

## 实验四 基于 OpenGADL 软件的迁移学习

### 一、实验目的

通过对自标注数据集进行迁移学习实验，进一步理解深度学习的实际应用；掌握从数据标注到模型选择再到模型训练测试的深度学习全流程工程。

### 二、实验内容

1. 使用 OpenGADL Data 软件对提供的猫狗数据集中的**测试集**进行分类任务的数据标注；（训练集和验证集已完成标注）
2. 选用 OpenGADL Model 提供的预置模型对数据集进行迁移学习；
3. 对拟合后的模型进行测试。

### 三、实验步骤

示例：使用预置模型 Resnet18，在自标注的猫狗数据集进行训练验证和测试。

1. 选择猫狗数据集中的测试集，使用 OpenGADL Data 软件进行标注；
2. 整理数据集：将训练集 train，验证集 val 以及测试集 test 按照 OpenGADL 数据集标准格式进行整理；【数据集的格式见附录】

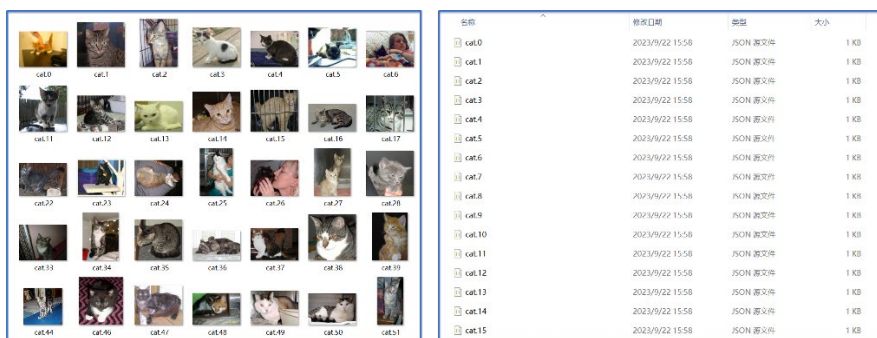


图 1 数据集部分图片 image 和标签 annotation

3. 打开 OpenGADL Model 软件，新建一个项目，命名为 Resnet18DogCat；选择任务类型为“图像分类”，并且选择项目保存路径；



图 2 新建一个项目

#### 4. 切换界面到【模型训练】，进行配置

1. 选择模型路径：选择预置模型的情况下，模型文件夹不需要选择，由软件自动生成；
2. 选择数据集：本例使用自标注的猫狗数据集 DogCat；点击“使用用户数据集”按钮，点击打开文件按钮，选择数据集路径；软件将通过这个路径去自动匹配目录下的 train 和 val 数据集；
3. 使用预训练模型：本例使用软件预置模型，因此点击按钮“使用系统提供的预训练模型”，下拉框中依次选择“resnet”，“18”；
4. GPU 加速：本例使用单张 GPU 进行训练，因此点击按钮“使用单个 GPU 训练”；
5. 输入尺寸：根据数据集图片大小，选择一个合适的大小  $224 \times 224$ ；
6. 训练参数设置界面：配置迭代轮次数为 20，批量大小（Batch Size）为 64（取决于 GPU 显存大小）；图像均值和方差全设为 0 交由软件根据数据集自动计算；优化器 Optimizer 选用 Adam 算法（参数默认）；
7. 配置完成后，点击“训练”按钮，开始训练；
8. 训练结束后，在当前项目文件夹/project/train 得到一个单次训练的文件；

参数配置

选择模型路径

\Resnet18DogCat\project\models\resnet18

选择数据集

使用公开数据集

CIFAR10

使用用户数据集

D:\Datasets\DogCat\_

选择预训练模型

不使用预训练模型

使用系统提供的预训练模型

模型

resnet

version

18

使用用户提供的预训练模型

GPU加速

使用CPU训练

使用单个GPU训练

GPU:0

使用多个GPU并行训练

输入尺寸

width(px):

224

height(px):

