

评分规则

- 1. 分赛道布置任务，同学们可以根据自己的兴趣选择赛道，每人只能选择一个赛道；
- 2. 根据每位同学在不同赛道上的成绩排名比例给分，对于选题人数少于 67 人的赛道，按照 67 人计；
- 3. 只要提交作品，就能获得排名；
- 4. 给分规则表如下

排名	5%	15%	30%	45%	65%	80%	90%	95%	100%
得分	15	14	13	12	11	10	9	8	7

各赛道排名向下取整，例如，某同学的排名为 5.9%，则视为 5%。

若各赛道排名 90% 以后的同学完成度较高，则记为 9 分。

动物分类挑战赛赛题说明

比赛内容

根据给定数据集自行搭建网络进行训练，使得网络可以对十类动物进行分类。

模板文件内已提供 train.py, test.py 和 net.py, 同学们需要：

- 在 net.py 内搭建网络 Animal_Net
- 在 train.py 中补全代码用于训练网络
- test.py 内已提供了 dataloader 和测试过程，请补全图像预处理部分及网络权重路径，保证运行 test.py 时能正确得到测试精度结果。
- 撰写一份技术报告 report.pdf，内容包括但不限于方法简述、网络结构示意图、模型在训练集和测试集的精度。

最终大家需要提交所有代码和模型权重，我们将替换 test 数据集并运行 test.py 检验模型精确度。

数据集说明

一共十种动物：

'butterfly', 'cat', 'chicken', 'cow', 'dog', 'elephant', 'horse', 'sheep', 'spider', 'squirrel'

动物类别和网络输出序号的对应关系请见 test.py 内的 label_dict。

训练集包含 24179 张图像，测试集包含 1000 张图像，每个类别 100 张；

为保证公平性，最终的测试集图像不会公开，提供给大家一个伪测试集供大家进行调试，最终测精度时会用真正测试集代替伪测试集。

数据集目录：

```
——data
    ——train
        ——butterfly
        ——cat
        .....
    ——test
    ——test_label.txt
```

如图所示，数据集下有两个文件夹 train 和 test，还有一个文本文件 test_label.txt。其中 train 包含十个文件夹，分别包含十种动物的训练图片；test 文件夹里有 1000 张图像；test_label.txt 中包含 test 文件夹内图片名称和对应类别。

数据集和代码模板下载地址：

链接：<https://pan.baidu.com/s/11iWzM87xeZGOfp0YxASxCQ?pwd=gz35> 提取码：gz35

提交要求

- 请独立搭建网络，不允许直接调用 pytorch 库内已搭建好的模型如：from torchvision import models
- 模型权重大小不要超过 100M
- 技术报告必须为 pdf 格式，务必在技术报告第一行填写个人信息姓名_学号_班级，如：张三_2023000001_未来 2301。
- 代码和权重放在一个文件夹内，文件夹命名为：姓名_学号_班级，如：张三_2023000001_未来 2301。将文件夹打成压缩包，请在 12 月 30 日之前发送到：805704731@qq.com
- 我们会对代码进行查重，抄袭者将取消本赛季成绩

成绩构成

我们会替换 test 数据集对同学们的模型进行分类精度测试，根据测试得分排名，再根据排名结果赋分。

对于不能正确输出测试精度结果的同学，我们将根据代码完成程度（网络搭建，dataloader，训练过程，测试过程）酌情给分。

提示

本次提供的图像分辨率比较大，且每张图像的分辨率也不相同，因此训练前建议把图像 resize 到比较合适的统一大小。其中可能需要用到函数：

cv2.resize(img, dsize)

- img: 输入图像
- dsize: 要 resize 的图像大小

例如，我们想把一张图像 resize 到(64, 64)大小：

img = cv2.resize(img, (64, 64))

请大家在提交前运行 test，保证能够成功输出在 test 文件夹图像的分类测试得分。

校赛博弈赛道题目

赛题说明

连六棋，也被称为 "六子棋"，是一种起源于中国的传统棋类游戏。它是五子棋的变种，但玩法更加复杂和深入。连六棋在中国的许多地方都有广泛的玩家群体，尤其在学校和家庭中，这种游戏通常被用作一种休闲和娱乐活动。

连六棋的棋盘通常是一个 15×15 的网格，与五子棋相同。游戏的目标是在棋盘上形成六个连续的同色棋子。

连六棋的规则

以下是连六棋的一些基本规则：

- 棋子的放置：一开始棋盘是空的，两名玩家轮流在棋盘上的交叉点放置棋子。一方用白色棋子，另一方用黑色棋子。
- 赢得游戏的条件：如果一名玩家在棋盘上形成了六个或更多的连续的同色棋子（无论是横、竖、斜方向），那么这名玩家就赢得了游戏。
- 禁手：不同于五子棋，连六棋没有禁手。也就是说，玩家可以在棋盘上的任何位置放置棋子，没有任何限制。
- 平局：如果棋盘被填满，没有任何一方形形成六子连线，那么游戏就是平局。

关于连六棋的更具体规则，请参考规则<https://wiki.botzone.org.cn/index.php?title=ConnectSix>

平台说明

本次大赛在 Botzone 平台上展开。请参考六子棋的具体规则<https://wiki.botzone.org.cn/index.php?title=ConnectSix>与平台格式交互<https://wiki.botzone.org.cn/index.php?title=Bot#.E4.BA.A4.E4.BA.92>。

