



**课 程 报 告**

|  |  |
| --- | --- |
| 课程名称 | 模式识别 |
| 题目名称 | 视网膜病变识别 |
| 专业班级 | 自动化卓越班 |
| 学号姓名 | 3120000881 童赞嘉 |
|  | 3120000875 马宏志 |
|  | 3120000860 何哲宇 |
|  |  |
|  |  |
| 指导教师 | 邢延 |

2023 年 5 月 3 日

**目录**

[1 模式识别系统设计 1](#_Toc30266)

[2 数据集与评价指标 3](#_Toc1215)

[2.1 图像裁剪及存储策略 3](#_Toc30022)

[2.2 多标签数据集的均衡分割 4](#_Toc20127)

[2.3 基于混淆矩阵的性能评估 6](#_Toc1081)

[3 特征提取 7](#_Toc5928)

[3.1 超参数进化 8](#_Toc6421)

[3.2 复合变换 9](#_Toc8529)

[3.3 半监督学习 10](#_Toc25606)

[4 分类器 11](#_Toc14685)

[4.1 决策树 12](#_Toc4579)

*[4.1.1 算法原理](#_Toc11237)* [12](#_Toc11237)

*[4.1.2 参数调整](#_Toc11156)* [14](#_Toc11156)

[4.2 SVM 15](#_Toc23061)

*[4.2.1 算法原理](#_Toc16856)* [15](#_Toc16856)

*[4.2.2 参数调整](#_Toc31982)* [17](#_Toc31982)

[5 结论 18](#_Toc20477)

[6 失败的尝试 19](#_Toc30609)

[6.1 网络模块 19](#_Toc21889)

*[6.1.1 SEReLU](#_Toc22746)* [19](#_Toc22746)

*[6.1.2 注意力池化](#_Toc26053)* [20](#_Toc26053)

[6.2 网络架构 21](#_Toc28370)

*[6.2.1 ViT](#_Toc9774)* [21](#_Toc9774)

*[6.2.2 FastViT](#_Toc8903)* [22](#_Toc8903)

[6.3 自监督学习 23](#_Toc13923)

[参考文献 24](#_Toc19209)

[附录 27](#_Toc7064)

Github链接：<https://github.com/Instinct323/DR_DME_grading>

项目结构：

├── engine：训练相关的引擎

├── trainer：神经网络训练器

├── loss：损失函数 (对比损失，焦点损失)

├── crosstab：混淆矩阵 (多分类)

├── evolve：超参数进化算法

├── scaling：EfficientNet 论文中的模型复合缩放

├── extension：模型预训练方案 (e.g., SimCLR, MAE)、线性探测

└── result：训练过程信息的结构化存储方法

├── model：计算机视觉模型

├── common：CNN、ViT 网络单元

├── model：yaml 文件配置的模型

├── fourier：傅里叶特征映射

├── utils：网络单元的注册方法，局部变量的传递方法

└── ema：Mean Teacher 的维护方法、半监督学习方法

├── deploy：torch 模型转换

├── onnx\_run：ONNX 模型管理

└── openvino：OpenVINO 模型管理

├── utils：拓展工具包

├── utils：通用工具箱

├── imgtf: 图像处理方法 (e.g., 颜色失真, 边界填充)

├── data：数据集相关处理方法 (e.g., 留出法, 欠采样, 数据池)

├── gradcam：梯度加权的类激活映射 (i.e., Grad-CAM)

├── prune：非结构化剪枝

├── plot: 可视化的基础函数, 以及部分高阶函数 (e.g., 参数利用率分析)

├── rfield：网络感受野可视化

└── teacher：快速知识蒸馏 (FKD) 的知识管理系统

├── config：yaml 配置文件

├── vovnet：VoVNet 的模型配置文件、权重文件

└── ml：SVM、决策树的超参数文件

├── data：图像裁剪及存储策略，基于混淆矩阵的性能评估

├── train：多标签分类的损失函数，训练方案

└── ml：VoVNet 实现的特征降维, 降维数据的分析, 机器学习算法评估

# 1 模式识别系统设计

视网膜病变识别任务有两个子任务，即糖尿病视网膜病变 (Diabetic Retinopathy)、糖尿病性黄斑水肿 (Diabetic Macular Edema)。

该数据集共含有1200张视网膜图像，每张图像都有一个DR等级 (i.e., 0, 1, 2, 3) 和一个DME等级 (i.e., 0, 1, 2)。我们对样本的等级进行了统计 (Figure 1)，发现DME等级标签的分布极其不均衡，其中DME-0的实例共有974个 (占比81.67 %)，而DME-1的实例只有75个 (占比6.25 %)。这意味着即使将所有视网膜图像预测为DME-0也会有81.67 %的正确率 (Accuracy)，所以我们引进一致性检验指标Kappa系数 [1] 以评估模型。

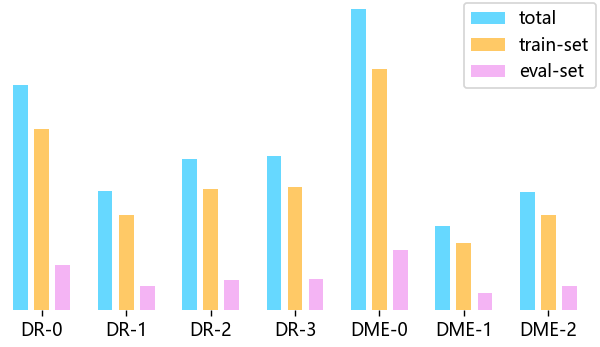


Figure 1: 各个类别所拥有的实例数，total表示数据集总体

严重的类别不均衡是视网膜病变识别任务的难点之一，图像的超高分辨率也是该任务的另一个挑战。如Figure 5所示，即使对视网膜图像进行必要的裁剪，图像分辨率的中位数 (假设为均值) 也高达1.3 k。如果直接使用传统的机器学习算法将该2D三通道图像视为1D向量，则该向量的维度达到了恐怖的5.07 M。

通过下采样降低图像分辨率以减少计算量是一个不错的选择。但另一方面，视网膜图像的病灶面积小 (Figure 2)，怎样设计合理的分辨率防止过多的信息丢失是另一个我们需要研究的问题。

为了处理高分辨率的视网膜图像，我们选择了卷积神经网络 (CNN) 作为特征提取器。ResNet [2] 受益于残差连接，相比于ViT [3]、MLP-Mixer [4] 具有极其优越的可训练性 [5]。虽然ResNet并没有出众的性能，但因其网络结构的超参数较少，受到了很多学术研究者的欢迎。

在选择ResNet-50之外，我们又选择了基于特征复用的网络ELAN [6] 和VoVNet [7]，并使用分辨率为512×512的视网膜图像进行实验。其中的每个模型均经过了长时间的参数利用率优化，直至无法突破性能瓶颈。此外，我们又通过自动化寻优方法 (超参数进化、复合变换 [8])、Mean Teacher算法 [9] 将VoVNet的性能提升到一个更高的水平 (详情见第5节)。

VoVNet通过大量的卷积算子、非线性激活函数将2D图像信息转变为144维的1D向量。基于多层感知机 (MLP) 的分类方法往往容易过拟合 [10]，为此我们重新选择了分类头。在对所提取的特征进行分析后，我们选择了决策树、SVM进行最终的分类。

|  |  |  |
| --- | --- | --- |
| 1-8-cam | 2-12-cam | 3-14-cam |
| DR-1 | DR-2 | DR-3 |
| 0-8-cam | 2-12-cam | 2-14-cam |
| DME-0 (阴性) | DME-1 | DME-2 |

Figure 2: 利用类激活映射 (CAM) [11], [12] 可视化视网膜病灶

# 2 数据集与评价指标

为了解决图像超高分辨率、类别严重不均衡所带来的一系列问题，在本章节中我们提出了一系列解决方案。

## 2.1 图像裁剪及存储策略

如Figure 3所示，未经任何处理的视网膜图像含有大量的“黑边”。这些冗余的像素点会大大降低特征提取器的效率，甚至像膨胀卷积一样引起卷积核退化 [13]。

图像处理中的腐蚀操作可被通俗地解释为“最小池化”。通过腐蚀操作可以有效抑制“黑边”中的噪声点，从而可以精准地提取有效区域的最小包围框，并对图像进行裁剪 (效果如Figure 2所示)。



