第30组 常见皮肤病智能分析 技术报告

> 组长：李思锐 12012719
>
> 组员：徐剑 12010923，张富珩12011322

一、项目背景

随着计算机技术的发展，基于人工智能的辅助诊疗系统已经陈伟一个主要的研究和发展趋势。到目前为止，这些系统已经在眼科、肿瘤学科和放射科学领域开发和研究，在皮肤科领域，一系列研究已经研究了良性和恶行皮肤肿瘤的分化，如良性痣、恶性黑色素瘤和角质细胞癌，例如反射共焦显微镜、皮肤镜和皮肤超声。然而，根据临床照片确定诊断良性色素性皮肤病尚无先例。我们的目标是提高医疗诊断的数字化、网络化以及基层医院的服务水平，减少医师培训时间，在提高医生的效率同时降低误诊率。

二、数据集介绍

我们本次使用的数据集是由香港大学深圳医院的曾丽华医生所提供的八分类皮肤病数据集。详细信息如下：

疾病名称	训练集	测试集
痤疮和玫瑰痤疮	556	191
湿疹和特应性皮炎	850	224
Nail Fungus甲癣	128	26
Nevi色素痣	171	43
Psoriasis银屑病	597	140
Seborrhoeic Dermatitis脂溢性皮炎	94	27
Urticaria Hives荨麻疹	155	39
Warts病毒疣	97	98

三、工作介绍

（一）训练框架

我们使用较为传统的深度学习框架来进行我们的训练。首先我们设置了随机种子，以确保得到稳定结果。其次我们令learning rate=0.0001、epochs=100，同时我们以交叉熵函数(CrossEntropy)作为Loss function，使用adam作为优化器，并使用配套的余弦退火调度器(CosineAnnealing Scheduler)来进行学习率的调整。Batch Size我们将在不同的网络模型上进行微调。

```python
# 设置随机种子
def setup_seed(seed):
    torch.manual_seed(seed)
    torch.cuda.manual_seed_all(seed)
    np.random.seed(seed)
    random.seed(seed)
    #  torch.backends.cudnn.deterministic = True
# 设置随机数种子
setup_seed(20)

# initialize the computation device
device = torch.device('cuda:5')

# initialize the model
model = ISIC_model.model(pretrained=True, requires_grad=False)
print(model)
model = model.to(device)

# learning parameters
lr = 0.0001
epochs = 100
batch_size = 32
optimizer = optim.Adam(model.parameters(), lr=lr)
scheduler = lr_scheduler.CosineAnnealingLR(optimizer, T_max = epochs)
# criterion_t = WeightedFocalLoss()
weight = torch.Tensor([5.2824, 3.4553, 22.9453, 17.1754, 4.9196, 31.2447, 18.9484, 7.6088])
criterion_t = nn.CrossEntropyLoss(weight=weight)
criterion_t = criterion_t.to(device)
critertion_v = nn.CrossEntropyLoss()
critertion_v = critertion_v.to(device)
```

上图为我们参数设置的具体代码。

考虑到原始数据集数据总量较少，仅有不到3000张输入图片，我们还使用了基础的数据增强来增强输入数据的多样性。我们所使用的数据增强有：RandomHorizontalFlip，RandomVerticalFlip以及RandomRotation。

（二）不同网络模型对比

我们主要对比了三种比较主流的图像处理模型：Resnet50，Efficientnet-b4以及Densenet201。考虑到Efficientnet-b4与Densenet201的网络大小较大，其batch size若与Resnet50设置成一样的话可能会导致显卡内存不够而使得训练无法正常进行，因此我们将二者的batch size调设置为8，learning rate调整为0.001，而Resnet50的batch size设置为100，learning rate仍然为0.0001。我们所使用的网络都已经过ImageNet的与训练。

最终训练出来的效果如下图所示：

	Accuracy	Precision	Recall	F1-Score
Resnet50	0.64	0.71	0.60	0.64
Efficientnet-b4	0.83	0.82	0.81	0.82
Densenet201	0.81	0.80	0.78	0.79

由此可见，Efficientnet-b4是最适合我们此次训练任务的网络模型。后续我们的一系列尝试也将使用Efficientnet-b4作为Baseline。

（三）图像混合策略使用

通过查阅文章我们得知，图像混合策略能够比起传统的旋转等数据增强更有效地增加输入特征的多样性，使得训练出模型的泛化性与鲁棒性得到提升。因此，我们选用了mixup以及Cutmix两种简单但却有效的图像混合策略，以期得到模型性能的提升。两种策略的简介如下：

mixup

> mixup通过叠加两张不同的输入图像来产生新的图像。我们可以控制超参数混合比例lam∈[0,1]来控制不同输入在混合图像中的显著程度。由于在图像混合的同时标签也会有所混合，而常见网络并不会处理带小数的软标签，因此我们需要把混合图像在两张输入图像的所属类下分别计算loss，再将loss以混合比例lam进行叠加。

