



# 机器学习课程设计——FINAL答辩

小组成员：胡旭东、付哲宇、杨宇圳

# 选题



## 常规赛：PALM病理性近视预测 [报名中](#)

ISBI2019 PALM眼科挑战赛赛题再现，提供800张眼底彩照训练数据集，要求选手训练模型完成病理性近视的预测任务。


标签：疾病分类

比赛时间：2021/04/30 - 2023/01/01

举办方：



已报名

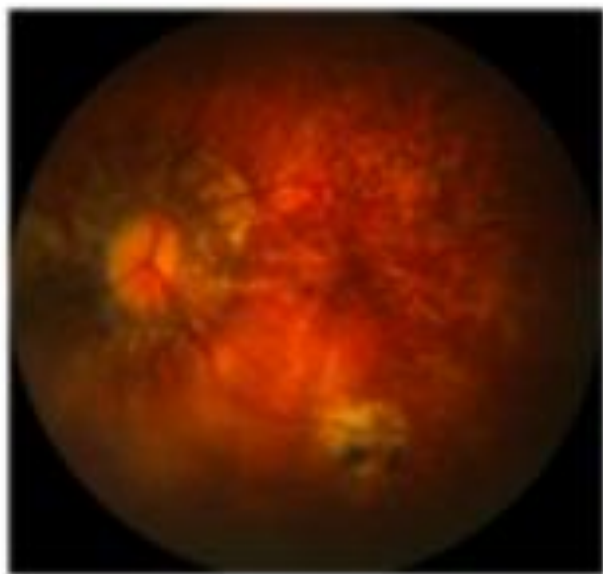
- 平台：百度 
- 题目：常规赛，PALM病理性近视预测
- 比赛简介：PALM病理性近视预测常规赛的重点是研究和发展与病理性近视诊断相关的算法。该常规赛的目标是评估和比较在一个常见的视网膜眼底图像数据集上检测病理性近视的自动算法。
- 具体任务为，将提供的图像分为病理性近视眼底彩照和非病理性近视眼底彩照，其中，非病理性近视眼底彩照包括正常眼底和高度近视眼底彩照。

# 数据集介绍

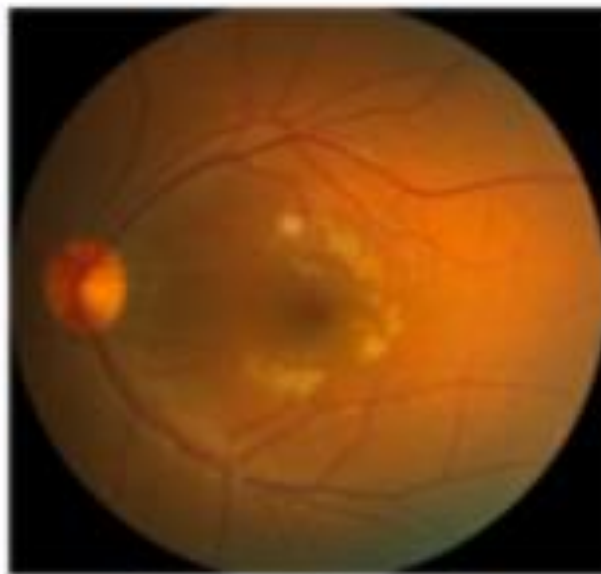
- 数据集由中山大学中山眼科中心提供800张带萎缩和脱离病变分割标注的眼底彩照供选手训练模型，另提供400张带标注数据供平台进行模型测试。图像分辨率为 $1444 \times 1444$ ，或 $2124 \times 2056$ 。标注金标准存储为BMP图像。分割图像大小与对应的眼底图像大小相同，标签如下：
  - 1、非病理性：0
  - 2、病理性：1
- 本次常规赛提供的病理性近视分类金标准是从临床报告中获取，不仅基于眼底彩照，还结合了OCT、视野检查等结果。

