模式识别与机器学习 **30250293-0 (2022 春)**

模式识别与机器学习大作业： 衣物颜色匹配

1) 请**在网络学堂提交电子版**

2) 每位同学须**独立**完成大作业

3) 请在 **6 月 26 日 23:59:59*** 前提交大作业，**不接受补交**

4) 如有疑问请微信或邮件联系助教

吴文绪: wuwx21@mails.tsinghua.edu.cn

李嘉琦: lijq19@mails.tsinghua.edu.cn

颜钱明: yanqm18@mails.tsinghua.edu.cn

蔡昊晓: chx21@mails.tsinghua.edu.cn

5) 带 * 部分为草拟部分，可能根据同学们的反馈以及其他情况进行相应的修改

目录

1 赛题描述

1.1 背景介绍

图 1: 示例-某店铺的各色款式的衣服

在各大电商平台上，衣物一般都是流通量较大的商品类目。一款衣物，可能根据其尺码、颜色、图案等等特征分为更细的子款式。在这次作业中，我们主要关注“颜色”所带来的款式区分。

对于一个服装店铺来说，一次上新可能需要耗费很多人力，用于甄别上传一个商品不同款式的实拍图片。既然分批上传图片费时费力，而“颜色”似乎又是一个比较好识别区分的特征，自然会想到利用一些识别手段，来辅助颜色的区分。以主流电商平台淘宝为例，其给出的方案是：将颜色空间划分为 16 个标准类别，并对所有上传图像分类。

但卖家并不一定采用淘宝的辅助识别结果，打开一款商品，一般你可以看到商家给出的颜色款式的标签，并不总是“xx 色”这样的淘宝定义的标准标签，可能是一些更无约束的，描述颜色的文字, 在选择页面传达更多信息，如图2所示。

图 2: 标签词云

在本次作业中，我们需要处理的是一个更为折衷的颜色识别和区分问题——若商家上传一个衣服所有款式的图片，和可供选择的一些描述颜色的文字标签，你需要设计一套算法，将这些图片和这些文字标签进行匹配。

1.2 问题形式化

- **输入**：一个商品的所有图片：$\mathcal{I} = \{I_1, I_2, \ldots, I_n\}$ 以及可选的描述颜色的字符标签 $\mathcal{T} = \{T_1, \ldots, T_m\}$
- **输出**：图片和文字标签的匹配关系，即需要输出 $(I_1, T_{I_1}), (I_2, T_{I_2}), \ldots, (I_n, T_{I_n})$。自然地，每张图的匹配标签需要在可选颜色字符标签集中做选取，即需要满足约束 $\forall i \in \{1, 2, \ldots, n\}, I_i \in \mathcal{I}, T_{I_i} \in \mathcal{T}$

1.3 数据细节

本次作业的数据集由淘宝网上随机爬取的约 2.7 万个商品 (总计约 20 万张图片) 组成。每个商品有大于等于 2 种的不同颜色款式。随机划分出约 5 千个商品作为测试集，余下数据进行发放。可供下载的版本有全尺寸、中等尺寸 (短边放缩为 224 像素)、小尺寸 (短边缩放为 50 像素) 的版本[1]。发放的数据集结构如下：

数据集文件夹结构

```
./
├── 543437665996            # 商品id
│   ├── 543437665996_0.jpg # 图片
│   ├── 543437665996_1.jpg
│   ├── 543437665996_2.jpg
│   ├── ...
│   ├── 543437665996_6.jpg
│   └── profile.json        # 可选颜色标签以及标签图片的匹配关系
├── 543447755424
│   ├── 543447755424_0.jpg
│   ├── 543447755424_1.jpg
│   ├── ...
│   ├── 543447755424_6.jpg
│   └── profile.json
├── 543470439046
├── ...
├── train_all.json          # 训练集各商品profile.json的合并
└── test_all.json           # 测试集各商品profile.json的合并
```

