

学生实验报告

学号	1120192933	学院	计算机学院
姓名	李桐	专业	人工智能

商品识别

1 实验目的

- (1) 理解 pytorch 在预测阶段的运用。
- (2) 理解测试时增强 (test time augmentation, TTA) 的作用。

2 实验原理

- (1) pytorch 模型预测方法。
- (2) TTA 原理。
- (3) resnest 模型。

3 实验条件与环境

要求	名称	版本要求	备注
编程语言	python	3.6 以上	
开发环境	dsw	无要求	
第三方工具包/ 库/插件	opencv-python	4.5 以上	
第三方工具包/ 库/插件	tqdm	4.32	
第三方工具包/ 库/插件	pytorch	1.0 以上	
第三方工具包/	mmdetection	1.2	

库/插件			
其他工具	无	无要求	
硬件环境	台式机、笔记本均可	无要求	

4 实验步骤及操作

步骤序号	1
步骤名称	构建 resnest50
步骤描述	构建预测模型 resnest50。
代码及讲解	<p>神经网络中常用的特征提取网络是 ResNet。ResNet 的特殊结构解决了深层网络所带来的梯度等一系列问题，并且可以使网络能够搭建的更深。ResNet 带来的影响是巨大的，现在在采用深度学习的方式时，利用神经网络时，很大程度上都采用了 ResNet 进行特征提取。</p> <p>从 ResNet 提出后也出现了很多基于其的改进的特征提取网络，比较成功的是结合了 Inception 结构的 ResNeXt，ResNeXt 的本质是采用了分组卷积。而在 ResNest 中也采用了切片思想，其中又加入了 SENet 及 SKNet 注意力思想。</p> <p>ResNest 的 block 内部是先将输入进行 k 切分，然后每个 k 切分又进行 r 切分，这个 r 切分后每个模块称为 Split Attention。ResNest 的 split attention 形式就是借鉴了 SKNet 的形式，在注意力方面，采用了 SENet 的思想，结合 r 个切片，然后经过池化、激活等操作，计算出 r 切片中每一个切片的对应注意力参数。</p> <p>我一开始是在自己构建的，发现好慢，而且没有预训练模型，就更完蛋了，所以后来查询了一下官方实现：</p> <pre>def resnest50(pretrained=False, root='~/encoding/models', **kwargs): model = ResNet(Bottleneck, [3, 4, 6, 3], radix=2, groups=1, bottleneck_width=64, deep_stem=True, stem_width=32, avg_down=True, avd=True, avd_first=False, **kwargs) if pretrained: model.load_state_dict(torch.hub.load_state_dict_from_url(resnest_model_urls['resnest50'], progress=True, check_hash=True)) return model</pre> <p>后来更改后：</p> <pre>@RESNEST_MODELS_REGISTRY.register() def resnest50(pretrained=False, root='~/encoding/models', **kwargs): model = ResNet(Bottleneck, [3, 4, 6, 3], radix=2, groups=1, bottleneck_width=64, deep_stem=True, stem_width=32, avg_down=True, avd=True, avd_first=False, **kwargs) if pretrained: #model.load_state_dict(torch.hub.load_state_dict_from_url(# resnest_model_urls['resnest50'], progress=True, check_hash=True)) model.load_state_dict(torch.load('resnest50.pth')) return model</pre>

步骤序号	2
------	---

步骤名称	测试时增强
步骤描述	在预测时额外对数据进行随机水平翻转。
代码及讲解	<pre>#数据处理 if kind=='train': self.transform = transforms.Compose([transforms.Resize((224, 224)), transforms.RandomHorizontalFlip(p=0.5), transforms.ToTensor(), transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225])) # 归一化</pre>