# 数据集样例

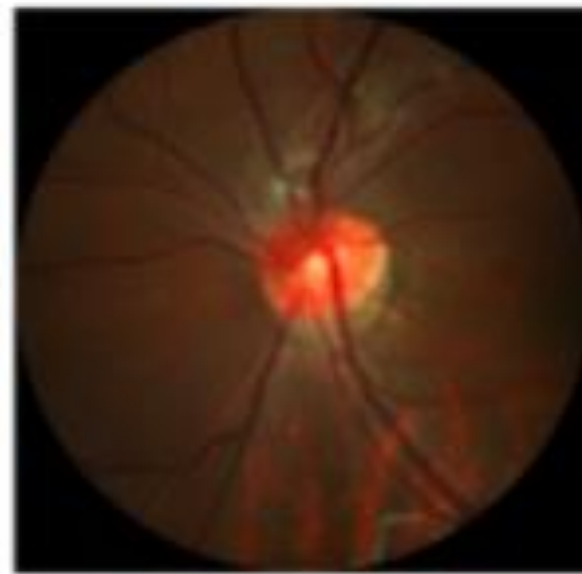
病理性近视样本



非病理性近视样本



正常



高度近视

# 评价指标

- 评价指标为AUC (Area Under Curve), 即ROC (Receiver operating characteristic) 曲线与坐标轴形成的面积。

# BASELINE基线说明

- 1. 数据加载与处理
- 2. 模型加载（高层API）
- 3. 模型预测
- 4. 保存提交结果

# 一、导入相应的包

```
import pandas as pd      # 处理xlsx文件
import os                # 文件操作
import time              # 时间记录
from tqdm import tqdm    # 进度条
import cv2 as cv         # 图像处理
import numpy as np       # 数据计算包

import paddle
from paddle import nn     # 网络层API
from paddle import optimizer # 优化器API
from paddle import regularizer # 正则化API
from paddle import metric  # 评价指标API
from paddle.nn import loss # 损失函数API
from paddle.nn import Layer # 网络层基类

from paddle.io import Dataset, DataLoader # 数据加载基类——Dataset, DataLoader——数据加载器
from paddle.vision import transforms      # 图像预处理API
```

## 二、导入数据文件，凭借完整图片路径

- 训练数据集有xlsx文件需要读取
- 测试数据要构建DataFrame表格存储文件路径信息和标签，方便后边预测数据读取与提交格式

```
Image_path = '常规赛：PALM病理性近视预测/Train/fundus_image'           # 数据存放根目录
Train_data = pd.read_excel('常规赛：PALM病理性近视预测/Train/Classification.xlsx') # 数据xlsx文件

for i in range(len(Train_data)):                                         # 将DataFrame表格中的图片补足路径
    Train_data.iloc[i, 0] = os.path.join(Image_path, Train_data.iloc[i, 0]) # 拼接路径

Train_data = Train_data.sample(frac=1.0, random_state=2021).reset_index(drop=True) # frac=1.0对应随机采样全部样本（表格数据），对应打
Train_data.head()
```



- 类似的，处理需要预测的数据，不过这时没有给定的xlsx文件，我们需要创建自定义的DataFrame表格

```
Test_data = [] # 测试图片路径(数据)
Test_path = '常规赛: PALM病理性近视预测/PALM-Testing400-Images' # 测试图片根目录
for _, _, files in os.walk(Test_path): # 获取目录下的所有图片文件
    for i in files: # 遍历文件
        Test_data.append([i, 0]) # 添加当前图片文件+一个默认标签0——以对应img, label的格式，方便预测结果存储
Test_data = np.asarray(Test_data) # 转换datatype
Test_data = pd.DataFrame(Test_data) # 转换为DataFrame表格数据
Test_data = Test_data.sort_values(by=0, ascending=True).reset_index(drop=True) # 对读取的文件排序--文件名字支持排序
for i in range(len(Test_data)): # 拼接完整图片路径
    Test_data.iloc[i, 0] = os.path.join(Test_path, Test_data.iloc[i, 0])
Test_data.head()
```

# 测试数据

	imgName	Label
0	常规赛: PALM病理性近视预测/Train/fundus_image/V0327.jpg	0
1	常规赛: PALM病理性近视预测/Train/fundus_image/V0189.jpg	1
2	常规赛: PALM病理性近视预测/Train/fundus_image/V0176.jpg	1
3	常规赛: PALM病理性近视预测/Train/fundus_image/N0116.jpg	0
4	常规赛: PALM病理性近视预测/Train/fundus_image/H0010.jpg	0

