

算法报告

1、团队介绍

团队名称：兄弟们别冲了准备投 SN

团队成员：王荣胜

成员邮箱：603329354@qq.com

指导老师：芦碧波

老师邮箱：27453714@qq.com

2、题目认知

2.1 题目描述

如今日新月异的计算机视觉技术可以对图片当中的实体物进行很有效的区分(比如经典的区分猫狗)，而天气却是一个看得见，感受得到，但是缺乏一个具体的实体的存在，本赛题希望通过对于标注了天气的图片进行分析、特征提取与建模，从而对于一张新的图片能够系统性地判断其中的天气。

2.2 数据概述

本次比赛将提供总共 252362 张图片，其中包含晴天图片 4000 张，阴天图片 4000 张。图片的大小不统一，但是格式统一为 jpg 格式。所有的训练集图片均有标注，标注文件可参考对应名称的 json 文件

2.2.1 训练集

训练集中包含 179584 张训练集图片和一份标注文件，标注文件的格式如下：

```
{ '1.jpg': 'sunny' } ##代表名称为 1.jpg 的图片是晴天图片
```

2.2.2 测试集

测试集中包含 72778 张图片，不包含标注文件。

2.3 提交格式

字段名称	格式类型	说明	示例
id	str	图片 id	1.jpg
weather	int	该图片是晴天还是阴天(0 为晴天 1 为阴天 2 为其他)	1

由此，我们将我们的任务确定为：图像多（三）分类任务。

2.4 题目分析

通过观察题目我发现，本次比赛决赛题目是经典的图像分类问题，对于这类问题最有效的建模方法是深度学习方法，在这一刻我便确定通过使用深度学习方法对问题进行建模。

本次数据集十分常见却又很特别。天气图像相较于猫狗花卉图像而言，并没有一个非常确定的物体来对它进行判断。我通常是通过多维度对天气进行感知，如温湿度，明暗度，云量，阳光，动物行为等。本次比赛中我只能通过视觉上的图片来对天气晴朗或多云做判断。则图像的明暗度，图像上方的云彩或干净的蓝色天空，图像下方的黄色阳光或积攒的雨水等都是图像的特征，需要在图像预处理时作为重要的参考依据。同时天气图像中出现了生活中的各种物体，与 ImageNet 数据集十分相似，在选择预训练模型时是重要参考依据。

3、方案概览

3.1 数据统计：

- 训练集：

```
-----  
169584 {'sunny': 56207, 'cloudy': 38614, 'others': 74763}  
6000 {'sunny': 3000, 'cloudy': 3000, 'others': 0}  
2000 {'sunny': 1000, 'cloudy': 1000, 'others': 0}  
2000 {'sunny': 1000, 'cloudy': 1000, 'others': 0}  
total: 179584 {'sunny': 61207, 'cloudy': 43614, 'others': 74763}  
-----
```

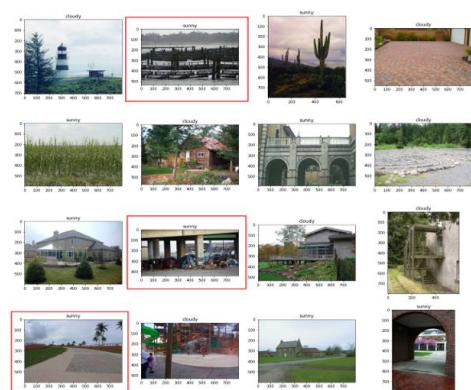
- 测试集：

```
-----  
test: 72778  
-----
```

3.2 数据预览：



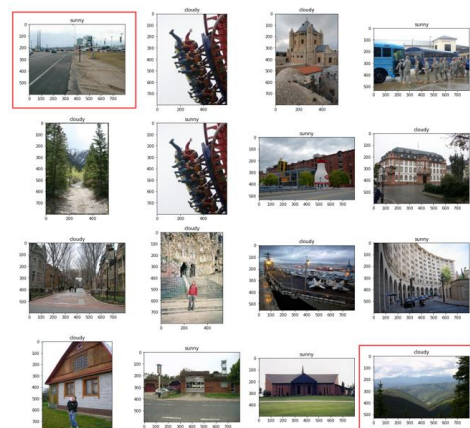
Part1



Part2



Part3



Part4

通过数据集的大小统计和查看，我们可以发现：

- (1) 训练集的 Part1 中有标签的数据为 sunny 和 cloudy(共 94821 张), 无标签数据为 others (共 74763 张);
- (2) 数据集 Part2、Part3、Part4 数据无 others 标签;
- (3) 不管是 Part1 还是 Part2、Part3、Part4 数据都存在错误标签的数据;
- (4) 对于验证集的分割，可以采用混合 Part1、Part2、Part3、Part4 四部分数据，然后按比例分割;