Figure 3: 未经任何处理的视网膜图像

对所有图像进行这样的裁剪后，统计这些图像的分辨率、纵横比 (Figure 4)。尽管在经过这样的裁剪后，图像仍有很高的分辨率，为此我们选择了512×512的分辨率进行初步的实验。

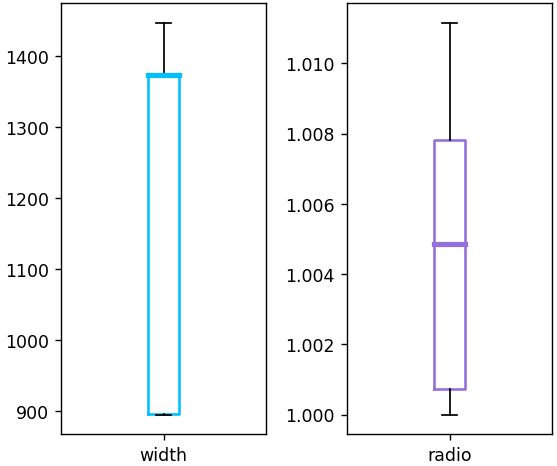


Figure 4: 处理后的视网膜图像的分辨率分布 (左)、纵横比分布 (右)

在数据集的存储上，我们选择以uint8的格式将图像存储于CPU上，在输入CNN之前再转化为float32。相比于直接转化为float32，这样的策略将CPU的内存开销降低到原来的1/4。

此外，数据类型的转化操作并非在CPU上进行，而是在GPU上进行的。这意味着从CPU“搬运”到GPU的数据类型是uint8而不是float32，从而有效降低张量的运输成本。这两个技巧有效地减少了我们训练CNN所使用的资源和时间。

## 2.2 多标签数据集的均衡分割

训练CNN是一项极其耗时的任务，而且其网络架构、训练配置中含有大量的超参数，故绝大多数研究使用留出法对数据集进行划分，以评价CNN的性能。

留出法即将数据集划分为两个互斥的子集，一个作为训练集，另一个作为验证集。对于单标签数据集而言，若要使某个类别的实例以一定比例分布于训练集、验证集，只需将该类别的实例进行简单的随机分割。但对于多标签数据集而言，这样的分割无法兼顾实例的其它标签，会使得各类别的实例分布失衡。

Table 1: 多标签数据集的表示格式

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | DR-0 | DR-1 | DR-2 | DR-3 | DME-0 | DME-1 | DME-2 |
| train\_0.jpg | 1 | 0 | 0 | 0 | 0 | 1 | 0 |
| train\_1.jpg | 0 | 0 | 1 | 0 | 0 | 0 | 1 |
|  |  |  |  |  |  |  |  |
| train\_1199.jpg | 0 | 1 | 0 | 0 | 1 | 0 | 0 |

为了实现各类别的实例分布均衡，我们使用如Table 1所示的数据格式以表示多标签数据集，并记为 。同时，定义以下变量：

* 训练集统计 ：训练集中各个类别的实例总数，初始全为 0
* 验证集统计 ：验证集中各个类别的实例总数，初始全为 0
* 图像统计 ：第 i 张图像各个类别的实例总数

初始状态下，，给定比例为 4:1，定义某时刻分割状态的 SSE 损失值为：

打乱数据集的图像顺序后，依次取出数据集中的图像 id、图像统计 ，将该图像加入训练集 / 验证集，则衍生2个子状态：

* 加入训练集：训练集统计变为 ，验证集统计仍为 ，对应的损失值为
* 加入验证集：训练集统计仍为 ，验证集统计变为 ，对应的损失值为

执行较小的损失值所对应的动作后，更新​ ，最后可使得数据集的分割结果逼近我们想要的分布均衡。对于视网膜病变数据集的分割结果如Table 2所示，其中每个类别都有约80%的实例分布在训练集。

Table 2: 视网膜病变数据集的分割结果

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | DR | | | | DME | | |
| 0 | 1 | 2 | 3 | 0 | 1 | 2 |
| Train set (80%) | 438 | 122 | 197 | 203 | 779 | 60 | 121 |
| Validation set (20%) | 109 | 31 | 49 | 51 | 195 | 15 | 30 |

## 2.3 基于混淆矩阵的性能评估

Table 3: DR分级的混淆矩阵

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | | Prediction | | | |
| 0 | 1 | 2 | 3 |
| Target | 0 |  |  |  |  |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |

以DR分级为例，每一个实例根据其真实级别、预测级别可被混淆矩阵统计 (Table 3)。对角线上的元素和为预测正确的实例总数，其与整个矩阵的和之比便是分类器的准确率 (Accuracy)：

准确率是最常用的分类器评价指标，但其并不适用于类别严重失衡的数据集。就像我们在前文所说的，即使将所有实例的DME分级预测为DME-0，也会有81.67 %的正确率。所以我们引进了一致性检验指标Kappa系数，其被定义为：

kappa系数是统计学中度量一致性的指标, 值域为[-1,1]，通常大于0。对于评分系统, 一致性就是不同打分人平均的一致性；对于分类问题，一致性就是模型预测结果和实际分类结果是否一致。上式中的 是两个标注者随机分配标签时的预期一致性，用以消除类别不均衡所带来的准确率增益。

如Figure 1所示，无论是DR分级还是DME分级，阴性样本 (即级别为0) 的占比都是最大的。为了进一步观察类别严重失衡对分类器性能的影响，我们又定义了“阴性精度 ”、“阳性平均精度 ”

# 3 特征提取

为了处理高分辨率的视网膜图像，我们选择了CNN作为特征提取器。ResNet [2] 受益于残差连接，相比于ViT [3]、MLP-Mixer [4] 具有极其优越的可训练性 [5]。虽然ResNet并没有出众的性能，但因其网络结构的超参数较少，受到了很多学术研究者的欢迎。

|  |  |
| --- | --- |
| Module | Module |
| (a) ELA | (b) CSP-OSA |
| Module | |
| (c) VoVNet | |

Figure 5: (a)(b) Wang et al. [6] 结合CSP stage [14] 所设计/改进的CNN模块; (c) 本文最终用于特征提取的CNN

在选择ResNet-50之外，我们又选择了基于特征复用的网络ELAN [6] 和VoVNet [7]，并使用分辨率为512×512的视网膜图像进行实验。其中的每个模型均经过了长时间的参数利用率优化，直至无法突破性能瓶颈。其最终性能如Table 4所示。

Table 4: 各个模型使用交叉熵损失训练后的性能，以及VoVNet经过超参数调整之后的性能

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| method | Params | time |  |  |  |  |  |  |  |  |
| ResNet | 5.489 M | 8.47 ms | 65.22 | 54.51 | 92.68 | 62.10 | 43.31 | 63.92 | 62.92 | 89.58 |
| ELAN | 3.686 M | 7.79 ms | 66.67 | 50.59 | **94.09** | 79.06 | 42.87 | 71.74 | 62.92 | 91.67 |
| VoVNet | 8.008 M | 9.12 ms | 66.67 | 56.88 | 92.82 | 72.32 | 45.47 | 68.54 | 65.00 | 91.25 |
| + HE | 8.008 M | 9.12 ms | **80.47** | **67.42** | 92.75 | **84.26** | **64.97** | **70.63** | **76.67** | **91.67** |

CNN的训练任务中存在众多超参数，如：颜色失真的幅度和概率、DropBlock [15] 的核尺寸及丢弃概率、图像分辨率……要调节这些超参数，需要人为地观测每一次训练的结果，并做出相应的调整。遗憾的是，这些超参数共有14个，加上每个CNN模型的训练时间在1个小时左右，人为地制定调整方案显然不现实。

在本节中，我们设计了超参数进化算法、复合变换方案搜索算法，以实现参数的自动化寻优。此外，我们还引进了Mean Teacher进一步提升了VoVNet的性能。

## 3.1 超参数进化

YOLOv5中的超参数进化算法允许随机的初态，在每一轮进化时都会选取一定数量的最优个体进行合成——正因为如此，只有在个体数量足够多的时候，该算法才能够搜索到全局最优解。这样的算法虽然可以搜索到全局最优解，但是对于显卡资源的需求极大，特别不适用于大数据集。