	0	1
0	常规赛: PALM病理性近视预测/PALM-Testing400-Images/T0001.jpg	0
1	常规赛: PALM病理性近视预测/PALM-Testing400-Images/T0002.jpg	0
2	常规赛: PALM病理性近视预测/PALM-Testing400-Images/T0003.jpg	0
3	常规赛: PALM病理性近视预测/PALM-Testing400-Images/T0004.jpg	0
4	常规赛: PALM病理性近视预测/PALM-Testing400-Images/T0005.jpg	0

# 训练数据

### 三、构建数据集DATASET自定义CLASS类——用于加载数据集，把数据加载函数拼接进来

- 以Train\_Dataset为例

```
def __init__(self, df, trans=None):
    super(Train_Dataset, self).__init__()
    self.df = df
    if trans is None:
        self.trans = transforms.Compose([
            transforms.Resize(size=(960, 960)),
            transforms.ToTensor(),
            transforms.Normalize()
        ])
    else:
        self.trans = trans
    self.lens = len(df)
```

```
def __getitem__(self, indexs):
    im_data, im_label = self._load_img_and_label(self.df, indexs)
    im_data = self.trans(im_data)
    return im_data, paddle.to_tensor(im_label)
```

```
def _load_img_and_label(self, df, index):
    """加载DF中的路径为图片和标签
        df: 输入DF
        index: 第几条数据
    """
    assert index < self.lens, \
        'please check the index, which has more than the dataset length!'
    im_data = cv.imread(df.iloc[index, 0], cv.COLOR_BGR2RGB) # 转为RGB数据
    im_label = int(df.iloc[index, 1]) # 标签
    return np.asarray(im_data).astype('float32'), im_label
```

```
def __len__(self):
    return self.lens
```

## • Test\_Dataset同理

```
class Test_Dataset(Dataset):  
    ...加载测试集  
    | 把数据加载函数拼进来  
    ...  
  
    def __init__(self, df, trans=None): ...  
  
    def __getitem__(self, index): ...  
  
    def _load_img_and_label(self, df, index): ...  
  
    def __len__(self): ...
```

## 四、模型基本参数设置

```
1  # 训练参数-=dict
2  Train_Paramdict = {
3      'data_length':len(Train_data),    # 数据长度
4      'train_frac':0.8,                 # 训练集比例, 原始: 0.8
5      'num_class':2,                    # 类别, 原始: 2
6      'epoches':50,                     # 训练轮次, 原始: 5
7      'batchsize':8,                    # 批量大小, 原始: 8
8      'lr':0.001,                       # 学习率, 原始: 0.005
9  }
```

## 五、训练前准备

- 划分验证集

```
1  # 数据集划分
2  Fit_data  = Train_data.iloc[:int(Train_Paramdict['data_length']*Train_Paramdict['train_frac'])]
3  Eval_data = Train_data.iloc[int(Train_Paramdict['data_length']*Train_Paramdict['train_frac']):]
```

- 加载数据集

```
1  # 数据加载
2  Fit_dataset = Train_Dataset(Fit_data)
3  Eval_dataset = Test_Dataset(Eval_data)
4
5  Fit_dataloader = DataLoader(Fit_dataset, batch_size=Train_Paramdict['batchsize'], shuffle=True)
6  Eval_dataloader = DataLoader(Eval_dataset, batch_size=Train_Paramdict['batchsize'])
```

# • 创建模型

- 利用基线--MobileNetV1，并在此基础上，选择MobileNetV2进行训练
- 后续使用残差网络ResNet训练
- 选择较为稳定的优化器、损失函数、评价指标

```
1  #创建模型
2  #model = paddle.vision.models.ResNet(num_classes=2,block=2,depth=100)
3  #model = paddle.vision.models.MobileNetV1(num_classes=2)# 使用paddle自带的基础模型进行基线测试
4  model = paddle.vision.models.MobileNetV2(num_classes=2)
5
6  model = paddle.Model(model)                # 使用高层API简化训练过程
7
8  # 优化器
9  O = optimizer.Adam(learning_rate=Train_Paramdict['lr'], parameters=model.parameters() )
10 # 损失函数
11 L = loss.CrossEntropyLoss()
12 # 评估指标--这里baseline选用精确率
13 M = metric.Accuracy()
14
15 # 预载模型训练配置
16 model.prepare(O, L, M)
```