3.3 图像预处理：

- (1) 图像增强：图像随机左右翻转和 cutout

考虑到图像的亮暗是天气图像的一个重要特征，所以我很谨慎地放弃使用随机亮度，对比度和饱和度的图像增强方法。同时考虑到云彩，蓝天，照射到地上的阳光或积水是区分天气的重要特征，且位置较为固定，一般云彩蓝天在上，阳光积水在下，所以我放弃使用上下

翻转，只使用了左右翻转的图像增强方法。由于数据的特殊性，图片中不具有如猫狗一般十分明显的物体作为区分依据，则裁剪的物体很可能与判断天气无关，如下图示例中的红色墙壁，非常可能对模型产生误导作用，所以我放弃使用 Cutmixx 增强方法。

	efficientnet-b3	Cutout	CutMix
Image			
Label	Sunny1.0	Sunny1.0	Sunny0.8、Cloudy0.2

正是因为图像不具有形象的物体作为区分依据，所以 cutout 很适合作为本数据集的数据增强方式。其他与数据特征无关的数据增强均未使用，以加快训练速度。

(2) Resize 300x300:

由于 EfficientNet-b3 的 ImageNet 预训练模型所使用的图像尺寸为 300x300，为了最大程度上利用预训练模型的高起点，我也选择将图像 resize 到 300x300 。

(3) 数据清洗:

本次比赛数据集质量较高，训练过程中只发现因部分图像为灰度图像而报错，并未出现因图像损坏不可打开而报错的情况。而考虑到训练环境不固定，我选择使用线上数据清洗，在训练过程中实时将非 RGB 图像转为 RGB 图像。

对于数据集存在标签错误的问题，我曾尝试只使用标签错误问题较少的 Part1 文件夹作为训练数据进行训练，但效果并不理想。到比赛后期想到了使用精度最高的单模对训练数据做纠正，将预测置信度大于某固定值（如 0.95）且与原标签不同的标签作为真标签来训练新的模型。但由于时间问题没有来得及尝试，如果赛后开放数据则会继续验证想法。

3.4 模型搭建:

(1) 模型选择: EfficientNet-b3

本次建模我使用的 baseline 为参数量仅 12.23M的 EfficientNet-b3 。与同类相比 (b1, b7 等)，b3 具有较高的精度和较快的迭代速度，是我在权衡了精度和计算量后选择的 baseline。相比于常用的 Resnet50 的 25.6M 的参数量来说，推理时间大幅下降，而精度却相当甚至更高。

model	best_score	epochs	params # (M)	image_size
efficientnet-b1	0.7837540461952024	12	7.8	240
efficientnet-b2	0.7684045492528572	10	9.2	260
efficientnet-b3	0.7919572596542356	3	12	300
efficientnet-b4	0.7940881889323068	3	19	380
efficientnet-b5	0.8015530818774002	3	30	456
resnet50	0.7640881889323068	3	25.6	224
resnet101	0.7795530818774002	3	44.7	224

(2) 损失函数: Cross Entropy Loss

Cross Entropy Loss 很经典且常用的损失函数。

(3) 优化器: Adam+Lookahead+ SWA(随机权重平均)

由于本次比赛为**算力被限制**的线上赛, **调参**成了很大的问题。为了解决这个问题, 我经过多次尝试最终选择了 Adam+Lookahead+ SWA 这套优化器。这套优化器对超参数的选择不是十分敏感, 我并没有进行 LR_find 而是直接选择了常用的 $3e-4$ 作为初始学习率。**学习率微调则全权交给优化器**, 最终单模最高 0.8388, 由于技术不足虽然没进入 0.840 梯队, 但在 0.830 梯队中是单模最高的, 且**调参过程较为省心省力**。

3.5 模型训练: ImageNet 预训练, 混合精度

在训练之前选择使用 EfficientNet-b3 在 ImageNet 数据集上的预训练模型作为训练的初始模型, 因为经过观察, 天气数据集与 ImageNet 数据集非常相似。使用此预训练模型可以给出一个效果较好的初始起点, 降低了训练的时间成本, 且经过实际检验, 不使用预训练参数的模型的最终效果要远远低于使用预训练参数的效果。

训练过程中使用了混合精度技术**降低时间与空间成本**, 加快模型训练, 减少显存占用。

3.6 模型推理: 单模推理

根据决赛评价规则使用 F1-score 作为模型的评价指标, 对训练出的模型进行效果评估。选择出效果最好的单模对测试集进行推理, 获得 csv 文件后进行提交评分。

曾尝试过使用多模对不同模型进行融合, 虽然有一定的得分提升, 但综合考虑模型的**准确性, 高效性和实用性**之后, 我选择放弃多模方法, 而专注于**制造一套训练与推理都轻量快速且省心的模型**。

