit), shift_limit的取值范围的绝对值应在[0,1
6 ]之间, 默认值是(-0.0625, 0.0625).
7 # scale_limit 缩放因子范围, 这个倒像是进行缩放的,其取值范围为[0, 1] 默认为(-0.1, 0.1)
8 # rotate_limit 旋转角度 默认为(-45, 45)
9 image = albumentations.ShiftScaleRotate(
10     shift_limit=0.0625, scale_limit=0.1, rotate_limit=45, p=1.0 # 对图片进
11 行平移 (translate)、缩放 (scale) 和旋转 (rotate)
    )(image=image)["image"]
plt.figure(figsize=(5, 5))
plt.imshow(image)
```



## Normalize

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 # 这个就是标准化操作，显示出来的不是正常的图片是值缩放到[0,1]之间的图片，一般处理完图片后会
4 再缩放回去，
5 # 这里显示下就是为了体现进行了标准化操作
6 # [0.485, 0.456, 0.406] 均值, [0.229, 0.224, 0.225] 标准差 max_pixel_value最大像素值，
7 图片的最大像素值为255
8 image = albumentations.Normalize(
9     [0.485, 0.456, 0.406], [0.229, 0.224, 0.225],
10     max_pixel_value=255.0, always_apply=True
11 )(image=image)["image"]
plt.figure(figsize=(5, 5))
plt.imshow(image)
```





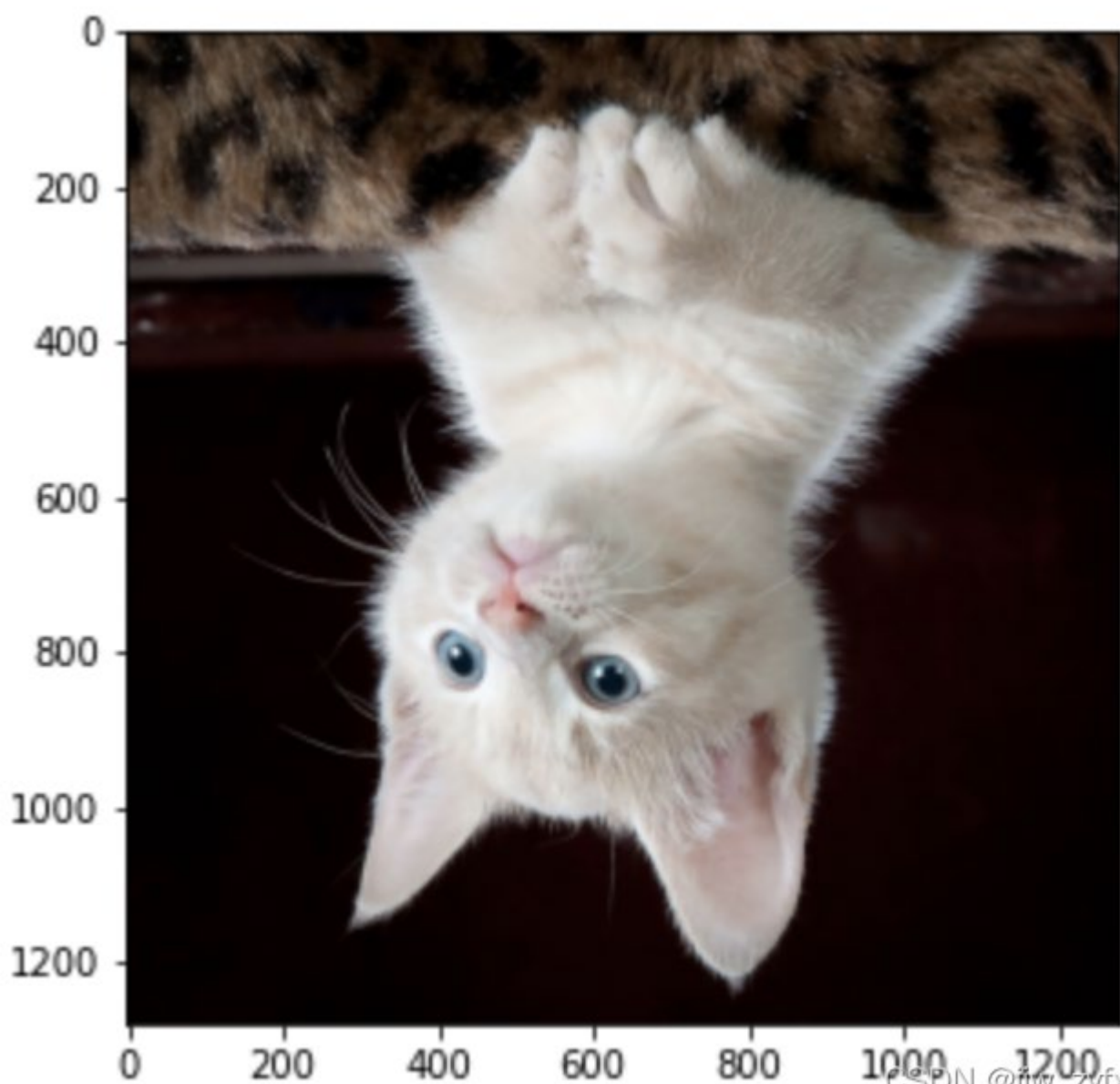
## ToTensorV2

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 print(type(image))
4 print(image.shape)
5 # 从这两个输出上可以看出来ToTensorV2的作用就是转换维度即将numpy的HWC转换成pytorch 的`CHW`.
6 image = ToTensorV2()(image=image)["image"]
7 print(image.shape)
8 # plt.figure(figsize=(5, 5)) # 转换完格式后是不能进行画图的，因为画图的格式就是numpy的HWC
9 格式
# plt.imshow(image)
```