## 六、开始训练

```
model.fit(  
    Fit_dataloader,  
    Eval_dataloader,  
    epochs = Train_Paramdict['epoches'],  
    eval_freq=1,          # 验证频率--几个轮次验证一次  
    log_freq=2,          # 日志频率--几个step输出一次训练日志信息  
    # save_dir=None,      # 如果需要保存模型，None改成路径  
    # save_freq=1,        # 保存频率--几个epoch保存一次  
)
```



## 七、模型预测

- 预测结果是一个多维概率数据

```
1 results = model.predict(Test_dataloader)
```

## 八、保存结果文件

```
results = np.asarray(results)

submit_result = []
for i in results[0]:          # 提取结果数据
    i = paddle.to_tensor(i)    # 便于使用paddle的方法
    i = paddle.nn.functional.softmax(i)      # softmax获取预测概率结果
    result = i[:, 1]           # 获取1类别对应的概率--是否病理性
    submit_result += result.numpy().tolist()  # 拼接list结果
submit_result = np.asarray(submit_result)

Test_data.iloc[:, 1] = submit_result      # 将结果数据用于修改最初的Test数据DataFrame表格中的Label项数据
Submit_data = Test_data.copy()            # 拷贝一份测试数据
Submit_data.columns = ['FileName', 'PM Risk'] # 修改表格表头，以适应提交需要
for i in range(len(Submit_data)):         # 取出原Test中的图片文件名称--不要根目录
    Submit_data.iloc[i, 0] = Submit_data.iloc[i, 0][-9:]
Submit_data.head()

Submit_data.to_csv('Classification_Results.csv', index=False, float_format="%.1f") # 保存结果csv
```

## 九、各次提交结果

- 1、MobileNetV2初始参数，简易跑，5月份第10名

10	○泡果奶+旺仔	0.98431	0.98431	2021-05-24 23:20
----	---------	---------	---------	------------------

江震雨扬	0.98431	0.98431	完成	2021-05-24 23:20
江震雨扬	0.98299	0.98299	完成	2021-05-24 23:15
江震雨扬	0.98299	0.98299	完成	2021-05-24 22:50
江震雨扬	0.97488	0.97488	完成	2021-05-18 20:28

# 目前最好成绩

- 2、MobileNetV2调参过后，5月份第4名

202105				
排名	参赛团队	Score	AUC	提交时间
	DDD	0.99879	0.99879	2021-05-29 20:39
	JavaRoom的团队	0.99877	0.99877	2021-05-29 17:21
	zhouxw的团队	0.99789	0.99789	2021-05-30 11:29
4	C++是世界上最好的语言	0.99772	0.99772	2021-05-31 20:15

• 3、ResNet50，

• 第6名

• 注：由于比赛要求每一队伍成员1名，故多次提交的参赛队伍不同。

排名	参赛团队	Score	AUC	提交时间
 1	misaka8080的团队	0.9991	0.9991	2021-06-16 17:12
 2	jsdbzcm的团队	0.99887	0.99887	2021-06-05 17:02
 3	qwerqwerqwera7的团队	0.99832	0.99832	2021-06-04 13:53
4	zhouxw的团队	0.99777	0.99777	2021-06-06 11:48
5	C++是世界上最好的语言	0.99772	0.99772	2021-06-03 17:16
6	zzzswift的团队	0.99139	0.99139	2021-06-07 22:41

- 经过数周努力将成绩做到了 5/141
- 收到了官方的审核邮件并通过了审核
- 项目已公开，链接：[飞桨常规赛：PALM病理性近视预测基本方案，2021年5月第4名 - 飞桨AI Studio - 人工智能学习实训社区 \(baidu.com\)](#)

谢谢！