为此我们设计了另一种只搜索局部最优解的超参数进化算法。对于给定的超参数初态，在每一轮进化中只改变其中一个超参数 (随机选取)，并使用适应度函数 (模型在训练一定轮次后的性能) 计算是否取得了“进步”。如果是则更改超参数，否则令该超参数的耐心值减少 (归零时固定该超参数)。

为了更加高效地进行超参数进化，我们又为每个超参数定义了“搜索方向”。这些搜索方向会随着适应度的反馈而改变。比如，DropBlock的丢弃概率增大时，如果适应度减小则搜索方向为-1，否则搜索方向为+1。

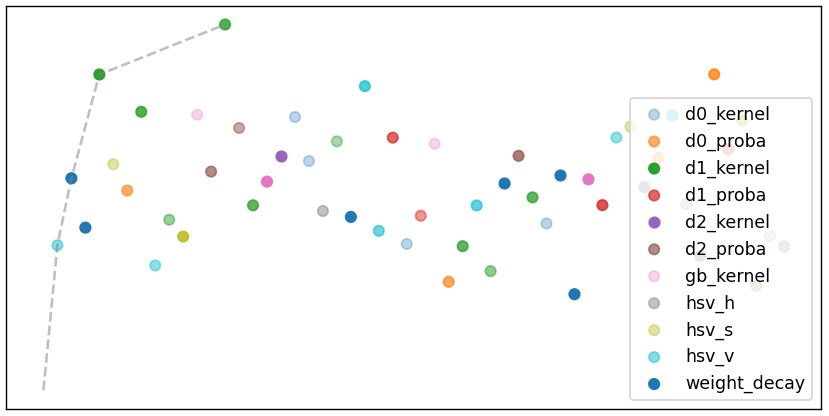


Figure 6: 超参数进化过程，横轴为进化轮次，纵轴为Acc，散点的颜色深/浅表示该参数增大/减小，虚线为最优Acc的变化曲线

## 3.2 复合变换

在前文的实验中，我们使用了分辨率512×512的视网膜图像对CNN进行训练。但是这样随意地指定分辨率是毫无理论依据的，过大的分辨率会提供过多的细节并对网络深度有更大的需求，过小的分辨率会损失太多的图像信息并导致模型欠拟合。

模型的浮点运算量 (FLOPs) 与图像分辨率的二次方成正比，也与模型宽度增益的二次方成正比。增加图像分辨率的同时，FLOPs也会显著地增加。所以我们使用了图像分辨率和模型宽度增益的复合变换 [8]，在确保FLOPs约等于baseline的1.2倍的情况下进行实验。

Table 5: 复合变换对VoVNet性能的影响，\* 为baseline，·为最优模型

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | img-size | width | FLOPs |  |  |  |  |  |  |  |  |
| 1 | 416 | 1.625 | 1.116× | 74.05 | 64.98 | 93.17 | 83.12 | 59.19 | 72.70 | 72.92 | 92.08 |
| 2 | 448 | 1.625 | 1.294× | 76.12 | 69.24 | 92.82 | 87.66 | 62.10 | 73.01 | 75.00 | 92.50 |
| 3 | 480 | 1.500 | 1.266× | 74.62 | 65.71 | 92.27 | 91.38 | 61.73 | 69.09 | 74.58 | 91.25 |
| 4 | 512 | 1.375 | 1.210× | 77.61 | 64.68 | 91.04 | 77.82 | 62.63 | 65.91 | 75.42 | 90.83 |
| \* 5 | 512 | 1.250 | 1.000× | 80.47 | 67.42 | 92.75 | 84.26 | 64.97 | 70.63 | 76.67 | 91.67 |
| 6 | 544 | 1.250 | 1.129× | 75.89 | 80.41 | 92.79 | 82.72 | 65.97 | 71.89 | 77.92 | 92.08 |
| 7 | 576 | 1.250 | 1.266× | 77.86 | 63.73 | 92.42 | **100.0** | 64.11 | 75.47 | 76.25 | 93.33 |
| ·8 | 608 | 1.125 | 1.142× | 81.15 | 66.75 | **94.63** | 92.52 | 64.86 | **81.32** | 76.25 | **94.58** |
| 9 | 640 | 1.125 | 1.266× | **83.33** | **80.50** | 92.79 | 77.08 | **66.76** | 70.38 | **77.92** | 91.67 |
| 10 | 672 | 1.000 | 1.103× | 81.30 | 63.95 | 93.27 | 82.60 | 64.63 | 71.76 | 76.25 | 92.08 |
| 11 | 704 | 1.000 | 1.210× | 79.20 | 58.65 | 92.27 | 89.66 | 63.09 | 67.61 | 75.42 | 90.83 |
| 12 | 736 | 1.000 | 1.323× | 80.49 | 64.91 | 92.75 | 75.25 | 64.09 | 69.20 | 75.83 | 91.25 |

如Table 6所示，图像分辨率的增加可以防止过多的信息丢失，但是一味地增加图像分辨率并不会为模型的性能带来提升。这是因为VoVNet在固定的网络架构下，其神经元的感受野不会随着图像分辨率、宽度增益而改变。随着图像分辨率的增加，VoVNet的感受野也发生了相对的缩小，最后使模型欠拟合。

## 3.3 半监督学习

Mean Teacher [9] 的目的是通过利用辅助模型的平滑权重来提供更稳定的监督信号。该算法主要涉及了两个模型 (Figure 7)：一个是学生模型 (即要训练的目标模型)；另一个是教师模型。

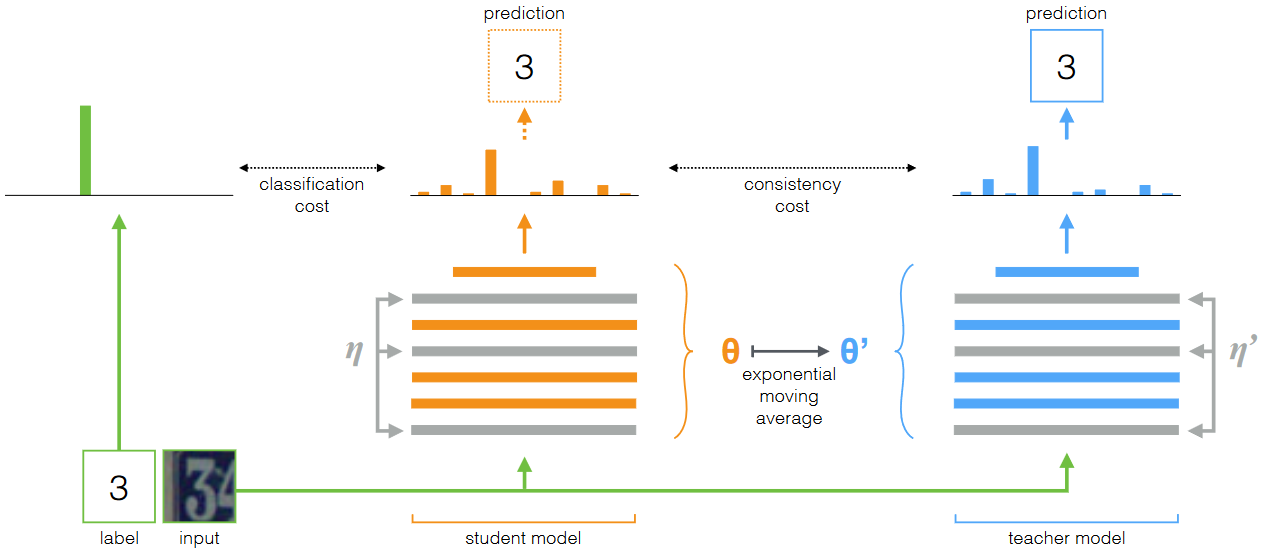


Figure 7: Mean Teacher 参与训练的方式

学生模型通过传统的监督学习方式进行训练，使用带有标签的数据。教师模型则通过指数移动平均参数将自身的权重平滑更新为学生模型的权重。教师模型的预测结果与学生模型的预测结果之间的差异以MSE损失的形式，参与学生模型的优化。

Table 6: Mean Teacher对VoVNet性能的影响

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| method |  |  |  |  |  |  |  |  |
| baseline | 81.15 | 66.75 | 94.63 | 92.52 | 64.86 | 81.32 | 76.25 | 94.58 |
| + Mean Teacher | **85.16** | **88.59** | **94.63** | **100.0** | **78.73** | **84.31** | **85.83** | **95.42** |