[1]下载链接：https://cloud.tsinghua.edu.cn/d/27849370d8774de3a2e2/ 其中全尺寸 (前缀 full) 数据集为分 4 卷压缩，中等尺寸 (前缀 medium) 和小尺寸 (前缀 thumbnail) 数据集各有一个压缩包。下载链接中还有这些压缩文件的 md5 可供下载后校验。解压后约使用空间：全尺寸 15G，中等尺寸 5G，小尺寸 1G。

profile.json 记录可选标签和匹配关系的格式

```
{
    "optional_tags": [  # 可选标签
        "绛红",
        "米色"
    ],
    "imgs_tags": [      # 各图与标签的匹配关系
                        # 测试集内此列表内的字典取值为{图片文件名: null}
        {
            "543437665996_0.jpg": "绛红"
        },
        {
            "543437665996_1.jpg": "绛红"
        },
        ...,
        {
            "543437665996_6.jpg": "米色"
        }
    ]
}
```

此外，在本次作业中我们还做一定的约束和简化：

1. 所给的可选标签的个数 m 一定小于等于图片数 n。
2. 在 ground truth 中，任意一个可选标签下一定存在至少一张匹配的图片

2 作业安排与要求 *

本次作业满分 100 分，由代码报告、模型性能两部分组成。

2.1 代码报告 (60 分)

本部分主要考虑代码的正确性以及报告的规范性和完整性。

代码部分：需要提交完整的训练测试代码。**代码中要求用相对路径以方便复现**。另外需要附一个 README.md 说明：

1. 简述代码部分各部分文件作用
2. 说明代码运行环境和库的版本
3. 数据集在项目目录的放置位置
4. 提供复现你的最佳模型的训练过程命令以及对应地在测试集上测试的命令。按此命令训练出的模型，不应当与你评测网站上评测结果差距过大。

完成作业过程中可以参考已有的代码，但要在代码和报告中相应部分给出引用说明。否则会影响代码查重结果和本部分得分。

报告部分：报告部分需要展示你对本问题的分析和解决方案。报告建议包括如下章节：问题分析与形式化、数据处理流程、算法原理、实现过程、实验结果、结果分析等。报告部分同样需要查重。

提交要求：本部分内容以 zip 压缩包的形式提交到网络学堂。提交的压缩包的命名格式要求为：**“班号-名字-学号.zip”**。压缩包内的目录和文件均采用英文命名防止因操作系统不同造成乱码；另外压缩包内不应该包含数据集，如果你对数据集做了静态的扩展或修改，请上传到清华云盘并在报告中附下载链接。最后**压缩包内结构的要求为：**

提交的作业目录结构要求

```
./
├── codes                    # 项目目录
│   ├── README.md            # 代码运行说明
│   ├── ...                  # 代码，可包含多个文件、目录
└── report.pdf               # 作业报告
```

2.2 模型性能 (40 分)

本部分分数以算法方案在预先划分的测试集上的性能表现综合给出。所包含的指标有

1. 图片文字匹配准确率：$\text{Acc} = \frac{\#\text{correct image-text pairs}}{\#\text{images}}$
2. 商品全匹配率: $\text{EM} = \frac{\#\text{products all matched}}{\#\text{products}}$

2.2.1 绝对性能 (20 分)

图片文字匹配准确率达到不同水平时分别得到如下分数

Acc	$0.3 \sim 0.7$	$0.7 \sim 0.75$	≥ 0.75
分数	$0 + \frac{8-0}{0.7-0.3} \times (\text{Acc} - 0.3)$	$8 + \frac{10-8}{0.75-0.7} \times (\text{Acc} - 0.7)$	10

商品全匹配率达到不同水平时分别得到如下分数

EM	$0.3 \sim 0.5$	$0.5 \sim 0.6$	≥ 0.75
分数	$0 + \frac{6-0}{0.5-0.3} \times (\text{EM} - 0.3)$	$6 + \frac{10-6}{0.6-0.5} \times (\text{EM} - 0.5)$	10

2.2.2 性能排名 (20 分)

我们会开放一个评测网站用于上传你的预测结果和算法。在每天中你会得到两次上传机会。其中开放评测第一周和法定节假日时获得的机会没有使用期限，其余情况下获得的上传机会的使用期限为当天。提交的**预测结果的格式参考 train_all.json 或 test_all.json**。

