

## Pytorch--transforms图像数据增强



巫婆

8 人赞同了该文章

总结介绍transforms中图像预处理方法。

### 一、裁剪——Crop

#### 1.随机裁剪：transforms.RandomCrop

```
class torchvision.transforms.RandomCrop(size, padding=None, pad_if_needed=False, fill=0, padding_mode='constant')
```

功能：依据给定的size随机裁剪

参数：

size- (sequence or int)，若为sequence,则为(h,w)，若为int，则(size,size)

padding-(sequence or int, optional)，此参数是设置填充多少个pixel。

当为int时，图像上下左右均填充int个，例如padding=4，则上下左右均填充4个pixel，若为3232，则会变成4040。

当为sequence时，若有2个数，则第一个数表示左右扩充多少，第二个数表示上下的。当有4个数时，则为左，上，右，下。

fill- (int or tuple) 填充的值是什么（仅当填充模式为constant时有用）。int时，各通道均填充该值，当长度为3的tuple时，表示RGB通道需要填充的值。

padding\_mode- 填充模式，这里提供了4种填充模式，1.constant，常量。2.edge 按照图片边缘的像素值来填充。3.reflect，暂不了解。4.symmetric，暂不了解。

#### 2.中心裁剪：transforms.CenterCrop

```
class torchvision.transforms.CenterCrop(size)
```

功能：依据给定的size从中心裁剪

参数：

size- (sequence or int)，若为sequence,则为(h,w)，若为int，则(size,size)

#### 3.随机长宽比裁剪：transforms.RandomResizedCrop

```
class torchvision.transforms.RandomResizedCrop(size, scale=(0.08, 1.0), ratio=(0.75, 1.3333333333333333), interpolation=2)
```

功能：随机大小，随机长宽比裁剪原始图片，最后将图片resize到设定好的size

参数：

size- 输出的分辨率

ratio- 随机长宽比设置

interpolation- 插值的方法，默认为双线性插值(PIL.Image.BILINEAR)

4.上下左右中心裁剪：transforms.FiveCrop

```
class torchvision.transforms.FiveCrop(size)
```

功能：对图片进行上下左右以及中心裁剪，获得5张图片，返回一个4D-tensor

参数：

size- (sequence or int)，若为sequence,则为(h,w)，若为int，则(size,size)

5上下左右裁剪后翻转：transforms.TenCrop

```
class torchvision.transforms.TenCrop(size, vertical_flip=False)
```

功能：对图片进行上下左右以及中心裁剪，然后全部翻转（水平或者垂直），获得10张图片，返回一个4D-tensor。

参数：

size- (sequence or int)，若为sequence,则为(h,w)，若为int，则(size,size)

vertical\_flip (bool) - 是否垂直翻转，默认为false，即默认为水平翻转

## 二、翻转和旋转——Flip and Rotation

6.依概率p水平翻转：transforms.RandomHorizontalFlip

```
class torchvision.transforms.RandomHorizontalFlip(p=0.5)
```

功能：依据概率p对PIL图片进行水平翻转

参数：

p- 概率，默认值为0.5

7.依概率P垂直翻转：transforms.RandomVerticalFlip

```
class torchvision.transforms.RandomVerticalFlip(p=0.5)
```

功能：依据概率p对PIL图片进行垂直翻转

参数：

p- 概率，默认值为0.5

8.随机旋转：transforms.RandomRotation

```
class torchvision.transforms.RandomRotation(degrees, resample=False, expand=False, center=None)
```

功能：依degrees随机旋转一定角度

若为sequence, 如(30, 60), 则表示在30-60度之间随机旋转

resample- 重采样方法选择, 可选 PIL.Image.NEAREST, PIL.Image.BILINEAR, PIL.Image.BICUBIC, 默认为最近邻

expand- ?

center- 可选为中心旋转还是左上角旋转

### 三、图像变换

#### 9.尺寸变换resize: transforms.Resize

```
class torchvision.transforms.Resize(size, interpolation=2)
```

功能: 重置图像分辨率

参数:

size- If size is an int, if height > width, then image will be rescaled to (size \* height / width, size), 所以建议size设定为h\*w

interpolation- 插值方法选择, 默认为PIL.Image.BILINEAR

#### 10.标准化: transforms.Normalize

```
class torchvision.transforms.Normalize(mean, std)
```

功能: 对数据按通道进行标准化, 即先减均值, 再除以标准差, 注意是 hwc

#### 11.转为tensor: transforms.ToTensor

```
class torchvision.transforms.ToTensor
```

功能: 将PIL Image或者 ndarray 转换为tensor, 并且归一化至[0-1]