Mean Teacher算法通过辅助模型的平滑权重更新和知识传递，提供更稳定的监督信号，从而提高了模型的泛化能力、鲁棒性和半监督学习能力：

1. 提高模型的泛化能力：教师模型通过指数移动平均参数将自身的知识传递给学生模型。这种知识传递有助于主模型学习更鲁棒的特征表示和泛化能力，从而提高模型在未见过的数据上的性能。
2. 扩展半监督学习能力：Mean Teacher算法能够有效地利用未标记的数据，充分利用有限的标记数据和大量的未标记数据进行训练。这使得模型在标记数据有限的情况下仍然能够取得良好的性能。
3. 提高鲁棒性和抗噪能力：通过使用教师模型的预测结果作为额外的损失函数，Mean Teacher算法能够减少噪声样本对学生模型的影响。这使得学生模型能够更好地适应噪声和异常样本，提高模型的鲁棒性和抗噪能力。

# 4 分类器

前文中的VoVNet通过大量的卷积算子、非线性激活函数将2D图像信息转变为144维的1D向量，完成了特征提取的工作。为了选择合适的分类器，我们对所提取的特征进行了分析。

|  |
| --- |
|  |
|  |

Figure 8: DR各等级的特征均值 (上)，DME各等级的特征均值 (下)

进一步地，我们对计算了各个等级特征间的余弦相似度，并转化为它们在高维空间中的夹角 (Table 7)。其中DR-0与DR-1、DR-2、DR-3的特征夹角依次递增 (79.75°→ 160.2°)，也就是DR等级之间的差距越大，特征夹角也越大。这样的规律同样适用于DME等级。而仔细观测这些夹角又可以发现，DR-2和DR-3、DME-1和DME-2的夹角较小，这意味着它们是比较难以区分的，这使得它们成为模型的主要误差来源。

Table 7: 各等级的特征均值在高维空间中的夹角 (°)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | DR-0 | DR-1 | DR-2 |  |  | DME-0 | DME-1 |
| DR-1 | 79.75 | - |  |  | DME-1 | 155.7 | - |
| DR-2 | 141.9 | 105.6 | - |  | DME-2 | 172.1 | 32.25 |
| DR-3 | 160.2 | 116.6 | 48.02 |  |  |  |  |

Table 7证明了不同等级实例的特征在高维特征空间中有较大的夹角，所以可以使用基于特征加权的算法将它们映射到低维空间上进行分类。在此我们又选择了泛化能力最强的SVM算法进行分类。

在本节中，我们利用VoVNet所提取的特征，使用传统的机器学习算法进行了视网膜的病变分级，并使用K折交叉验证法。因为DME-1的实例过少，所以我们仅将K折交叉验证的K值定为5，并对这些机器学习算法进行了评估 (Table 8)。

Table 8: 以VoVNet作为特征提取器，各种分类头在5折交叉验证中的性能。灰色数据为baseline的性能，黑色数据为模型经过超参数进化之后的性能。

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| method |  |  |  |  |  |  |  |  |
| Decision Tree | 94.41 | 89.60 | 98.87 | 86.41 | 88.36 | 90.46 | 92.00 | 96.92 |
| 93.96 | 91.72 | 98.88 | 90.42 | 88.80 | 92.16 | 92.33 | 97.50 |
| + Bagging | 93.55 | 88.60 | 98.18 | 91.58 | 86.95 | 91.49 | 91.08 | 97.33 |
| **96.59** | **95.06** | 98.88 | 95.76 | **93.56** | 95.01 | **95.58** | 98.42 |
| + AdaBoost | 92.92 | 90.49 | 98.58 | 91.04 | 88.15 | 92.07 | 91.92 | 97.50 |
| 96.22 | 93.56 | **98.98** | **96.48** | 92.11 | **95.51** | 94.58 | **98.58** |
| SVM | 96.28 | 96.16 | 98.58 | 98.18 | 94.27 | 95.47 | 96.08 | 98.58 |
| **96.11** | 96.15 | **98.68** | **98.92** | 94.15 | **96.02** | 96.00 | **98.75** |
| + Bagging | 95.61 | 96.57 | 98.58 | 98.92 | 94.27 | 95.76 | 96.08 | 98.67 |
| 95.94 | **96.68** | **98.68** | **98.92** | **94.52** | **96.02** | **96.25** | **98.75** |

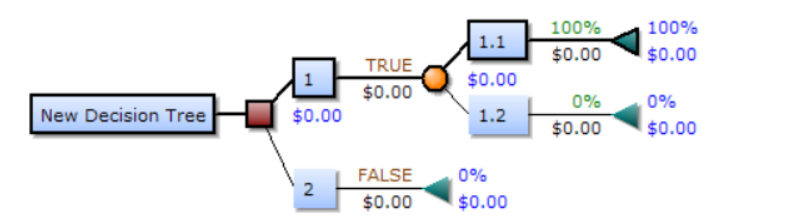
**4.1 决策树**

### *4.1.1 算法原理*

决策树是一种基于树形结构的分类和回归模型，它通过对数据集的分裂来构建一棵树，每个叶子节点代表一个分类或回归结果。决策树的构建过程是基于对数据的特征进行测试和分裂，使得在每个分裂后的子集中都能够得到最好的分类或回归结果。

一个决策树包含三种类型的节点：

1. 决策节点：通常用矩形框来表示
2. 机会节点：通常用圆圈来表示
3. 终结节点：通常用三角形来表示

****

|  |  |
| --- | --- |
| ID3算法 | 其核心是在决策树的各级节点上，使用信息增益方法作为属性的选择标准，来帮助确定生成每个节点时所采用的合适属性 |
| C4.5算法 | C4.5决策树生成算法相对于ID3算法的重要改进是使用信息增益率来选择节点属性。C4.5算法可以可是ID3算法存在的不足：ID3算法只适用于离散的描述属性，而C4.5算法既能够处理离散的描述属性，也可以处理连续的描述属性 |
| CART算法 | CART决策树是一种十分有效的非参数分类和回归方法，通过构建树，修剪树，评估树来构建一个二叉树。当终点是连续变量时，该数为回归树；当终点是分类变量，该数为分类树 |

1. 信息增益

熵与条件熵之差称为互信息，决策树学习中信息增益就等价于

训练数据集中类与特征的互信息，对于训练数据集和特征,信息增益可以表示为

信息增益的特征选择方法为：对训练集计算其每个特征的信息增益并比较大小，选择信息增益最大的特征。

信息增益的算法如下：

输入：训练数据集和特征

输出：特征对训练数据集的信息增益

1. 设有个为的1范数，表示样本个数)，计算数据集D的经验熵
2. 计算特征对数据集的经验条件熵
3. 计算信息增益
4. 随机森林

随机森林是一种集成算法，它属于Bagging类型，通过组合多个弱分类器，最终结果通过投票或取均值，使得整体模型的结果具有较高的精确度和泛化性能。其可以取得不错成绩，主要归功于“随机”和“森林”，一个使它具有抗过拟合能力，一个使它更加精准。随机森林相对于Bagging只是对其中一些细节做了自己的规定和设计。

1. 弱分类器：随机森林使用了CART决策树作为弱学习器。
2. 随机性：在生成每棵树的时候，每个树选取的特征都仅仅是随机选出的少数特征，一般默认取特征总数m的开方。而一般的CART树则是会选取全部的特征进行建模。因此，不但特征是随机的，也保证了特征随机性。
3. 样本量：相对于一般的Bagging算法，随机森林会选择采集和训练集样本数N一样个数的样本。
4. 特点：由于随机性，对于降低模型的方差很有作用，故随机森林一般不需要额外做剪枝，即可以取得较好的泛化能力和抗过拟合能力。当然对于训练集的拟合程度就会差一些，也就是模型的偏倚会大一些，仅仅是相对的。