```
<class 'numpy.ndarray'>
(1279, 1280, 3)
torch.Size([3, 1279, 1280])
```

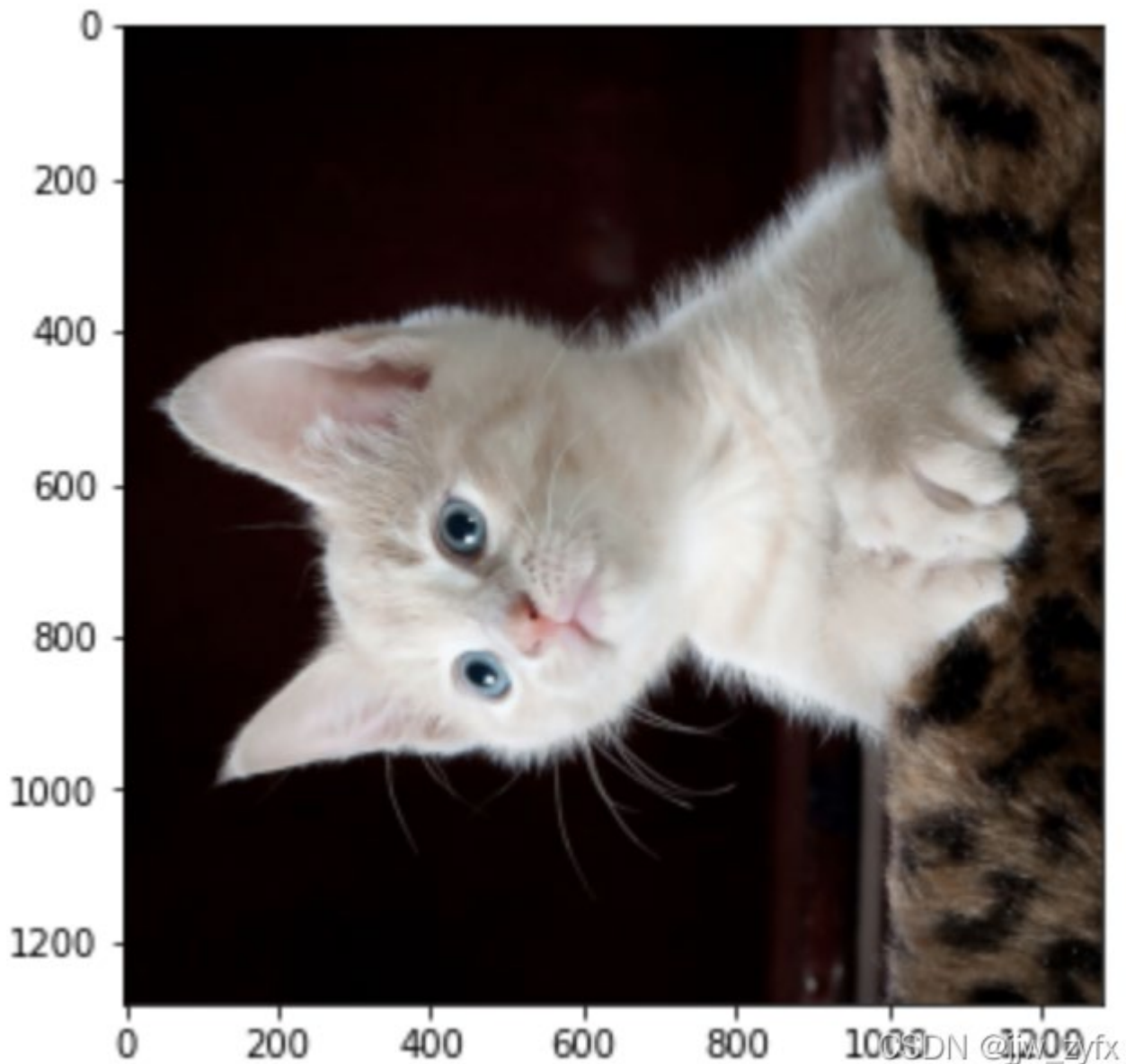
RandomRotate90

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 image = albumentations.RandomRotate90(p=1)(image=image)["image"] # 随机旋转90度
4 plt.figure(figsize=(5, 5))
5 plt.imshow(image)
```