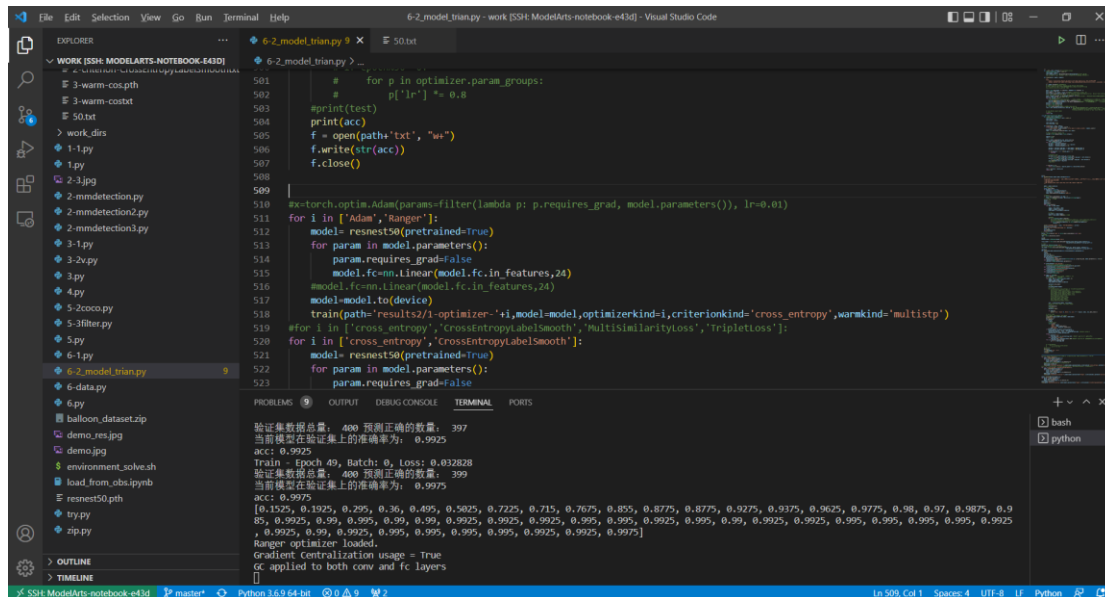
步骤序号	3
步骤名称	商品识别
步骤描述	对检测出来的候选框进行商品识别。
代码及讲解	<p>图片输入上述网络进行测试。</p> <pre>model= resnest50(pretrained=True) for param in model.parameters(): param.requires_grad=False model.fc=nn.Linear(model.fc.in_features,24) model=model.to(device) with torch.no_grad(): for data in data_loader: images, true_labels = data images = images.to(device) true_labels = true_labels.to(device) # 前向传播 pre_labels = model(images) total += pre_labels.shape[0]#总测试数 #判断正误 for i in range(pre_labels.shape[0]): #print(pre_labels[i],torch.argmax(pre_labels[i]) ,torch.argmax(true_labels[i])) #print(torch.argmax(pre_labels[i]) ,torch.argmax(true_labels[i])) if torch.argmax(pre_labels[i]) == torch.argmax(true_labels[i]): correct += 1</pre>

5 实验结果及分析讨论

(1) 最终结果的具体结果（文字说明）

一开始识别的结果较差，准确率大概为 22%。后来发现了问题，结果还不错。

(2) 最终结果界面截图（界面截图）



上边这个图是没找到问题之前跑的一次，效果只有 15%，后来找到问题了能到 100%。

(4) 最终结果的解读与讨论

一开始在实验的时候在图片数据集上训练，在视频的截图数据集上测试。但是效果比较

后来找到了问题！

```
for p in optimizer.param_groups:
    p['lr'] *= learning_rate
```

6 收获与体会

在构建 ResNest50 时，遇到了很多问题：首先是这样一个问题，这是我拿了一部分官方 ResNeSt 给的代码一部分自己写得代码写出来的问题。

```
(PyTorch-1.8) [ma-user work]$python 6.py
Traceback (most recent call last):
  File "6.py", line 218, in <module>
    net = resnest50(pretrained=True)
  File "6.py", line 180, in resnest50
    avd=True, avd_first=False, **kwargs)
TypeError: __init__() got an unexpected keyword argument 'radix'
```

后来尽量使用官方代码。遇到的问题就是在华为云下载的预训练模型，下次启动就关了，很烦。然后更改了官方代码库的逻辑，并且作为一个文件下载到了 work 文件夹下面，作为一个自定义的代码库使用。

后来训练的时候用了好久，发现一直等准确率都是 100%，怎么也找不到问题。结果发现标签错了，用 test 内个标签标的，全标了 1，实际上应该用 train 内个标标签。

罪魁祸首就是这几步：

```
# with open('./data3/i_train.json','r') as f:
#     res_i=json.load(f)

# # for item in res_i['annotations']:
# #     item['category_id']=1
# #print(res_i)
# #res_i['categories']=res_i['category_id'][:1]

# print(res_i['categories'])

# with open('./data3/i_train.json','w') as f:
#     json.dump(res_i,f)

# with open('./data3/v_train.json', 'r') as f:
#     res_i=json.load(f)

# for item in res_i['annotations']:
#     item['category_id'] = 1
```

7 备注及其他

无。