### *4.1.2 参数调整*

用于分类的决策树有几个较为重要的超参数：

1. 最大深度max\_depth：决策树的最大深度是指从根节点到最深叶子节点的层数。该参数控制着决策树的复杂度和泛化能力。当max\_depth太小，决策树无法拟合数据，出现欠拟合；当max\_depth太大，决策树可能过度拟合，出现过拟合。因此，要根据具体数据集来选择合适的max\_depth，以达到最优的性能和准确率。
2. 节点分裂的最小样本数min\_samples\_split：该参数定义了在拆分节点之前必须有的最小样本数。如果一个节点的样本数量小于min\_samples\_split，则不会继续拆分。min\_samples\_split的选择影响着决策树的生成速度和泛化能力。当min\_samples\_split太大，决策树可能过于简单，无法捕获数据的复杂结构，从而导致欠拟合；当min\_samples\_split太小，决策树可能会过于复杂，过度拟合数据，从而导致泛化性能差。
3. 叶子节点的最小样本数min\_samples\_leaf：该参数定义了叶子节点必须拥有的最小样本数。如果一个叶子节点的样本数量小于min\_samples\_leaf，则会被认为是无效的，被剪枝。与min\_samples\_split类似，min\_samples\_leaf的选择影响着决策树的生成速度和泛化能力。当min\_samples\_leaf太大，可能会导致决策树欠拟合，因为模型无法分割出足够的决策规则；当min\_samples\_leaf太小，可能会导致决策树过拟合，因为模型过于敏感于噪声和异常值。

为了高效地调节这些超参数，我们使用了3.1中的超参数进化算法进行调节。从Table 8可以看到，这样的调节对于决策树的性能的改进是显著的。

随机森林是一种基于决策树的集成学习方法 (Bagging)，它有效地解决了决策树容易过拟合的问题。随机森林通过对样本和特征进行随机采样，生成多棵决策树并通过投票来进行预测。这样可以降低单棵决策树的方差和提高模型的泛化能力。相比之下，Adaboost 不是对样本进行随机采样，而是在训练过程中，通过对误分类样本的权重进行调整，使得基分类器更加关注于误分类样本，从而提高整个模型的预测能力。

|  |  |
| --- | --- |
|  |  |

Figure 9: 随机森林 (左) / AdaBoost (右) 的超参数进化过程 (纵轴为Acc)，n\_uint为决策树的数量，mss\_uint为节点分裂的最小样本数，msf\_uint为叶节点的最小样本数，lr\_float为AdaBoost算法的学习率

Adaboost相比于Bagging有以下几个优势：

1. 更强的泛化能力：Adaboost 可以通过对误分类样本的加权来使得基分类器更加关注于难分类的样本，从而提高模型的泛化能力。
2. 可以有效减少偏差：Adaboost 可以通过迭代的方式逐渐减少模型的偏差，因此可以在一定程度上解决过拟合的问题。

相比于简单的决策树，Adaboost和Bagging这两种集成学习方法都引入了新的超参数，我们同样使用了3.1的超参数进化算法进行调节 (Figure 9)。

**4.2 SVM**

### *4.2.1 算法原理*

支持向量机 (SVM)是一种常用的分类和回归算法。其基本思想是找到一个超平面 (线性或非线性)，使得该超平面能够将不同类别的数据点分开，并且具有最大的间隔。SVM算法主要有以下步骤：

1.建立模型：

假设有一组训练样本数据：{(, ), (, ), ..., (, )}其中表示第个样本的特征向量，表示第个样本的类别标签，取值为1或-1。我们的目标是找到一个超平面，将不同类别的数据点分开，并且具有最大的间隔。超平面的方程可以表示为：，其中w是一个n维向量，表示超平面的法向量，b是一个常数，表示超平面与原点的距离。对于二分类问题，我们希望将训练样本分为正例和反例两类，可以将正例的类别标签设为+1，反例的类别标签设为-1。

2.硬间隔最大化：

如果训练数据是线性可分的，那么我们可以使用硬间隔最大化来确定超平面的参数。硬间隔最大化的目标是找到一个超平面，使得所有正例样本都在超平面的一侧，而所有反例本都在超平面的另一侧，并且两侧之间的距离最大。这可以表示为以下优化问题：

其中，表示向量W的模长，表示间隔的倒数。目标函数表示要最大化间隔，约束条件表示超平面必须将正例和反例样本正确分类。这个优化问题可以通过拉格朗日乘子法转化为对偶问题：

其中，是拉格朗日乘子，通过求解最大化对偶问题的，可以得到超平面的参数和：

3.软间隔最大化：

如果训练数据是线性不可分的，那么我们可以使用软间隔最大化来确定超平面的参数。软间隔最大化的目标是找到一个超平面，使得分类误差最小，同时也要最大化间隔。这可以表示为以下优化问题：

其中，是松弛变量，表示第个样本被误分类的程度，是正则化参数，用于控制分类误差和间隔的平衡。目标函数表示要最大化间隔，并且最小化误分类样本的总数，约束条件表示超平面必须将正例和反例样本正确分类，

并且允许一定程度的误分类。同样可以通过拉格朗日乘子法转化为对偶问题：

其中，是拉格朗日乘子，同时需要引入另一个拉格朗日乘子来处理松弛变量。通过求解最大化对偶问题的，可以得到超平面的参数和。

4. 核函数：

在实际应用中，很多时候数据并不是线性可分的，需要使用非线性的分类器。这时可以使用核函数来将数据映射到高维空间，使得在高维空间中数据可以线性可分。常用的核函数包括线性核、多项式核和高斯核等。

对于一个核函数，可以将其表示为内积的形式：，其中φ(x)是将x映射到高维空间的函数。通过这种方式，我们可以在低维空间中使用内积的方式来计算高维空间中的距离，从而避免了直接计算高维空间中的内积。

SVM算法的核心在于如何构建支持向量和选择合适的核函数。支持向量是指离超平面最近的数据点，它们对于模型的构建和预测具有重要的作用。而核函数则是用来将低维空间的数据映射到高维空间，从而使得数据可以线性可分或非线性可分。

### *4.2.2 参数调整*

在scikit-learn所提供的SVM算法中，有几个较为重要的超参数：正则化参数C、核函数参数gamma。

正则化参数C控制着SVM分类器对误分类的惩罚力度，C越大则分类器越不容许出现误分类，C越小则分类器容忍出现误分类。参数C的增大可以提高训练集的拟合能力，但可能会导致过度拟合，降低分类器的泛化能力；参数C的减小可以提高分类器的泛化能力，但可能会导致欠拟合，降低分类器的分类准确度。

参数gamma是SVM分类器的一个核函数参数，它控制着数据点对分类结果的影响范围，gamma越小则影响范围越广，分类边界比较平缓；gamma越大则影响范围越小，分类边界变得更加陡峭。参数gamma的选择直接影响到分类器的复杂度和泛化能力。调节参数C和gamma是SVM分类器的重要步骤，但调节过程存在一定的困难：

1. 参数C和gamma的取值范围通常很大，需要手动尝试多个值来找到最优的超参数组合，需要耗费大量的时间和计算资源。
2. 参数C和gamma的选择相互影响，需要在多个参数组合之间进行权衡，增加了参数调节的复杂度。
3. 参数C和gamma的影响往往需要在训练集和测试集上进行实验验证，调节过程比较繁琐。

使用自动化的超参数调优可以有效地解决这些问题。我们沿用了3.1中的超参数进化算法对这些参数进行调节 (Figure 10)。

|  |  |
| --- | --- |
|  |  |

Figure 10: SVM (左) / 装袋SVM (右) 的超参数进化过程，c\_float为正则化参数，gamma\_float为径向基核函数的惩罚参数，n\_uint为弱学习器数量

Bagging是集成学习中一种常用的方法，它通过随机抽取训练集中的样本，训练多个分类器，并将它们的结果进行集成，可以有效地降低分类器的方差，提高分类器的泛化能力。结合集成学习的Bagging可以提高SVM分类器的性能和泛化能力：

1. Bagging可以减少SVM分类器的方差，提高分类器的泛化能力。通过对训练集进行随机采样，可以得到多个样本子集，从而训练多个分类器，最终将它们的结果进行集成，可以有效地降低分类器的方差，提高分类器的泛化能力。
2. Bagging可以降低SVM分类器的过拟合风险。通过随机抽样和集成多个分类器，可以降低分类器对训练集的过拟合风险，提高分类器的泛化能力。
3. Bagging可以提高SVM分类器的准确度和稳定性。通过集成多个分类器，可以获得更为准确的分类结果，并提高分类器的稳定性。