## Transpose

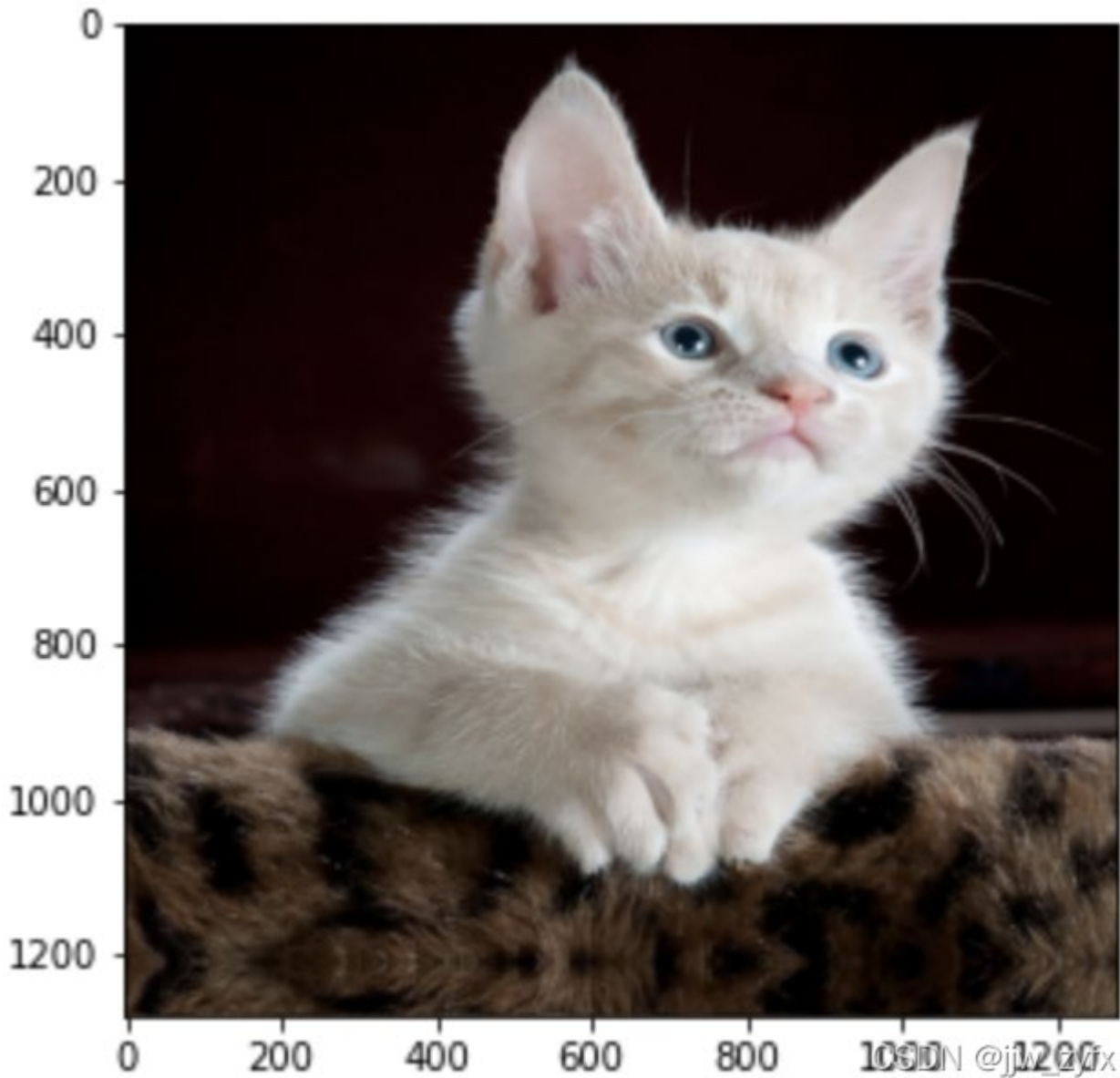
```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 image = albumentations.Transpose(p=1)(image=image)["image"] # 交换行和列 即转置
4 plt.figure(figsize=(5, 5))
5 plt.imshow(image)
```



## GridDistortion

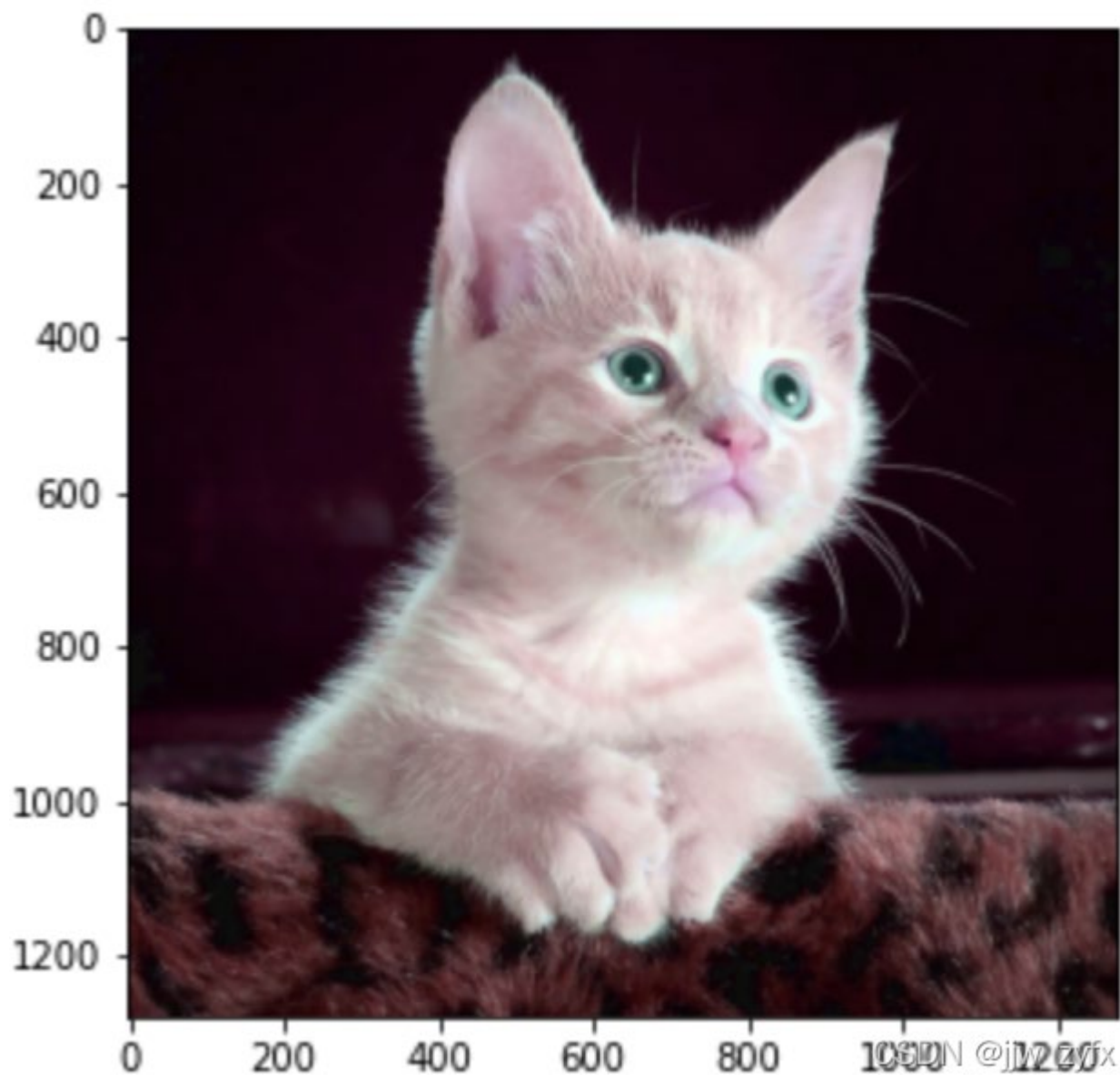
```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3
```

```
4 # 网格畸变 感觉能变瘦和变胖是随机的
5 image = albumentations.GridDistortion(p=1)(image=image)["image"]
6 plt.figure(figsize=(5, 5))
7 plt.imshow(image)
```