Cutmix

> Cutmix与mixup原理类似，只是产生新图像的方式由叠加混合变成将输入图像1的一部分挖去，再填上输入图像2的对应部分来代替。混合比例lam∈[0,1]主要是指两个输入图像在输出图像中所占的面积。Loss部分的计算与mixup相同。

我们在保留**（二）不同网络模型对比**中Efficientnet-b4训练参数的基础上，分别加入了上述两种策略，取得的结果如下：

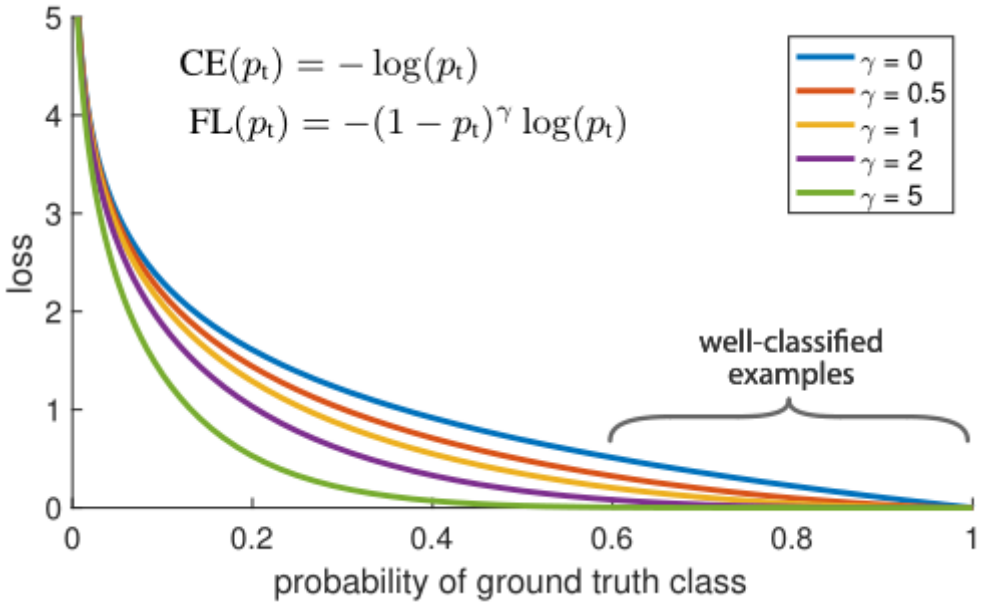
	Accuracy	Precision	Recall	F1-Score
Baseline	0.83	0.82	0.81	0.82
Mixup	0.85	0.87	0.82	0.84
Cutmix	0.86	0.87	0.85	0.86

可以看到，相较于mixup，Cutmix在皮肤病数据集上取得了相对优秀的结果。我们分析这是因为皮肤疾病图片特征较少且相对明显，mixup所带来的两张图片混合叠加可能会带来相对较多的不必要特征混合，因此效果可能稍微逊于Cutmix。

（四）Focal Loss的尝试

我们发现，初始的数据集其实呈现的是长尾分布(Long-tailed Distribution)，其特征为部分类别数据量较大，而部分类别数据量较少。这样的数据分布会使得深度学习网络在训练的过程中过度关注头部类（多数类）的特征，而忽略尾部类（少数类）的特征，从而降低训练出模型的性能。

通过查阅文献我们得知，Focal Loss是在解决长尾分布问题中应用较为广泛的一种Loss function，其在传统CrossEntropy的基础上加入了一个权重系数，使得头部类分类分错给模型参数调整带来的影响降低，而使尾部类分裂分错带来的影响提升。Focal Loss与CrossEntropy在具体数据上Loss的对比如下图：



但是比较遗憾的是，我们尝试在我们的数据集上引入Focal Loss时，模型的性能并没有得到很好的增加，甚至比起Baseline略有降低，而且我们一直未能找到原因所在。这是我们整个项目目前为止最大的遗憾，希望我们能在未来成功的找到问题所在！

我们在Efficientnet-b4中加入Focal Loss所得的训练结果如下：

	precision	recall	f1-score	support
0	0.94	0.91	0.92	191
1	0.82	0.73	0.78	224
2	0.87	1.00	0.93	26
3	0.79	0.95	0.86	43
4	0.72	0.77	0.74	140
5	0.55	0.63	0.59	27
6	0.71	0.90	0.80	39
7	0.91	0.86	0.88	98
accuracy			0.82	788
macro avg	0.79	0.84	0.81	788
weighted avg	0.83	0.82	0.82	788

