

项目 I: 动物多分类实战 (建议 PyTorch 完成。不反对学员用个人喜好的其他框架。)

在开始真正的大项目前, 可以用本项目来热热身。建议对照项目 II 中的紫色、绿色部分一起看。

本项目使用兔子、老鼠、鸡组成的数据集完成三个子任务。

Stage1: 完成动物纲(Classes)分类, 预测该动物是属于哺乳纲(Mammals)还是鸟纲(Birds)。

Stage2: 完成动物种(Species)分类, 预测该动物是兔子、老鼠还是鸡。

Stage3: 完成多任务分类, 同时预测该动物的“纲”和“种”。

## 一、 素材结构:

文件夹 4 个:

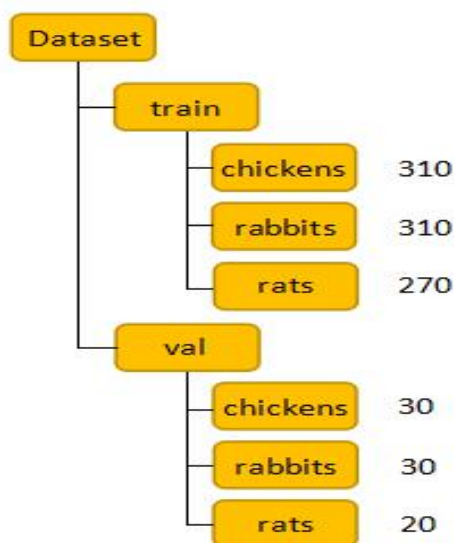
### A. Dataset: 数据集。

(1). 数据集结构如下, 数字为相应图片个数。

(2). Images-rename.py

将数据集中图片乱七八糟的名字批量重命名。

### B. Stage 1 Classes classification:



(1). Classes\_make\_anno.py

生成“纲”分类的训练与测试数据标签。选兔子和鸡作为数据集, 预测属于哺乳纲(Mammals)还是鸟纲(Birds)。

(2). Classes\_Network.py

用于“纲”分类的网络。

(3). Classes\_classification.py

程序主体, 用来进行模型训练/验证, 并调用训练好的模型进行预测。

(4). Classes\_train\_annotation.csv/Classes\_val\_annotation.csv

用于“纲”分类的标签。由 Classes\_make\_anno.py 生成。

### C. Stage 2 Species classification:

- (1). Species\_make\_anno.py  
生成“种”分类的训练与测试数据标签。预测是兔子、老鼠还是鸡。
- (2). Species\_Network.py  
用于“种”分类的网络。
- (3). Species\_classification.py  
程序主体，用来进行模型训练/验证，并调用训练好的模型进行预测。
- (4). Species\_train\_annotation.csv/Species\_val\_annotation.csv  
用于“种”分类的标签。由 Species\_make\_anno.py 生成。

### D. Stage 3 Multi-classification

- (1). Multi\_make\_anno.py  
生成同时对“纲”、“种”进行分类的训练与测试数据标签。同时预测是兔子、老鼠还是鸡，哺乳纲还是鸟纲。
- (2). Multi\_Network.py  
用于“纲”、“种”多分类的网络。
- (3). Species\_classification.py  
程序主体，用来进行模型训练/验证，并调用训练好的模型进行预测。
- (4). Multi\_train\_annotation.csv/Multi\_val\_annotation.csv  
用于“纲”、“种”多分类的标签。由 Multi\_make\_anno.py 生成。

## 二、 主要思路：

### 1. 数据处理

实际应用中很多深度学习的数据集需要自己来建。本项目中的数据图片是从网上爬来的，这些图片需要被统一命名。为了得到训练标签，针对不同的任务生成相应的标签集。如：Stage1 中需要将哺乳纲、鸟纲的数据分别标为 0, 1 作为训练标签；Stage2 中将兔子、老鼠、鸡的数据分别用 0, 1, 2 作为标签。

### 2. 读取数据

定义数据的变换方式及顺序，train 和 test 可不同。主要用到 torchvision.transforms 的 transforms 系列，以及 torch.utils.data 中 DataLoader 函数。可以查查除了 Dataloader 外还有什么加载数据的方式。

定义数据组成，如：

Stage1 中我们的数据是 sample = {'image': image, 'classes': label\_classes};

Stage2 中我们的数据是 sample = {'image': image, 'species': label\_species};

Stage3 中我们的数据是 sample = {'image': image, 'classes': label\_class, 'species': label\_species}

### 3. 数据验证

在训练之前需要验证我们的数据集与标签集是否处理正确。可任意输出数据图片与对应标签，验证是否符合我们的设定。

### 4. 搭建网络

搭建网络时注意课上提到的 Function layer 的选择以及安放位置，可根据训练效果自行调整。注意每层网络维度的变化关系，要做到自洽。Stage1 和 Stage2 都是单任务分类，网络只需在 FCLayer 后接一个分类器即可。Stage3 是多任务分类，我们在 FC 后接两个分类器，一个用来对“纲”分类，另一个对“种”分类。

#### 5. 训练、测试网络

在训练集中训练一次，在测试集中测试一次，分别记录 loss 和 accuracy，根据变化趋势来判断训练是否有效，并根据测试集中的 accuracy 记录最优模型。Stage1、Stage2 单任务训练时，loss 就是单一分类任务的 loss；Stage3 多任务训练时，将每个任务的 loss 进行线性加权作为训练的 loss。考虑有哪些 loss 函数可以使用？除了 SGD 外还有哪些训练方式可以使用？

#### 6. 调参

可调整 Lr, momentum 等参数对比训练结果，找到最优方案。

#### 7. 可视化预测结果

用训练好的模型对一些数据进行预测，直观地看看训练好的预测器效果如何。