## HueSaturationValue

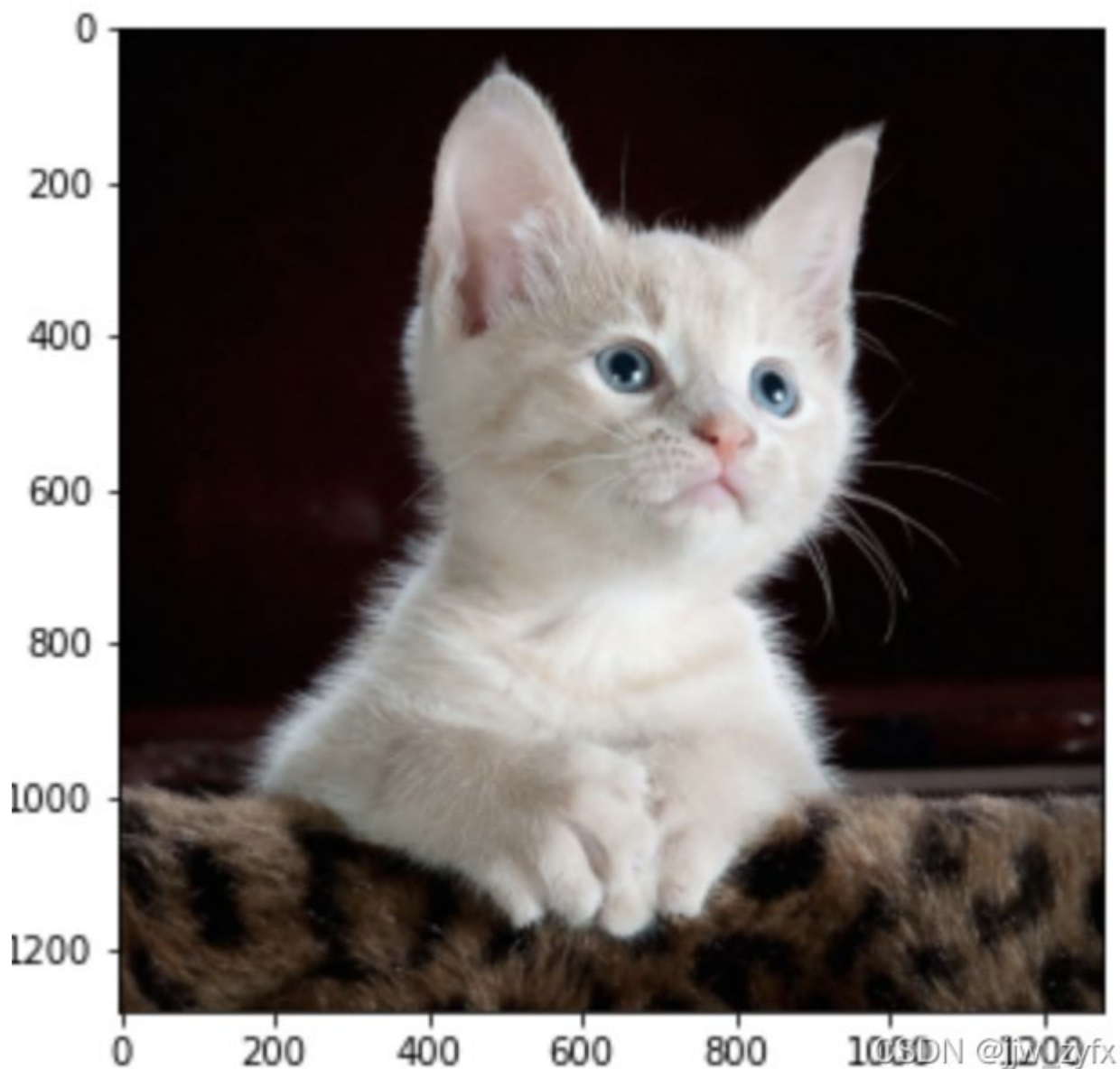
```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 # 随机改变图片的 HUE(色相)、饱和度和值
4 image = albumentations.HueSaturationValue(p=1)(image=image)["image"]
5 plt.figure(figsize=(5, 5))
6 plt.imshow(image)
```