显而易见的是 (Table 8)，Bagging有效改进了SVM算法的性能，并取得了最优的分类效果。

# 5 结论

在本次模式识别的系统设计中，我们使用了VoVNet [7] 作为特征提取器，使用集成的决策树、SVM作为分类器，对视网膜图像进行了DR分级和DME分级。

针对特征提取器，我们使用留出法对卷积神经网络 (CNN) 的性能进行验证。我们使用了超参数进化算法、复合变换方法 [8]、Mean Teacher算法 [9] 改进了VoVNet的性能，使其DR分级的准确率从65.00 %提升到85.83 %，DME分级的准确率从91.25 %提升到95.42 %。

针对分类器，我们使用5折交叉验证对机器学习算法的性能进行验证。其中以决策树为基学习器的AdaBoost的两个准确率分别为94.58 %、98.58 %，装袋SVM的两个准确率为96.25 %、98.75 %。

在第2节中我们对视网膜图像进行了裁剪处理，但裁剪后的图像仍存在约21.46 %的无效面积，这些无效面积会引起CNN的卷积核的退化。如何对视网膜图像的数据做进一步优化是接下来我们需要研究的问题。

# 6 失败的尝试

对于视网膜的病变分级任务，除了以上的有效措施之外，我们还进行了很多有意义的尝试。但由于这些尝试的相关数据都没有保留，所以我们仅仅展示了部分代码、程序输出。

**6.1 网络模块**

在初期的网络结构调整中，VoVNet的DR分级准确率便遇到了瓶颈。我们经过初步的评估，猜测原因在于网络的感受野不够。于是我们尝试添加具有全局建模能力的模块对模型进行改进。

### *6.1.1 SEReLU*

SEReLU由SEBlock [16] 与激活函数ReLU级联组成。SEBlock首先通过对特征图进行全局平均池化以获得通道信息 ，然后通过多层感知机 (MLP) 将其变换成通道注意力 ，以此重新评估各通道信息的重要性。

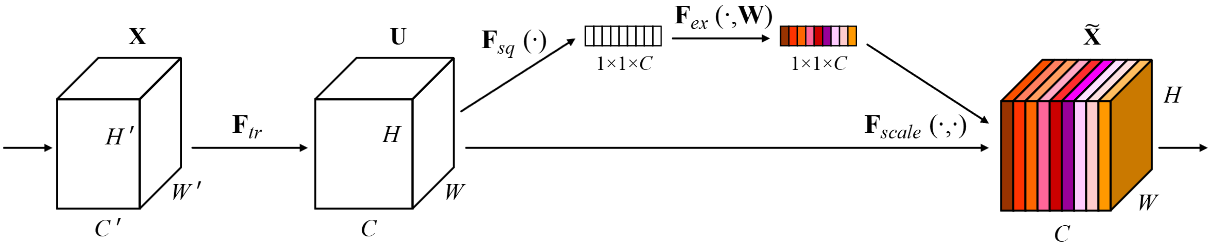


Figure 11: SEBlock的运算过程

但是SEReLU并没有改善网络的性能，反而因为全局平均池化降低了模型的运算效率。我们猜测可能的原因是，视网膜图像的病灶较小，不同位置对不同通道的需求不一致，而SEBlock的全局平均池化因无法捕获这种信息而发生了退化。

### *6.1.2 注意力池化*

注意力池化最早被应用于CLIP [17]，其可以替代全局平均池化进行全局信息的聚合。其将2D的特征图看作自然语言处理 (NLP) 中的1D单词序列，以全局平均池化值作为query，特征图的各像素点作为key，借助多头注意力 [18] 完成全局信息对局部信息的查询。

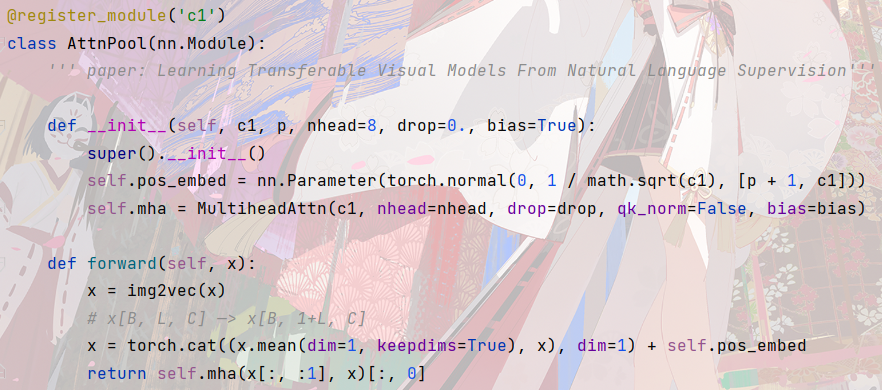


Figure 12: 注意力池化的代码 (位于model.common.py)，主要参数来源于位置编码、多头注意力

强大的注意力池化同样没有带来应有的性能提升。注意力池化的核心在于将特征图各个像素点的信息进行加权，但从Figure 2、Figure 13可以看到，病变等级与病变区域面积有着强烈的联系，因此每个位置的信息都应该享有相同的权重。由此可见，全局平局池化是最优的选择。

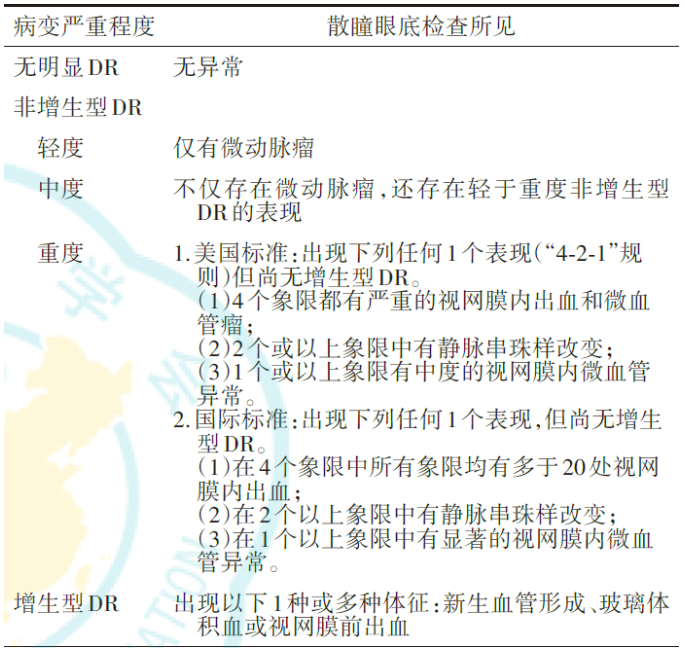


Figure 13: DR的国际临床分级标准 (2019年)，病变等级与病灶面积强相关

**6.2 网络架构**

尽管获得了SEReLU、注意力池化的相关实验结果，但我们仍不能确定神经网络是否受益于更大的感受野。为此我们选择了全局建模能力更强大的网络架构进行实验。

### *6.2.1 ViT*

Vision Transformer (ViT) [3] 由于堆叠了具有全局建模能力的Transformer Encoder [18]，从而具有全局感受野。但庞大的参数量、二次的计算复杂度使得ViT是难以训练的。

针对具有圆形视野的视网膜图像，我们利用Patch Embedding的特性，使得ViT可以忽略视网膜图像中21.46 %的无效区域 (Figure 14)。这直接移除了对无效区域无意义的计算，有效预防了网络参数的退化，并将ViT的运算速度提升到原来的151.8 %。