我们会对两个指标分别维护一个排名榜。你的**综合排名**为 Acc 和 EM 两个榜中的最高排名。最终，**本部分分数将由综合排名线性给出**，即得分为 $20 \times (1 - \text{排名百分位数})$

3 参考方案

抽象地来说，本次任务是一个较为粗粒度的图像和文本两模态数据匹配的任务。所以按照两个模态数据的处理方法是基于规则还是基于模型，可以得到不同方案。

3.1 用规则处理文字标签，用模型处理图片的方案

类比淘宝的标准方案，首先定义一个标准颜色词库。设定一定的文字标签和标准色库的匹配方案，利用标准色库将文字标签转为可以标准的“数字标签”。如下示例的文字标签和标准色库的匹配方案就是“标准色库的字词在文件标签中，即认为匹配”

```
def tag_to_label(tag):
    for idx, standard_word in enumerate(STANDARD_LEXICON):
        if standard_word in tag:
            return idx
    return len(STANDARD_LEXICON) # 未匹配中，所有未登录词处理为一个标签

labels = [tag_to_label(tag) for tag in tag_list]
```

转为标签后即可按照常规的分类任务，训练一个图像分类器。训练好图像分类器后，将算法部署到推理环节中，还需要思考怎么将分类器输出转为所需的“图文匹配关系”。

```
label_to_tag = {label:tag for tag in tag_list}
pred = torch.argmax(output, dim=-1)# 输出概率的argmax，作为预测标签
result = [(i, label_to_tag[label]) for i, label in enumerate(pred)]
```

这套方案里，你可能需要考虑的问题还有：

1. 文本匹配规则的构建。如以上的简单匹配方法中，假设标准色库只有“红”，在可选标签是“红色、粉红色”之类的情况下，分类会完全失效。另一方面，可能存在一些不能匹配上你设定的标准色库的文字标签，需要处理。
2. 文本匹配规则和类别数的平衡。极端地来说，你可以将每个一文字标签单独对应一个数字标签，但显然这样得到的类别数会偏多，难以训练。
3. 类别不平衡问题。从词云分布不难看出，做如上的转换后，我们得到的类别间的样本数是不平衡的，你可能需要在训练模型时加以考虑。

4. 不同商品间，相似标签在描述不同现象。想象这样的两种有“蓝色、黑色”两个标签的商品。一种是纯蓝/黑色的短袖，另一种是白底有蓝色/黑色条纹的短袖。对于后者，很有可能是白色的类别预测概率最高，而蓝色/黑色预测概率都被放缩到较小且难以比较数量上。

3.2 文字标签和图片均用模型处理的方案

使用模型分别提取颜色字符标签和图片的特征向量，并做特征向量之间的匹配

```
img_features = [model_img(img) for img in img_list]
tag_features = [model_text(tag) for tag in tag_list]
match_scores = match(img_features, tag_features)
# 处理match_scores变为最终结果
```

这套方案里，你可能需要考虑的问题还有：

1. 文字标签的预处理。
2. 两模态特征数值大小差异，可能需要一定的归一化处理
3. 两模态特征提取网络训练速度的差异。可能需要分别调节两模态特征提取网络的学习率等优化参数
4. 两模态特征向量匹配的方案。如计算特征向量之间的内积作为相似度，又或把图文特征向量一并输入一个“匹配模型”做二分类，又或者先使用图片特征向量做聚为“可选标签个数”个类别，再将聚类类别和标签做一一地匹配。

4 FAQ*

1. Q: 能否使用模型库中的模型，可否使用对应的预训练权重？

   A: 都可以。此外建议在报告中分析你使用模型的特点。

2. Q: 可否自行增删数据集

   A: 可以。可以仿照助教爬取数据的流程或自己想办法 (如自己下载合作标注)。若采用了这类静态的扩展数据集的方法，需要在报告中分析扩增前后模型性能的变化，并标注出其他人的贡献 (如果你采用合作标注等方案)。但若你没有相关经验，不建议在本次任务中尝试，因为 1. 所给训练集有约 2 万款商品，15 万张图片，在助教预测试的几种方案中数据量都是饱和的 2. 淘宝的反爬虫机制很强，初学者不容易处理。