## GaussNoise

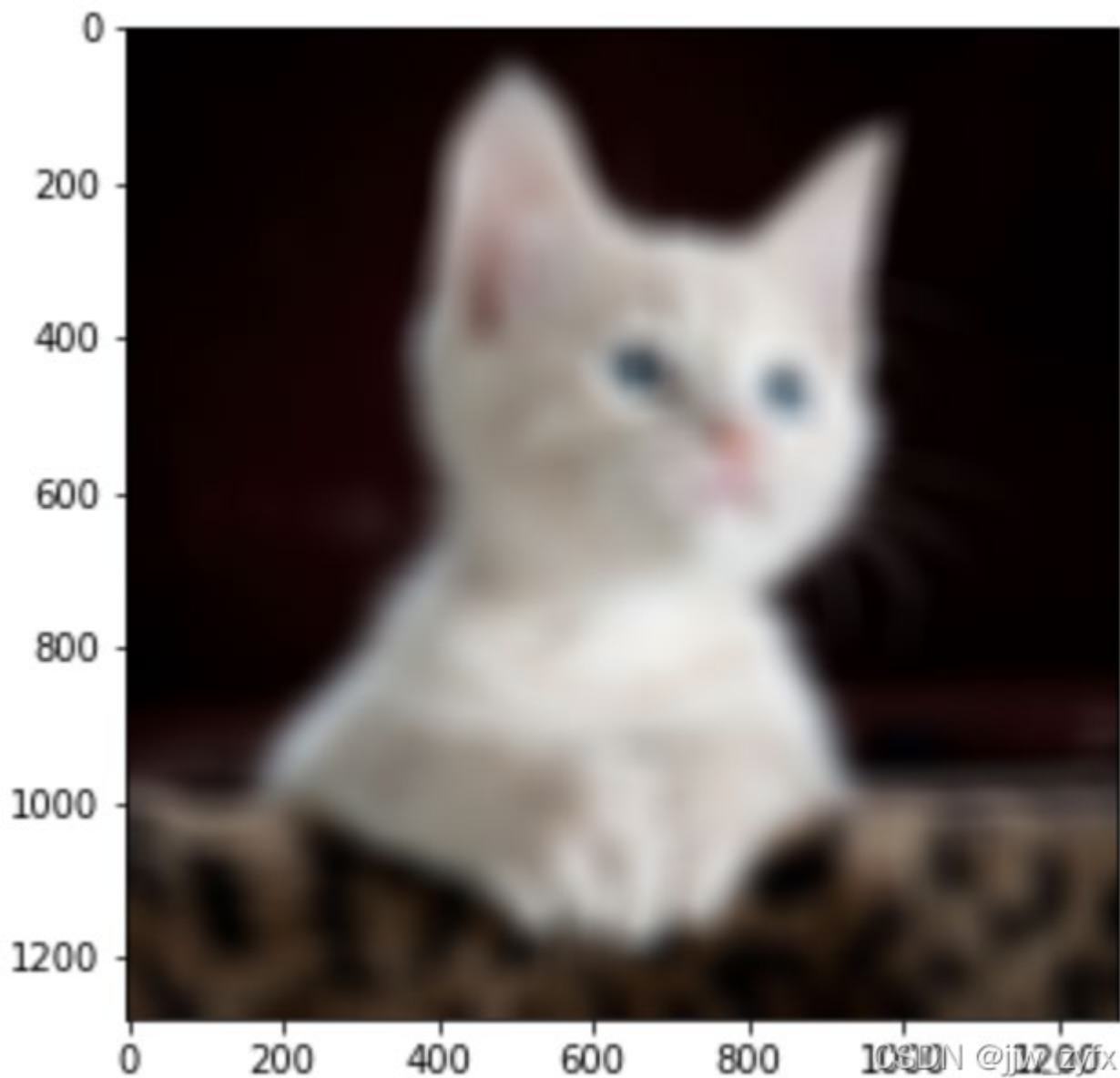
```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 # 高斯噪音 看不出来区别
4 image = albumentations.GaussNoise(var_limit=(10, 1000), p=1)(image=image)["image"]
5 plt.figure(figsize=(5, 5))
6 plt.imshow(image)
```