参赛方式

参赛选手需要首先注册比赛平台Botzone账号，注册之后可以参与比赛。

1. 每支队伍参赛人数为1-2人；

2. 参赛选手报名必须保证提供的个人信息真实、准确、有效。如报名信息与参赛者个人身份不相符，取消比赛资格。

赛题：A + B

1. 题目描述

请你设计一个 Python 程序，这个程序只需实现一个功能：输入两个数字 a 和 b ，输出 $a + b$ 的值。

对于所有输入数据，保证 a 和 b 都是小于 10 的自然数。例如，输入 11，程序应该输出 2。

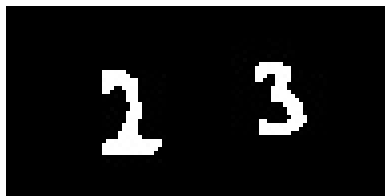
本赛题不使用 `input` 和 `print` 进行输入输出，你的程序需要从若干个文件中读取输入，并把计算结果输出到指定的文件中。

2. 文件说明

本赛题提供了一个压缩包，解压后可以得到一个文件夹，文件夹的结构如下

```
A+B
├── inputs
│   ├── 1.jpg
│   ├── 2.jpg
│   ├── ...
│   └── 10000.jpg
└── samples
    ├── explanations
    │   ├── 1.txt
    │   ├── 2.txt
    │   ├── ...
    │   └── 2023.txt
    ├── outputs
    │   ├── 1.txt
    │   ├── 2.txt
    │   ├── ...
    │   └── 2023.txt
    └── inputs
        ├── 1.jpg
        ├── 2.jpg
        ├── ...
        └── 2023.jpg
```

在 `inputs` 文件夹中，共有 10000 张图片，所有图片的像素都为 48×96 ，每张图片中都有两个手写数字，如下图所示。你的程序需要读取 `inputs` 文件夹中的每张图片中的两个数字，并计算这两个数字的和，然后把结果输出到 `outputs` 文件夹中的 `txt` 文件中（`outputs` 文件夹和 `txt` 文件需要自行创建），要求：把 `i.jpg` 的计算结果输出到 `i.txt` 中，并且 `i.txt` 中只包含一个表示答案的整数，不能出现其他任何内容。



我们在 `samples` 文件夹中提供了一些输入输出样例和对样例的解释。这个文件夹包含三个文件夹，其中，`inputs` 文件夹包含 2023 个输入样例，`outputs` 文件夹给出了针对这些样例的输出样例。另外，`explanations` 文件夹中给出了对这些样例输入的解释，共包含 2023 个文本文件，每个文本文件都有两行，每行包括用空格分隔的五个数，格式如下：

```
n1 x11 y11 x12 y12
n2 x21 y21 x22 y22
```

其中，`i.txt` 说明了 `inputs` 文件夹下的 `i.jpg` 中包含的两个数字的信息，`n1` 表示第一个数字的值，`x11` 表示第一个数字的外接正方形左上角到图片最左侧的距离，`y11` 表示第一个数字的外接正方形左上角到图片最上方的距离，`x12` 表示第一个数字的外接正方形右下角到图片最左侧的距离，`y12` 表示第一个数字的外接正方形右下角到图片最上方的距离，第二行同理，所有距离的单位都是像素。

3. 提交结果

请将你的源代码、输出文件、技术报告和模型文件（如果有）放在一个文件夹中，文件夹命名规则为 班级号_姓名_学号，然后把该文件夹压缩成一个 `.zip` 文件或 `.rar` 文件提交，文件夹的结构要求如下：

```
班级号_姓名_学号
├── codes          # 此文件夹存放所有源代码
├── outputs        # 此文件夹存放你的输出结果
│   ├── 1.txt
│   ├── 2.txt
│   ├── ...
│   └── 10000.txt
├── models         # 此文件夹存放你的模型文件（如果有）
└── report.pdf     # 你的技术报告，简要阐述你的方法，没有字数要求
```

说明：

1. 文件夹的命名规则为 班级号_姓名_学号，例如：小未的班级是 2300 班，学号是 20231301000，则文件夹应命名为 0_小未_20231301000；
2. 请严格按照命名要求对压缩包中的文件夹和文件进行命名；
3. codes 文件夹只存放源代码，不要放输入文件等其他文件；
4. 如果你有模型文件，请将模型文件放在 models 文件夹中；
5. 如果你没有模型文件，则不需要 models 文件夹，或者该文件夹为空；
6. 请将你的技术报告命名为 report.pdf，技术报告没有字数要求，但要求为 pdf 格式，你只需要在技术报告中简要说明自己使用的方法，技术报告里不需要放代码，如果必要，可以使用流程图描述算法。

请按照同样的命名规则将压缩包命名为 班级号_姓名_学号.zip 或 班级号_姓名_学号.rar，以附件形式发送到 iamtsuki@mail.dlut.edu.cn，邮件主题为 A+B_姓名。例如，小未发送的邮件主题应为 A+B_小未，附件名为 0_小未_20231301000.zip。

4. 评分规则

本赛题共有 10000 个输入数据，根据程序输出结果的正确个数进行排名，正确个数越多的选手排名越靠前，如果两个或多个选手的正确个数相同，则排名并列。最终得分根据排名进行赋分。

如果发现提交的 outputs 文件夹中的结果并非程序输出，或者抄袭他人的结果或代码，将取消本赛题的成绩。

题目四

说明一

开放性题目，课题自拟，内容不限，用 python 实现即可。

说明二

最终根据大家所做内容的创新性，工作量，完成度来评分，大家需要完成大作业报告，我们会找时间组织大家答辩。

说明三

请大家独立完成，在最后一次考试结束后，我们会发下卷子让大家描述大作业的设计思路，核心类设计，关键代码，流程图等。