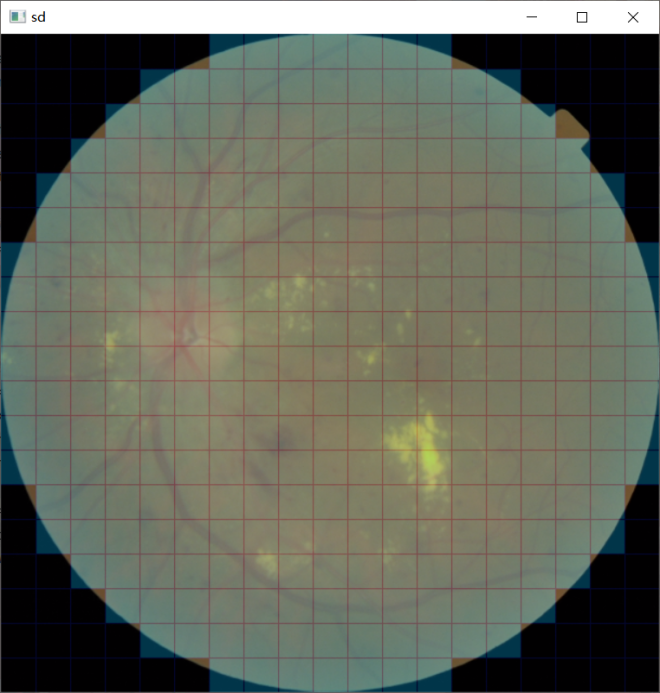


Figure 14: 分辨率608×608的图像被分割成19×19个补丁，只有蓝色区域的补丁 (293×) 才被选取参与ViT的前向传播

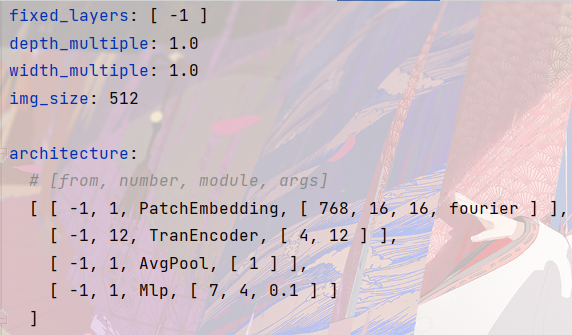


Figure 15: 本项目所使用的ViT配置文件，其中Patch Embedding使用了傅里叶特征 [19] 作为位置编码，Transformer Encoder采纳了 [20] 对多头注意力的修改建议，使用Average Pooling表示全局信息为 [21] 的建议。

在训练ViT的过程中，我们发现ViT的收敛速度极慢——尽管参数量远远大于前文中的CNN。为此我们探索了一系列改善ViT训练稳定性的技巧，但仍然无法解决。我们认为可能有以下原因：

1. 极低的可训练性：Chen et al. [5] 对CNN和ViT的损失地貌进行了可视化，发现ViT收敛于极尖锐的局部最小值。而且ViT对初始化较为敏感，容易过拟合甚至发散。

2. 数据需求量大：ViT在 [3], [5], [22] 中被证明严重依赖于大规模的预训练、数据增强。而我们的视网膜数据集只有1200张图像，这是远远不够的。

### *6.2.2 FastViT*

最近，Kumar et al. 提出了时延极低的FastViT [23]。与直觉相反的是，FastViT并不属于ViT系列，而是CNN。FastViT使用大尺寸的深度可分离卷积替换了多头注意力，这样做使得网络可以以廉价的计算获取大的感受野。而且FastViT采用过参数化训练，在训练完成后利用重参数化技术 [24] 合并多个并行的卷积核。这样的训练方式既借助了多分支结构的优越的学习能力，又在推理阶段有效屏蔽了多分支结构带来的时延。

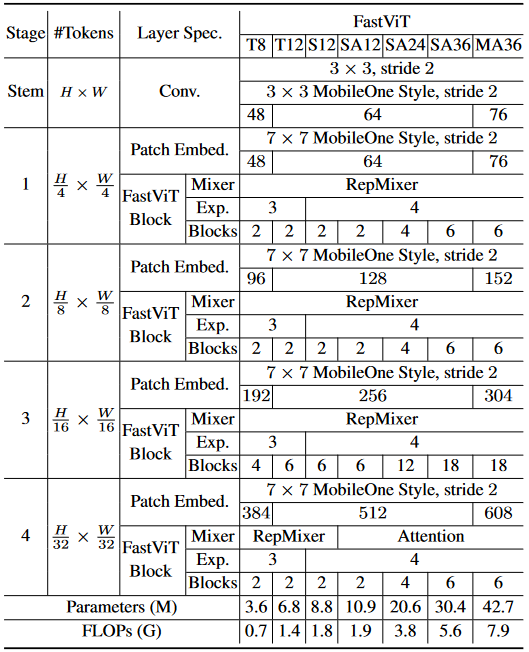


Figure 16: 论文中FastViT的结构 (左)；本文复现的FastViT (右)，其中的Fast Attention对应原文中的RepMixer

但是对于视网膜的病变分级，FastViT并没有取得改进，甚至劣于ResNet。在CNN系列中，FastViT拥有前所未有的超大感受野，但这也成为了它的硬伤。

Chen et al. [13] 发现，3×3的空洞卷积在膨胀系数较大时会退化成1×1卷积。这是由于空洞卷积对“无效区域 (padding)”的过度学习造成的。相似地，FastViT的边界像素点也因拥有超大的感受野而使得卷积核退化。CNN优越的性能并不来源于大的感受野，而是视觉任务和感受野大小的匹配程度。

**6.3 自监督学习**

我们认为模型的性能受限于数据集的样本量，所以研究了一系列与数据增强相关的训练方法。在6.1中，我们验证了视网膜的DR分级与病变面积高度的相关性，这样的特性使得随机裁剪技术无法应用。所以本项目的数据增强只使用了随机翻转、颜色失真、高斯模糊。

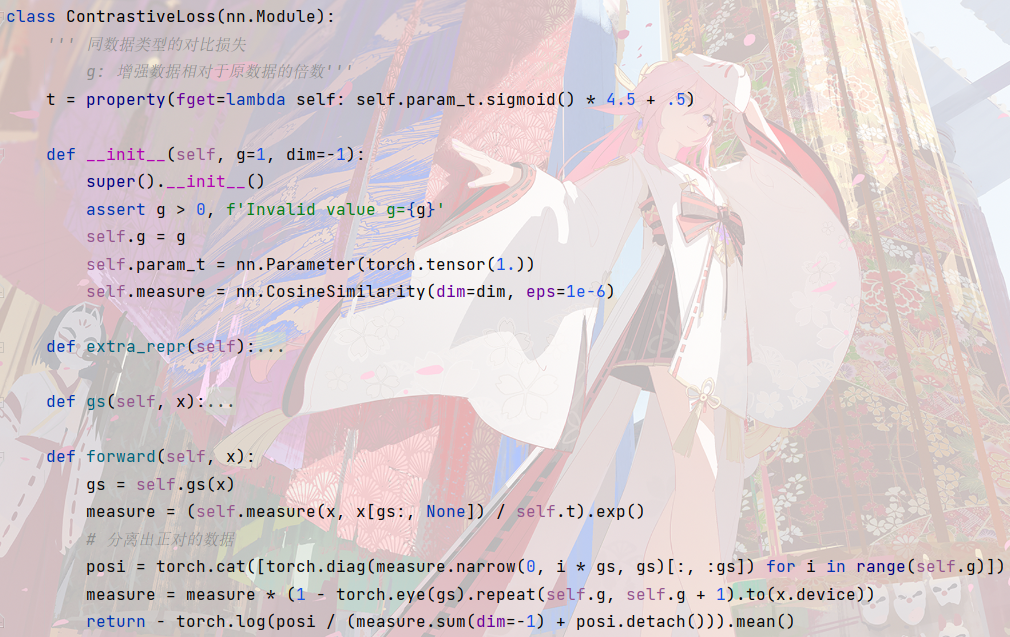


Figure 17: 根据 [25] 复现的对比损失函数

SimCLR [25] 提供了一种自监督学习方案：在每一次前向传播中，为每一张图片创建2个增强数据，并最大化这两者之间的余弦相似度。同时，最小化不同图像之间的余弦相似度 (Figure 17)。

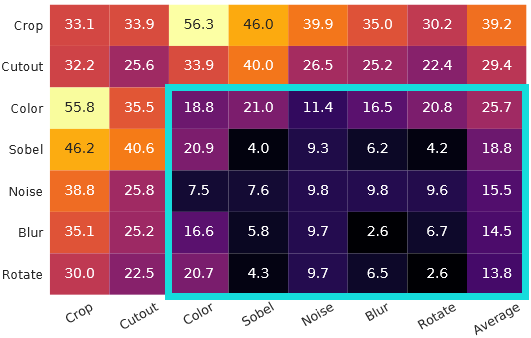
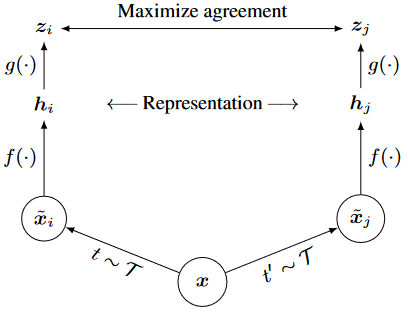


Figure 18: SimCLR的训练框架 (左); Chen et al. 关于数据增强的实验 (右)，其中的数字表示经过SimCLR训练后，线性探针在ImageNet数据集的Top-1准确率

Chen et al. 发现，这种对比学习受益于几种强大的数据增强技术：随机裁剪、颜色失真、高斯模糊。但只要缺少随机裁剪手段，对比学习的性能会急剧下降 (Figure 18)。相似地，我们的视网膜数据集由于缺少了随机裁剪，网络在初期的学习中很快收敛，而无法学习到可用于迁移学习的通用视觉表征。

为什么随机裁剪会如此重要？因为随机翻转、颜色失真、高斯模糊仅仅是对原图像的信息进行线性变换，这种线性变换很容易被模型所捕获，从而无法提供高难度的对比任务。而随机裁剪技术引入了图像的不同裁剪变化，有助于模型学习到更具判别性的特征，使得样本的表示在不同裁剪视图下保持一致性。

# 参考文献

[1] A. He, T. Li, N. Li, K. Wang, and H. Fu, “CABNet: Category Attention Block for Imbalanced Diabetic Retinopathy Grading,” *IEEE Trans. Med. Imaging*, vol. 40, no. 1, pp. 143–153, Jan. 2021, doi: 10.1109/TMI.2020.3023463.

