

图像实验说明

1. 实验任务

任务一：基于传统机器学习方法（如 HOG+SVM 分类方法），进行 19 类交通标志分类。



任务二：基于深度学习方法，在与任务一相同的数据上进行 19 类交通标注分类；

任务三：在额外给定的 11 类交通标志数据上进行单样本分类，11 类小样本类别每类提供 1 张有标注图片，且这 11 类不能引入额外有标注数据，要求模型学会区分这 11 类，并给出测试集的 11 类分类结果。



任务四（扩展）：在额外给定的交通标志检测数据集上，实现一个交通标志检测模型检出所有交通标志，并利用任务一或二中训练的得到的分类器，对检测到的交通标志进行分类，给出检测与识别的结果。

2. 数据格式说明

所有任务的数据均放置于天津院服务器的助教账号下: `/home/ass02/image_exp/`，大家可以在天津院服务器直接访问拷贝。

2.1 任务一至三（分类）：

1. 数据集 Classification/Data/

用于任务一和任务二：19 类交通标志分类。

Classification/Data/Train/下 19 个文件夹，每个文件夹以类别名命名，包含该类的训练数据；

Classification/Data/Test/下为所有测试图片，没有类别标注；

Classification/Data/train.json 是训练数据的标签文件，格式为{文件名:类别}，如 {'i2_0000.png':'i2', 'i4_0000.png':'i4'};

Classification/Data/test.json 格式与 train.json 完全相同，但类别标签为随机指定，要求同学利用本文件调通测试接口 test.py 方可上交测试结果。

注：可使用 Python 中的 json 标准库进行 json 文件的读写。

2. 数据集 Classification/DataFewShot/

用于任务三：11 类小样本交通标志分类。

数据格式与 1 类似。

3. 测试接口 Classification/TestCode/test.py

- (1) 对所有测试数据预测类别标签，并保存成与 test.json 相同的格式；
- (2) 使用命令"python test.py --predfile pred.json（所有训练数据的预测结果） --labelfile test.json"，也可自行划分验证集，并利用该接口得到验证集准确率；
- (3) 要求利用 test.json 调通测试接口 test.py，将预测结果提交给助教验收，得到测试集准确率，每个小组有 2 次测试机会（后期会在网络学堂开放提交窗口，禁止人工对测试集进行标注，如发现提交的代码及模型不能复现分类结果，将按作弊处理，任务四同）。

2.2 任务四（检测）

检测数据集一共包含 5588 张图像，13297 个交通标志实例（交通标志的数量）。其中，训练集包含 3690 张图像，8799 个交通标志实例；测试集包含 1898 张图像，4498 个交通标志实例。包含的交通标志一共 19 类，类别与任务一、二中的类别相同。

示例图像如下，图像中的交通标志已被标出：



1. 数据集 Detection/:

train/下包含所有 3690 张图像训练图像。

test/下包含所有 1898 张测试图像。

train_annotations.json 为训练集上的图像标注，

2. train_annotations.json 结构说明:

利用 json 库读取标注文件, 并转化为字典对象后, 字典对象 key 与对应的 value 分别为:

imgs: 包含所有图像标注的字典对象, 所有图像的 id (图像文件名) 为该字典的 key. 每个图像 id 的 key 的 value 也为一个字典对象, 包含单张图像的信息:

id: 图像 id

path: 对应图像的相对路径

object: 该图像的所有交通标志标注, 以列表方式存储。列表中每一个对象均为一个字典对象, 包含一个交通标志的标注, 其 key 与对应的 value 分别为:

category: 交通标志类别

bbox: 包含交通标志外接框的左上角(xmin, ymin)和右下角点坐标(xmax, ymax)的字典对象。

ellipse_org: 可选标注, 表示交通标志的边界点。

ellipse: 可选标注, 表示交通标志的椭圆掩膜标注, [x_o, y_o]为椭圆中心, [l_s, r_s]为长半径与短半径, 剩下一个值为椭圆的角度。

polygon: 可选标注, 多边形掩膜的边界点。

例如 annos 为读取 json 标注得到的字典, annos['imgs'][imgid]的值为

```
{
  u'id': 78307,
  u'objects': [
    {
      u'bbox': {u'xmax': 1116.309, u'xmin': 1097.11, u'ymax': 987.6815, u'ymin': 968.116},
      u'category': u'p5',
      u'ellipse': [[1106.3541259765625, 977.7344970703125], [18.043243408203125, 20.23980903625488
3], 122.83193969726562],
      u'ellipse_org': [[1111.86, 970.323], [1097.9, 974.499], [1101.05, 985.312], [1106.48, 987.428], [1116.
38, 977.588]]],
      u'path': u'test/78307.jpg'
    }
  ]
}
```

3. 测试接口

请将测试结果同样保存为 json 格式, 格式类似于训练标注, 但仅需要对于每个交通标志, 我们仅需要给出'bbox', 'category'. 'score'即可; 同时也不需要 path 与 id 对应字段。

例如 res 为读取 json 格式的测试结果得到的字典, res ['imgs'][imgid]的值为

```
{
  u'objects': [
    {
      u'bbox': {u'xmax': 1116.23, u'xmin': 1097.11, u'ymax': 987.68, u'ymin': 968.11},
      u'category': u'p5',
      u'score': 0.999999,
    }
  ]
}
```

检测任务测试脚本将之后在网络学堂放出。

3. 作业提交

提交作业时，需包含程序源代码及模型，测试结果以及实验报告。例如，提交的文件结构为：

- 张三_李四_王五_交通标志识别.zip
 - Codes/ **#程序源代码及模型**
 - Results/ #不同任务下测试数据的预测结果
 - 实验报告.pdf

每小组提交一份即可，并用独立段落说明成员贡献。数据无需提交。