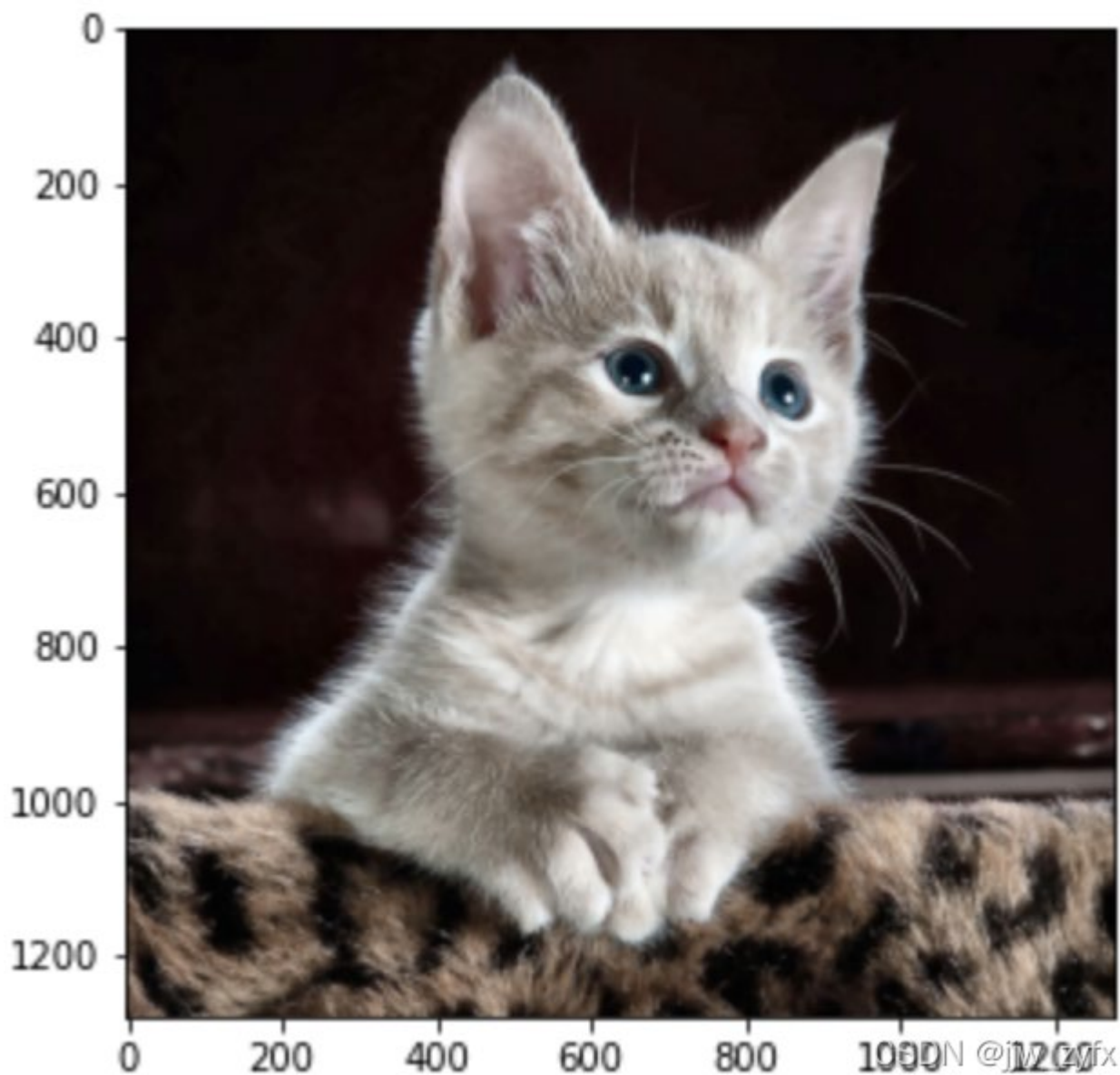
## Blur

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 # MotionBlur 这个模糊的轻些
4 # image = albumentations.MotionBlur(blur_limit=100, p=1)(image=image)["image"]
5 # MedianBlur 不支持 blur_limit 这个参数
6 # image = albumentations.MedianBlur(blur_limit=100, p=1)(image=image)["image"]
7 # 模糊图片, blur_limit的取值范围为: [3, inf), 默认为(3, 7)
8 image = albumentations.Blur(blur_limit=100, p=1)(image=image)["image"]
9 plt.figure(figsize=(5, 5))
10 plt.imshow(image)
```



## CLAHE

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 # 能看出来毛色变黑了些 百度翻译：对输入图像应用对比度受限自适应直方图均衡化
4 image = albumentations.CLAHE(clip_limit=2, p=1)(image=image)["image"]
5 plt.figure(figsize=(5, 5))
6 plt.imshow(image)
```