224

自动化训练

训练参数设置

数据预处理

迭代轮次

20

批量大小

64

保存间隔

5

图像均值

0.484

0.449

0.41

训练

停止

图 3 训练界面参数配置

3

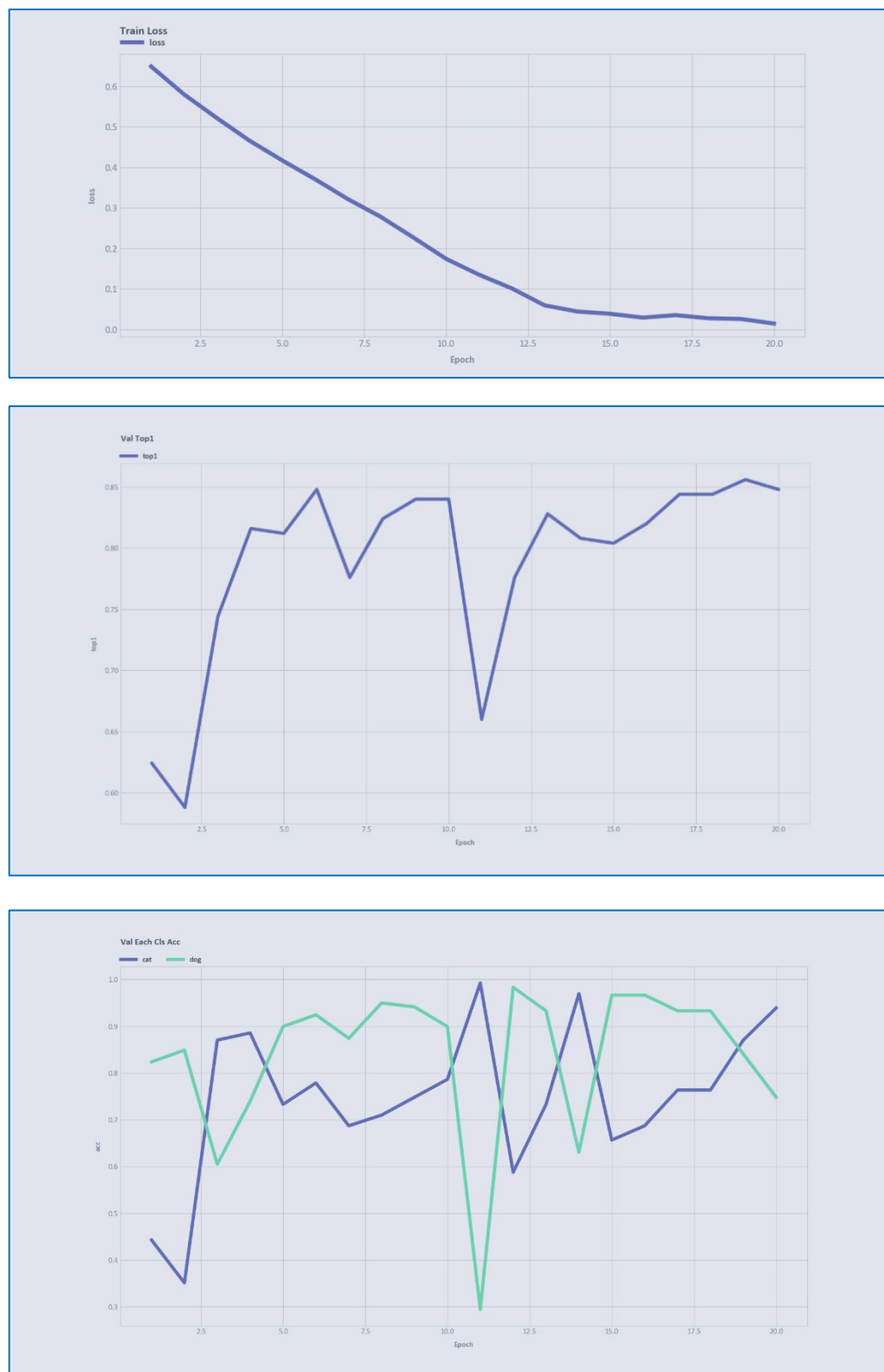


图 4 (a) (b) (c) 训练损失指标图、验证集准确率以及验证集每个类别准确率指标图

## 5. 切换界面到【评估部署】，进行配置

1. 选择模型：选择要进行测试的模型文件夹；软件自动会生成预置模型

的模型文件夹，存放在当前项目文件夹/project/models/下

2. 选择权重文件：软件会自动加载模型文件夹下的.pkl/.pth 文件，选择刚刚训练好的权重文件；
3. 选择数据：本例使用自标注的数据集，因此点击按钮“选择数据集”，选择猫狗数据集的路径；
4. GPU 加速：本例使用单张 GPU 进行测试，因此点击按钮“使用单个 GPU 测试”；
5. 测试参数配置界面：输入尺寸，图像均值，图像方差应该与训练界面的参数一致，才能得到正确的测试结果；【软件在训练后默认自动将测试界面的这几个参数与训练界面的这几个参数进行同步】；
6. 配置完成，点击“测试”，开始测试；
7. 测试结束后，将以表格形式显示测试结果，并在下方显示数据集的测试报告。

