

实战案例6：行人检测

作者：Robin

日期：2018/09

提问：[小象问答](#)

参考：[kaggle](#)

声明：[小象学院](#)拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散布。任何其他人或机构不得盗版、复制、仿造其中的创意，我们将保留一切通过法律手段追究违反者的权利

1. 案例描述

训练一个分类器可以用于行人的检测，即将图片中的行人用矩形框表示出来。该项目需要安装 `imutils` 模块，`pip install imutils`

2. 数据集描述

该项目的训练集包含两部分：

- **pos**, 正样本图片数据，包含行人的图片
- **pos**, 负样本图片数据，不包含行人的图片

3. 任务描述

- 使用机器学习方法进行行人检测

4. 主要代码解释

- 代码结构

```
lect06_proj
├── data
│   ├── pos    # 正样本图片目录
│   ├── neg    # 负样本图片目录
│   └── test_image # 测试图片目录
├── feats # 图片特征存放目录
├── models # 训练模型存放的目录
├── main.py    # 主程序
├── utils.py   # 工具文件
├── config.y   # 配置文件
└── lect06_proj_readme.pdf # 案例讲解文档
```

- **main.py**

整个项目可分为3个部分，特征提取，模型训练及检测

```
def main():
    ...
    # 特征提取
    utils.extract_feats()

    # 读取提取的HOG特征，用于训练svm模型
    utils.train_svm()

    # 对指定目录中的图片进行行人检测
    utils.detect_person_in_path('./data/test_image')
    ...
```

- **utils.py**

在进行行人检测时，对图片进行不同尺度的缩放，然后使用“滑动窗口”对每个窗口检测是否有行人出现

```
def detect_person(filename):
    ...
    for im_scaled in pyramid_gaussian(im, downscale=downscale):
        ...
        for (x, y, im_window) in sliding_window(im_scaled, min_win_size, step_size):
            ...
    ...
```

- **utils.py**

为了将多个尺度下的检测结果进行合并，可以使用 `non_max_suppression` 方法。

```
def detect_person(filename):
    ...
    non_max_suppression(rects, probs=scores, overlapThresh=0.3)
    ...
```

5. 案例总结

- 该项目通过使用机器学习进行图像中的行人检测，包含了如下内容：
 - 图像数据操作
 - 图像特征提取
 - 模型训练
 - 滑动窗口

6. 课后练习

- 课后学员可以更改项目中的参数，比如HOG特征的参数，阈值的选择等，观察对检测结果的影响。

参考资料

1. [OpenCV Python教程](#)