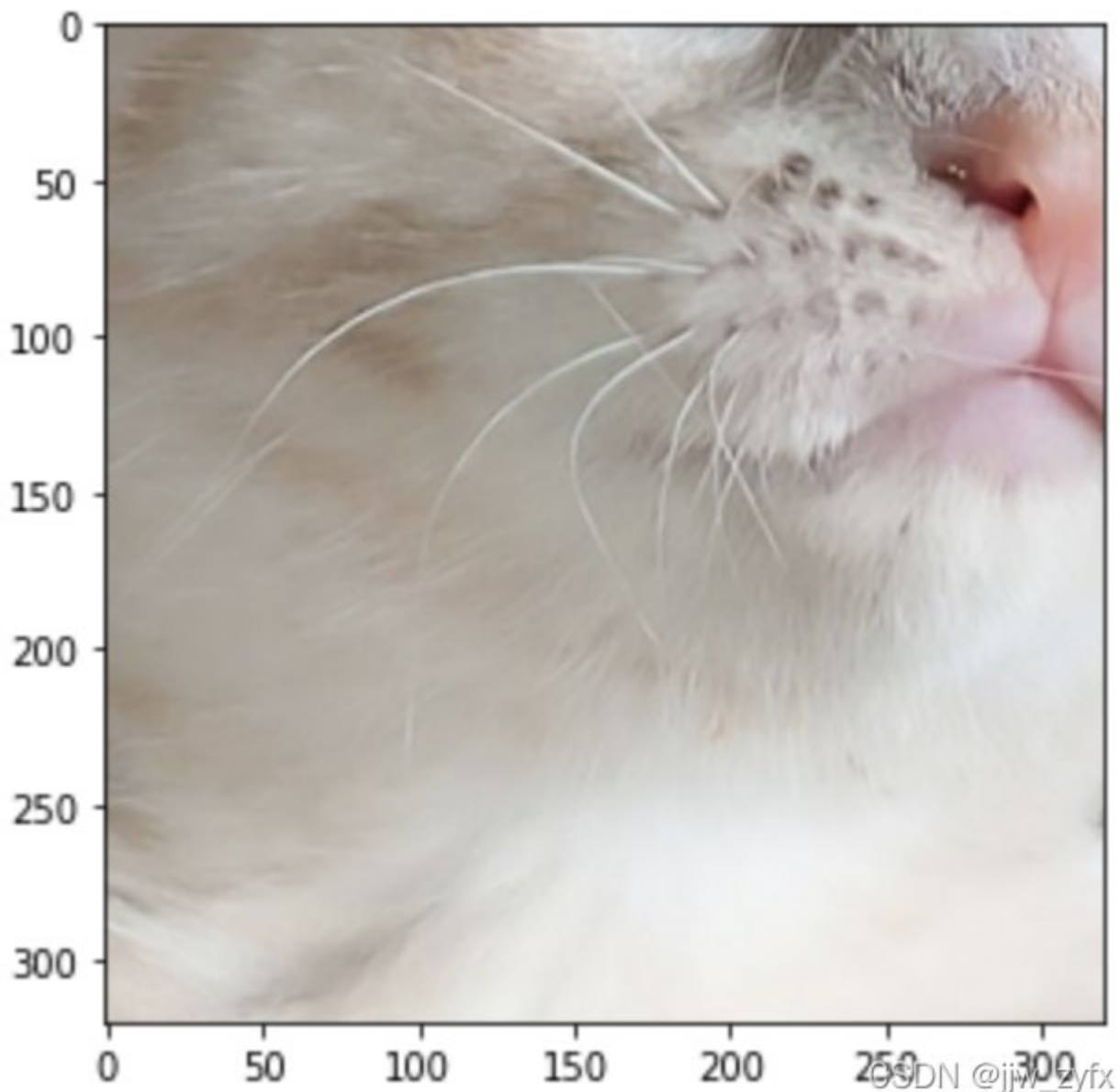


## CenterCrop

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 # 中心裁切, height:要裁切的高度, width: 要裁切的宽度
4 image = albumentations.CenterCrop(height=320, width=320, p=1)(image=image)["image"]
5 print(image.shape)
6 plt.figure(figsize=(5, 5))
7 plt.imshow(image)
```

(320, 320, 3)

<matplotlib.image.AxesImage at 0x132a3



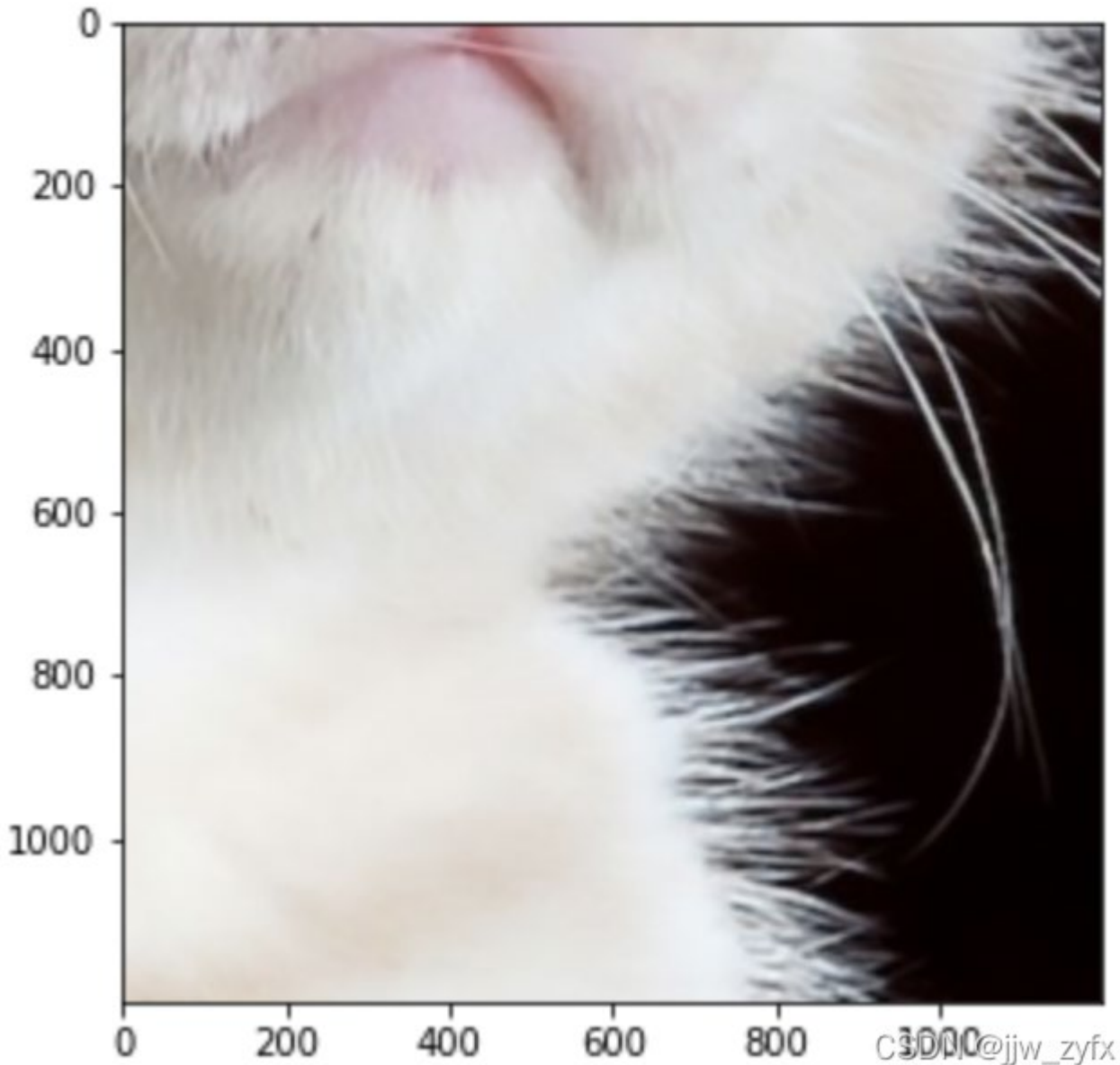
## RandomSizedCrop

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 print(image.shape)
4 # 随机大小裁切 裁切后的 height=1200, width=1200, min_max_height: 在(50, 1200)这个范围中
5 随机裁切一个尺寸, 裁切下来后再放大到height
6 image = albumentations.RandomSizedCrop(min_max_height=(50, 1200), height=1200, width=1
7 200, p=1)(image=image)["image"]
8 print(image.shape)
plt.figure(figsize=(5, 5))
```

```
plt.imshow(image)
```

```
(1279, 1280, 3)  
(1200, 1200, 3)
```

```
<matplotlib.image.AxesImage at 0x132a9d
```