但是，我们注意到Focal Loss的核心思维是“加权重”，那么别的形式的权重是否也能起到类似的效果？我们按 **总样本数 / 该类样本数**生成了新的权重，并将其加入了CrossEntropy中：

```
weight = torch.Tensor([5.2824, 3.4553, 22.9453, 17.1754, 4.9196, 31.2447, 18.9484, 7.6088])
criterion_t = nn.CrossEntropyLoss(weight=weight)
```

我们将加权重后的模型进行训练，并将加权重后的模型与Cutmix进行结合，取得了较为不错的最终结果：

	Accuracy	Precision	Recall	F1-Score
Baseline	0.83	0.82	0.81	0.82
Cutmix	0.86	0.87	0.85	0.86
Reweight	0.88	0.87	0.87	0.86
Cutmix + Reweight	0.89	0.86	0.88	0.87

（五）知识蒸馏使用

知识蒸馏是一种模型压缩方法，是一种基于“教师-学生网络思想”的训练方法。我们将已经训练好的模型包含的知识，蒸馏提取到另一个模型里面去。

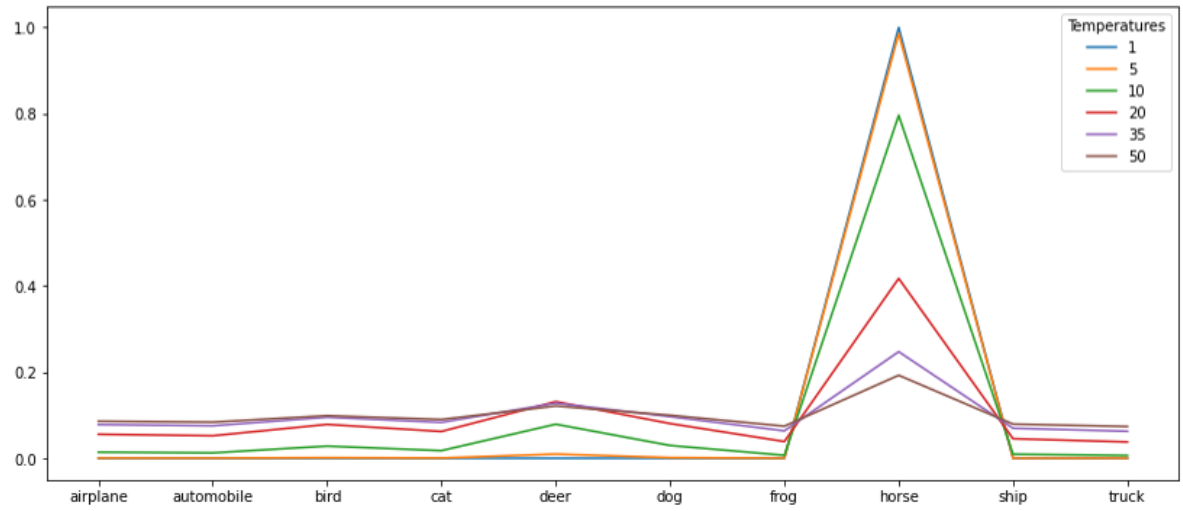
温度T是知识蒸馏中的一个重要超参数，下面这个公式是原始的softmax函数:

$$q_i = \frac{\exp(z_i)}{\sum_j \exp(z_j)}$$

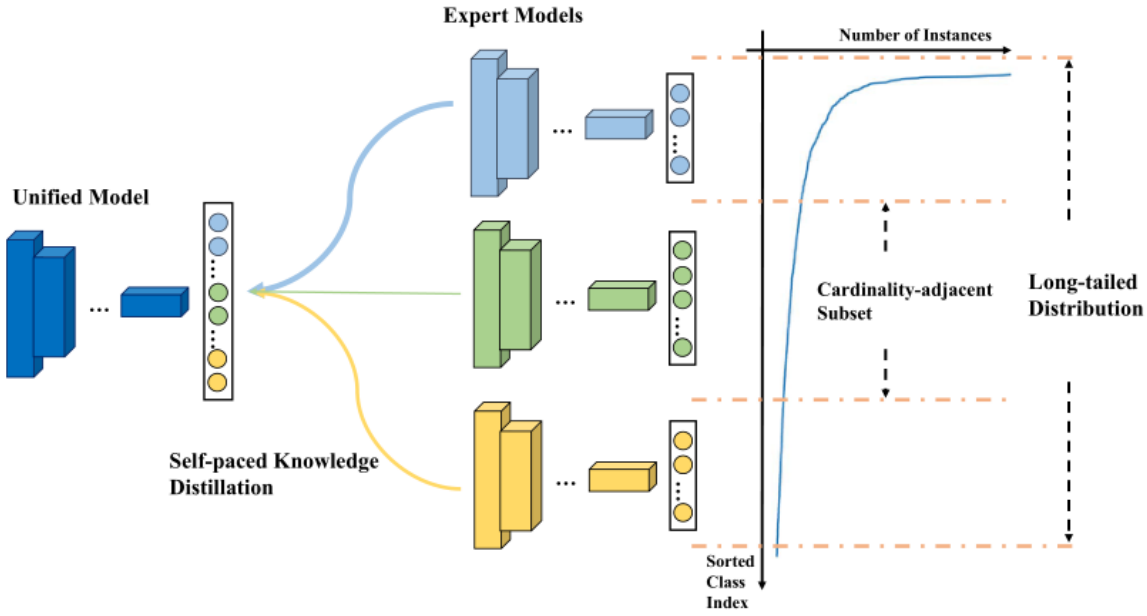
直接使用softmax层的输出值作为soft target, 这又会带来一个问题: 当softmax输出的概率分布熵相对较小时，负标签的值都很接近0，对损失函数的贡献非常小，小到可以忽略不计。因此**"温度"**这个变量就派上了用场。下面的公式时加了温度这个变量之后的softmax函数:

$$q_i = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}$$

T越高，softmax的输出分布越趋于平滑，其分布的熵越大，负标签携带的信息会被相对地放大，模型训练将更加关注负标签，从而学到更多信息。



在我们的实验中，则是运用了多个教师模型的知识蒸馏，我们将数据集的类别按照样张的数量分到三个子集中，每个子集中样本的数量差距较小，在一定程度上改善了原本数据集的不平衡问题。然后，我们在这三个数据集中分别训练出教师模型。再用这三个教师模型共同指导学生模型的训练，完成知识蒸馏。



在结果中我们可以看到，尽管整体的acc提升不大，但在这些样本较少的类别中，f1-score有了显著的提升。这意味着模型对少数类的辨别能力有了很大的提升，从而说明这一方法起到了作用。

Cross Entropy

	precision	recall	f1-score	support
0	0.91	0.95	0.93	191
1	0.84	0.79	0.81	224
2	0.96	0.96	0.96	26
3	0.90	0.88	0.89	43
4	0.77	0.76	0.77	140
5	0.50	0.59	0.54	27
6	0.84	0.79	0.82	39
7	0.84	0.89	0.87	98
accuracy			0.84	788
macro avg	0.82	0.83	0.82	788
weighted avg	0.84	0.84	0.84	788

Cross Entropy + Knowledge

	precision	recall	f1-score	support
0	0.90	0.94	0.92	191
1	0.79	0.83	0.81	224
2	1.00	0.92	0.96	26
3	0.97	0.91	0.94	43
4	0.74	0.76	0.75	140
5	0.80	0.59	0.68	27
6	0.87	0.87	0.87	39
7	0.92	0.85	0.88	98
accuracy			0.85	788
macro avg	0.88	0.83	0.85	788
weighted avg	0.85	0.85	0.85	788

未来工作

在使用知识蒸馏这一方法时，我们发现，教师在关系子集上的表现仍然有提升空间。这是因为，即使划分了关系子集，每个子集中各个类别的样本数量仍然存在着一定的不平衡的情况。针对这一问题，我们计划在教师的训练过程中融合进更多的训练方法，例如上文中提到的图像混合策略和Focal Loss。在提升了教师模型的表现后，我们认为蒸馏出的学生模型的表现也会有所改善。

课程学习 (Curriculum learning, CL) 是近几年逐渐热门的一个前沿方向。Bengio首先提出了课程学习（Curriculum learning，CL）的概念，它是一种训练策略，**模仿人类的学习过程，主张让模型先从容易的样本开始学习，并逐渐进阶到复杂的样本和知识。**CL策略在计算机视觉和自然语言处理等多种场景下，在提高各种模型的泛化能力和收敛率方面表现出了强大的能力。在这样的不平衡数据集中，我们可以先将从数据集中提取出各类别平衡的数据，让模型在这个子数据集上进行训练。随着训练的进行，我们将剩余的样本逐渐加入训练集中，从而实现一个从易到难的训练过程。

除此之外，由于我们在这一项目中使用的方法大多可以广泛应用在数据不平衡这一问题上。这意味着我们的数据集将可以不局限于皮肤病，在视网膜眼底图像等数据集中我们的方法应当同样适用。我们将会结合更多类型的数据集对我们的方法进行进一步的验证。