选择模型

D:\OpenGADL Model\Project\Resnet18Dog

选择权重文件

resnet18\_DogCat\_GADL\_best\_trained.pkl

选择数据

☐ 选择一张图片

☒ 选择数据集

D:\Datasets\DogCat\_GADL

GPU加速

☐ 使用CPU测试

☒ 使用单个GPU测试 GPU:0

☐ 使用多个GPU并行测试

批量大小

64

输入尺寸

width(px): 224

height(px): 224

图像均值

0.48

0.45

0.41

图像方差

0.23

0.22

0.22

模型部署

☒ 否

☐ 是

选择部署平台

NVIDIA GPU

☐ 模型压缩

量化 Not Use 比特 3

剪枝 Not Use 比例 0.0

测试

停止

图 5 测试界面配置参数

Id	File	X_image	y_true	y_predict
1	cat.45.jpg		cat	cat
2	cat.249.jpg		cat	cat
3	cat.287.jpg		cat	cat
4	cat.421.jpg		cat	cat
5	cat.663.jpg		cat	cat

首页

上一页

1

下一页

尾页

总页数 1

前往

页

前往

Overall Indicators

top1

0.96296

precision

0.96314

recall

0.96296

f1\_score

0.96290

AUC

0.96084

Categories Indicators

class

precision

recall

f1\_score

cat

0.95699

0.97802

0.96739

dog

0.97101

0.94366

0.95714

图 6 (a) 测试结果表格展示; (b) 测试数据集测试报告

## 附录

OpenGADL 数据集标准格式:

1. 划分为三个部分, 分别为标签文件夹“**annotations**”, 配置文件夹“**config**”以及保存所有数据集图片路径的“**image.txt**”文件。
2. **annotations** 文件夹中存放了所有图片的标签 json 文件, 包括训练集 train、验证集 val 和测试集 test。【测试集的标签 json 文件需要手动添加, 如图 7 所示, 将在 GADL\_Data 中完成标注后的测试集标签文件添加到标准数据集中】

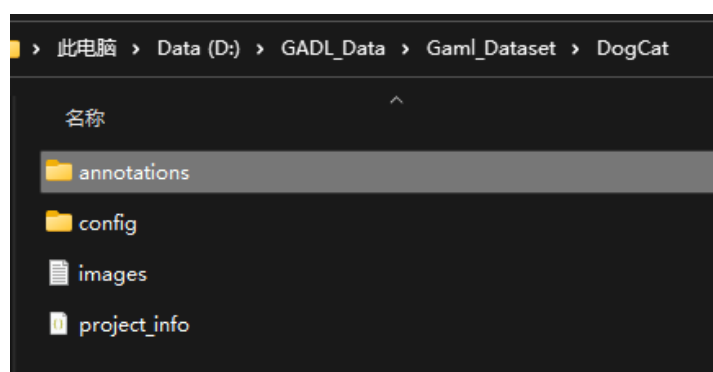


图 7 GADL\_Data 标注完成后的 annotations 文件位置

3. config 文件夹中存放了 categories.txt、test.txt、train.txt 和 val.txt 文件。其中 categories.txt 文件存储标签类别与 id 的映射字典, 如图 8 所示; 【注意: 标签类别 id 必须从 0 开始】test.txt、train.txt 和 val.txt 文件分别存储了测试集、训练集和验证集的图片路径。【存储的图片路径为绝对路径, 图片所在位置一旦改变了, 相应的路径需要修改!!!】



图 8 categories.txt 文件内容



4. image.txt 文件存储了该数据集中所有图片的绝对路径，如图 9 所示。【存储的图片路径为绝对路径，图片所在位置一旦改变了，相应的路径需要修改!!!】

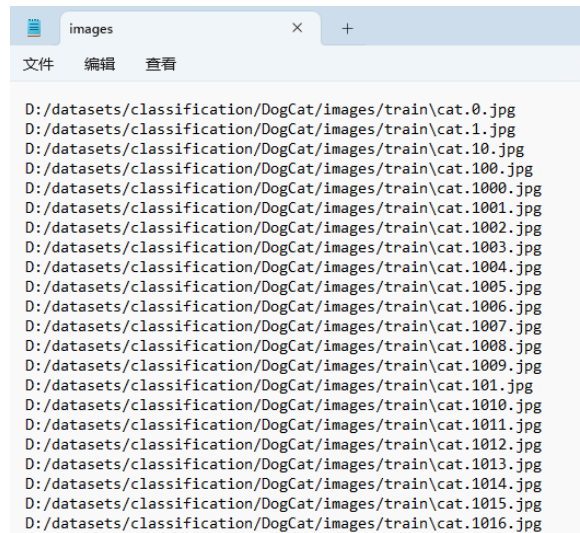


图 9 image.txt 文件内容

5. 标准数据集格式示例如下：

```
-- annotations
  --001.json
  --002.json
  ...
-- config
  --categories.txt
  --test.txt
  --train.txt
  --val.txt
-- images.txt
```

图 10 数据集文件夹结构