注意: 归一化至[0-1]是直接除以255, 若自己的ndarray数据尺度有变化, 则需要自行修改。

#### 12.填充: transforms.Pad

```
class torchvision.transforms.Pad(padding, fill=0, padding_mode= 'constant' )
```

功能: 对图像进行填充

参数:

padding-(sequence or int, optional), 此参数是设置填充多少个pixel。

当为int时, 图像上下左右均填充int个, 例如padding=4, 则上下左右均填充4个pixel, 若为3232, 则会变成4040。

当为sequence时, 若有2个数, 则第一个数表示左右扩充多少, 第二个数表示上下的。当有4个数时, 则为左, 上, 右, 下。

fill- (int or tuple) 填充的值是什么 (仅当填充模式为constant时有用)。int时, 各通道均填充该

13.修改亮度、对比度和饱和度: transforms.ColorJitter

```
class torchvision.transforms.ColorJitter(brightness=0, contrast=0, saturation=0, hue=0)
```

功能: 修改修改亮度、对比度和饱和度

14.转为灰度图: Transforms.Grayscale

```
class torchvision.transforms.Grayscale(num_output_channels=1)
```

功能: 将图片转换为灰度图

参数:

num\_output\_channels- (int) , 当为1时, 正常的灰度图, 当为3时, 3 channel with r == g == b

15.线性变换: transforms.LinearTransformation

```
class torchvision.transforms.LinearTransformation(transformation_matrix)
```

功能: 对矩阵做线性变化, 可用于白化处理! whitening: zero-center the data, compute the data covariance matrix

参数:

transformation\_matrix (Tensor) – tensor [D x D], D = C x H x W

16.仿射变换: transforms.RandomAffine

```
class torchvision.transforms.RandomAffine(degrees, translate=None, scale=None, shear=None, resample=False, fillcolor=0)
```

功能: 仿射变换

17.依概率p转为灰度图: transforms.RandomGrayscale

```
class torchvision.transforms.RandomGrayscale(p=0.1)
```

功能: 依概率p将图片转换为灰度图, 若通道数为3, 则3 channel with r == g == b

18.将数据转换为PILImage: transforms.ToPILImage

```
class torchvision.transforms.ToPILImage(mode=None)
```

功能: 将tensor 或者 ndarray的数据转换为 PIL Image 类型数据

参数:

mode- 为None时, 为1通道, mode=3通道默认转换为RGB, 4通道默认转换为RGBA

19.transforms.Lambda

四、对transforms操作, 使数据增强更灵活

功能：从给定的一系列transforms中选一个进行操作， randomly picked from a list

21.transforms.RandomApply(transforms,p=0.5)

功能：给一个transform加上概率，以一定的概率执行该操作

22.transforms.RandomOrder

功能：将transforms中的操作顺序随机打乱

编辑于 2019-11-28 13:12

图像处理

深度学习 (Deep Learning)

PyTorch

文章被以下专栏收录



CV & NLP & KG

推荐阅读



pytorch图像数据增强7大技巧

机器学习入坑者

pytorch中归一化  
transforms.Normalize的真...

关于transforms.Normalize的真正理解 我们都知道，当图像数据输入时，需要对图像数据进行预处理，常用的预处理方法，本文不再赘述，本文重在讲讲 transform.ToTensor和...  
Transformer



图像分类：数据增强 (Pytorch版)

TeddyZhang

1 条评论

切换为时间排序

写下你的评论...



想变优秀的小煤球

2021-11-01

作者您好，我有个疑问，就是transform是用来增强数据的，而且有些操作都是随机的，那对图片进行了transform也需要对label进行同样的处理吗。而且不管要不要，进行处理之后的图片都没有对应的标签图啊，这还可以用来训练吗？（因为transform大部分操作都是随机的）  
我不是很懂

赞