Compose

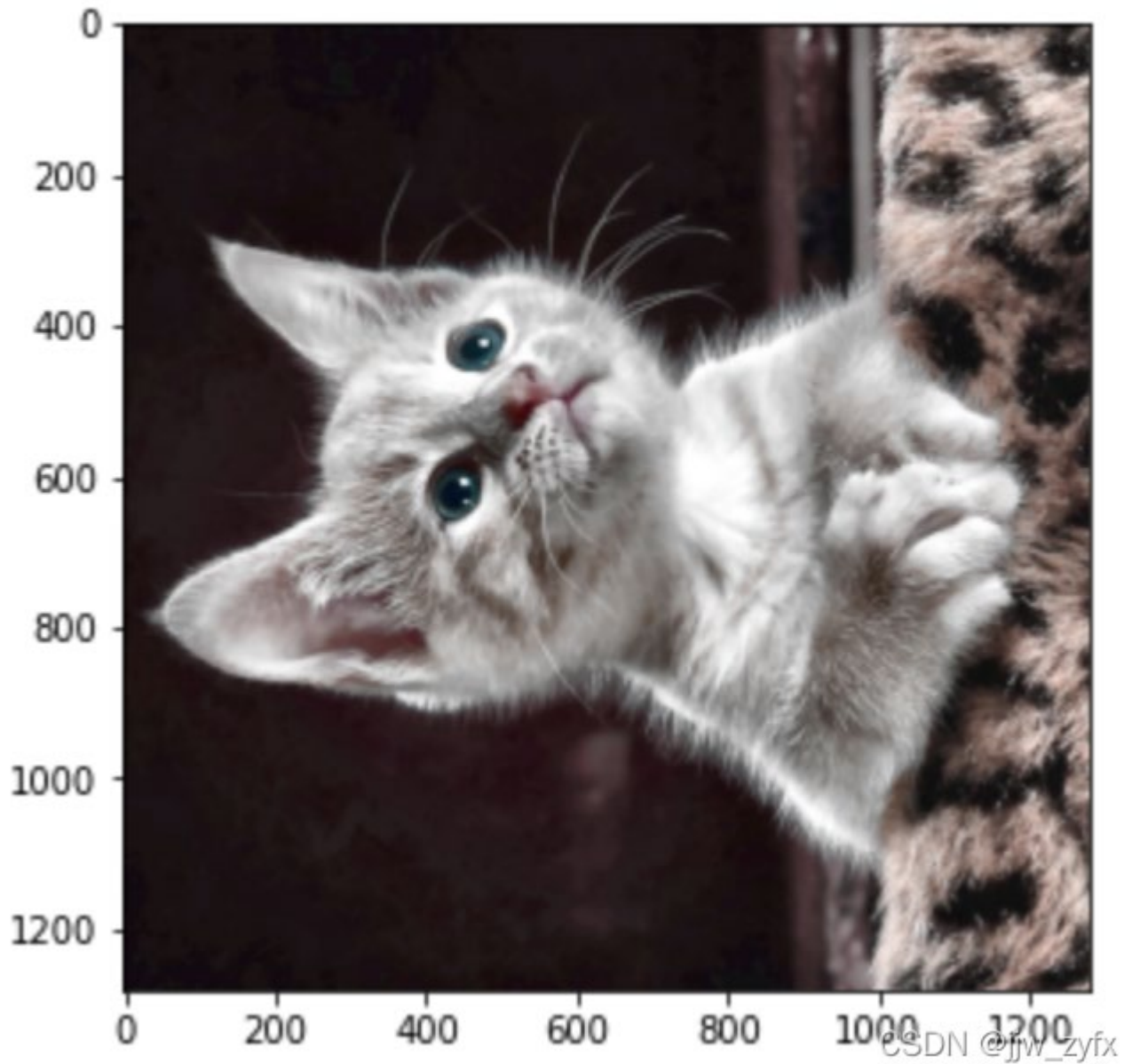
```
1 image = cv2.imread('./cat3.jpg')  
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)  
3 print(image.shape)  
4 组合操作
```



```
5  #
6  image = albumentations.Compose([
7      # 对比度受限直方图均衡
8          # (Contrast Limited Adaptive Histogram Equalization)
9      albumentations.CLAHE(),
10     # 随机旋转 90°
11     albumentations.RandomRotate90(),
12     # 转置
13     albumentations.Transpose(),
14     # 随机仿射变换
15     albumentations.ShiftScaleRotate(shift_limit=0.0625, scale_limit=0.50, rotate_l
16     imit=45, p=.75),
17     # 模糊
18     albumentations.Blur(blur_limit=3),
19     # 光学畸变
20     albumentations.OpticalDistortion(),
21     # 网格畸变
22     albumentations.GridDistortion(),
23     # 随机改变图片的 HUE、饱和度和值
24     albumentations.HueSaturationValue()
25     ], p=1.0)(image=image)['image']
26 print(image.shape)
plt.figure(figsize=(5, 5))
plt.imshow(image)
```

```
(1279, 1280, 3)
(1280, 1279, 3)
```

```
<matplotlib.image.AxesImage at 0x132b04
```



OneOf

```
1 image = cv2.imread('./cat3.jpg')
2 image = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
3 print(image.shape)
4 # 单一操作 即在这下边这几个中选择一个进行变换
5 image = albumentations.OneOf([
6     # 对比度受限直方图均衡
7     # (Contrast Limited Adaptive Histogram Equalization)
```

```
8     albumentations.CLAHE(),
9     # 随机旋转 90°
10    albumentations.RandomRotate90(),
11    # 转置
12    albumentations.Transpose(),
13    # 随机仿射变换
14    albumentations.ShiftScaleRotate(shift_limit=0.0625, scale_limit=0.50, rotate_l
15    imit=45, p=.75),
16    # 模糊
17    albumentations.Blur(blur_limit=3),
18    # 光学畸变
19    albumentations.OpticalDistortion(),
20    # 网格畸变
21    albumentations.GridDistortion(),
22    # 随机改变图片的 HUE、饱和度和值
23    albumentations.HueSaturationValue()
24    ], p=1.0)(image=image)['image']
25 print(image.shape)
26 plt.figure(figsize=(5, 5))
plt.imshow(image)
```