[2] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 770–778. doi: 10.1109/CVPR.2016.90.

[3] A. Dosovitskiy *et al.*, “AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE,” *ArXiv Prepr. ArXiv201011929*, 2020.

[4] I. Tolstikhin *et al.*, “MLP-Mixer: An all-MLP Architecture for Vision,” *Adv. Neural Inf. Process. Syst.*, 2021.

[5] X. Chen, C.-J. Hsieh, and B. Gong, “When Vision Transformers Outperform ResNets without Pre-training or Strong Data Augmentations.” arXiv, Mar. 13, 2022. Accessed: Mar. 25, 2023. [Online]. Available: http://arxiv.org/abs/2106.01548

[6] C.-Y. Wang, H.-Y. M. Liao, and I.-H. Yeh, “Designing Network Design Strategies Through Gradient Path Analysis.” arXiv, Nov. 09, 2022. Accessed: Nov. 20, 2022. [Online]. Available: http://arxiv.org/abs/2211.04800

[7] Y. Lee, J. Hwang, S. Lee, Y. Bae, and J. Park, “An Energy and GPU-Computation Efficient Backbone Network for Real-Time Object Detection,” in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 752–760. doi: 10.1109/CVPRW.2019.00103.

[8] M. Tan and Q. V. Le, “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks,” *Int. Conf. Mach. Learn.*, 2019.

[9] A. Tarvainen and H. Valpola, “Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[10] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overﬁtting,” *J. Mach. Learn. Res.*, vol. 15, pp. 1929--1958, 2014.

[11] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, “Learning Deep Features for Discriminative Localization,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, USA: IEEE, Jun. 2016, pp. 2921–2929. doi: 10.1109/CVPR.2016.319.

[12] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization,” *Proc. IEEE Int. Conf. Comput. Vis.*, p. 9, 2017.

[13] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam, “Rethinking Atrous Convolution for Semantic Image Segmentation.” arXiv, Dec. 05, 2017. Accessed: Feb. 20, 2023. [Online]. Available: http://arxiv.org/abs/1706.05587

[14] C.-Y. Wang, H.-Y. Mark Liao, Y.-H. Wu, P.-Y. Chen, J.-W. Hsieh, and I.-H. Yeh, “CSPNet: A New Backbone that can Enhance Learning Capability of CNN,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, Seattle, WA, USA: IEEE, Jun. 2020, pp. 1571–1580. doi: 10.1109/CVPRW50498.2020.00203.

[15] G. Ghiasi, T.-Y. Lin, and Q. V. Le, “DropBlock: A regularization method for convolutional networks,” *Adv. Neural Inf. Process. Syst.*, p. 11, 2018.

[16] J. Hu, L. Shen, and G. Sun, “Squeeze-and-Excitation Networks,” *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, p. 10, 2018.

[17] A. Radford *et al.*, “Learning Transferable Visual Models From Natural Language Supervision,” *Int. Conf. Mach. Learn.*, 2021.

[18] A. Vaswani *et al.*, “Attention is All you Need,” *Adv. Neural Inf. Process. Syst.*, vol. 30, 2017.

[19] M. Tancik *et al.*, “Fourier Features Let Networks Learn High Frequency Functions in Low Dimensional Domains,” *Adv. Neural Inf. Process. Syst.*, 2020.

[20] M. Dehghani *et al.*, “Scaling Vision Transformers to 22 Billion Parameters.” arXiv, Feb. 10, 2023. Accessed: Feb. 19, 2023. [Online]. Available: http://arxiv.org/abs/2302.05442

[21] X. Zhai, A. Kolesnikov, N. Houlsby, and L. Beyer, “Scaling Vision Transformers,” in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, USA: IEEE, Jun. 2022, pp. 1204–1213. doi: 10.1109/CVPR52688.2022.01179.

[22] X. Chen, S. Xie, and K. He, “An Empirical Study of Training Self-Supervised Vision Transformers.” arXiv, Aug. 16, 2021. Accessed: Apr. 25, 2023. [Online]. Available: http://arxiv.org/abs/2104.02057

[23] P. K. A. Vasu, J. Gabriel, J. Zhu, O. Tuzel, and A. Ranjan, “FastViT: A Fast Hybrid Vision Transformer using Structural Reparameterization.” arXiv, Mar. 24, 2023. Accessed: Mar. 28, 2023. [Online]. Available: http://arxiv.org/abs/2303.14189

[24] X. Ding, X. Zhang, N. Ma, J. Han, G. Ding, and J. Sun, “RepVGG: Making VGG-style ConvNets Great Again,” in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, USA: IEEE, Jun. 2021, pp. 13728–13737. doi: 10.1109/CVPR46437.2021.01352.

[25] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, “A Simple Framework for Contrastive Learning of Visual Representations,” *Int. Conf. Mach. Learn.*, pp. 1597--1607, 2020.

# 附录

表 A1 成员分工及贡献度自评

| 姓名 | 个人分工 | 小组任务贡献度 |
| --- | --- | --- |
| 童赞嘉 | 模式识别系统整体设计 (第1节)，数据预处理方法以及算法评价指标 (第2节)，神经网络训练框架搭建 (第3节)，Mean Teacher模型以及半监督损失函数 (第3.3节)，ViT的改进和训练方案 (第6.2.1节) | 34% |
| 马宏志 | 复合变换方法以及实验方案 (第3.1节)，神经网络的1D特征导出以及特征分析 (第4节)，决策树、随机森林、AdaBoost的调参及分类 (第4.1节)，FastViT的过参数化训练方法以及重参数化部署方法 (第6.2.2节) | 33% |
| 何哲宇 | 超参数进化算法的实现以及可视化 (第3.2节)，SVM以及装袋SVM的调参及分类 (第4.2节)，SEReLU以及注意力池化的应用方案 (第6.1节)，复现SimCLR框架所实现的对比损失函数 (第6.3节) | 33% |

课程体会与建议

视网膜数据集是一个极具挑战性的数据集，因为类别不均衡问题的存在，我们探索了除了准确率之外的性能评价指标 (Kappa系数)。此外，视网膜图像的格式较为统一，不像其它自然世界的数据集会出现实例的放缩，这使得很多数据增强手段都无法在训练中使用。

除了在课程报告中所提到的工作之外，我们在训练CNN的过程中做了很多的尝试 (e.g.，使用ViT模型进行分类，使用SimCLR框架的对比学习进行预训练)，但收效甚微。医学图像数据与自然世界数据存在很大的不同，如何使用学术界中新兴的技巧提升医学图像任务的准确率是我们接下来需要研究的课题。

通过模式识别这门课，我们完成了一个完整的项目。无论是未来读研还是就业，这都是一次宝贵的项目经历。

在深度学习技术盛行的当下，大多数人的知识面受限于简单的程序调用，而忽略了深度学习所蕴含的数学原理。模式识别这门课带领我们回顾了过往最辉煌的机器学习算法，领略这些算法的强大、高明之处，这是当今浮躁的学术环境所最需要的。无论是老师的授课，还是翻转课堂的课题设置，都可以感受到老师对机器学习算法有着透彻的理解，我们在这门课中收获了许多。