4、解题步骤

1. 问题分析
2. 数据预览和分析
3. 数据增强方法的选择和抛弃
4. 数据预处理准备训练
5. 确定 baseline
6. 模型训练（未过多调参，更多使用了自动学习率方法）
7. 模型评估
8. 模型反馈优化

5、创新细节

我对这次比赛的建模创新点主要体现在：保持较高的准确率情况下使训练和推理更轻便快速和省心。

我的模型在平台统一提供的显卡下，18 万张图片训练一轮需要不到 18 分钟，最好模型通常在第 4 轮出现。整个训练过程超不过两个小时，而 7.2 万张图片从推理到提交仅需不到 4 分钟。

```
100%|██████████| 4883/4883 [17:27<00:00, 4.66it/s]
100%|██████████| 730/730 [00:55<00:00, 13.14it/s]
0%|          | 0/4883 [00:00<?, ?it/s]
Epoch 0: train_loss 0.5471, train_F1_score 0.773, val_loss 0.5318, val_F1_score 0.784
100%|██████████| 4883/4883 [17:31<00:00, 4.65it/s]
100%|██████████| 730/730 [00:57<00:00, 12.79it/s]
0%|          | 0/4883 [00:00<?, ?it/s]
Epoch 1: train_loss 0.4590, train_F1_score 0.805, val_loss 0.4402, val_F1_score 0.810
100%|██████████| 4883/4883 [17:40<00:00, 4.61it/s]
100%|██████████| 730/730 [00:57<00:00, 12.79it/s]
0%|          | 0/4883 [00:00<?, ?it/s]
Epoch 2: train_loss 0.3938, train_F1_score 0.831, val_loss 0.3983, val_F1_score 0.828
100%|██████████| 4883/4883 [17:42<00:00, 4.60it/s]
100%|██████████| 730/730 [01:04<00:00, 11.34it/s]
0%|          | 0/4883 [00:00<?, ?it/s]
Epoch 3: train_loss 0.3668, train_F1_score 0.842, val_loss 0.3998, val_F1_score 0.830
21%|███       | 1007/4883 [03:40<13:36, 4.75it/s]
```

一轮不到18分钟

第四轮

线上0.8388


```

1 import pandas as pd
2 import csv
3 headers = ['id', 'weather']
4 f = open('test830.csv', 'w', newline='')
5 writer = csv.writer(f)
6 writer.writerow(headers)
7 predict()

```

100% | 285/285 [03:39<00:00, 1.30it/s]

模型中需要调整的超参数也非常少，训练过程中仅需调整 epochs 和 T_max 即可。其他参数设置之后无须再调整。

```

num_classes = 3
lr = 3e-4
epochs = 6
criterion = nn.CrossEntropyLoss()
optimizer = torch.optim.Adam(model.parameters(), lr=lr)
optimizer = Lookahead(optimizer)
swa_model = AveragedModel(model)
scheduler = CosineAnnealingLR(optimizer, T_max=3)
swa_scheduler = SWALR(optimizer, swa_lr=3e-5)

```

我认为我的模型轻便快速主要得益于以下几点：

1. 根据数据特征精心挑选了两个数据增强，又胆大心细地抛弃了其他数据增强，加快了模型收敛速度。
2. 使用预训练模型给模型一个较高的起点。
3. 使用参数量为 12.23M 的 EfficientNet-b3 作为主网络，同时兼顾速度与精度。
4. 根据比赛环境精心挑选的 Adam+Lookahead+ SWA 组合作为优化器使得超参数的调整变得简单，变相减少了训练成本。
5. 训练过程中使用了混合精度技术降低时间与空间成本，加快模型训练，减少显存占用。
6. 推理过程中本着轻量快速原则，不加任何融合手段，仅需不到四分钟即可完成推理提交。

6、后续改进

本次比赛较为遗憾的是没有完全发挥 SAW 方法的效果, 因为某些技术原因和时间原因, 我没有能成功利用上 saw_model 来对测试集进行推理, 理论上 saw_model 的效果应该比肩 Snapshot Ensembling (模型快照), 且推理时间不变, 仍为不到四分钟。其次是没有能够验证“用训练好的模型来纠正数据集中的标签错误”方法是否有效。

如果赛后数据集开放, 我将对以上两种方法进行验证, 即使没有开放, 也会在同类比赛中验证。由于参赛经验少, 前期浪费了很多时间在模型选择上, 训练起来没有章法, 边加 trick 边换模型, 到中后期才逐渐摸索到了方法, 从大家手里学到了很多東西。所以如果有机会, 我会对模型继续改进, 争取能在这场比赛中学习到更多知识和经验。