

---

# 大作业——交通标志识别

---

郭一隆

June 25, 2018

## CONTENTS

<b>1 问题背景</b>	<b>3</b>
1.1 数据集 . . . . .	3
1.1.1 GTSDB 原始数据集 . . . . .	3
1.1.2 GTSDB 子集 . . . . .	4
1.1.3 GTSDB 目标检测数据集 . . . . .	4
<b>2 问题描述</b>	<b>5</b>
2.1 GTSDB 子集 (12 类) 识别 . . . . .	5
2.1.1 PCA or Fisher . . . . .	5
2.1.2 HOG + SVM . . . . .	5
2.2 GTSDB 完整集 (4 大类) 识别 . . . . .	5
2.3 GTSDB 目标检测集目标检测 + 识别 . . . . .	5
<b>3 问题求解</b>	<b>6</b>
3.1 PCA+KNN 12 类识别 (题目 1) . . . . .	6
3.1.1 系统框图 . . . . .	6
3.1.2 思路简述 . . . . .	6
3.1.3 实验结果 . . . . .	7
3.1.4 结果分析 . . . . .	9
3.2 PCA+KNN 开集测试 (题目 3) . . . . .	9
3.2.1 思路简述 . . . . .	9
3.2.2 实验结果 . . . . .	9
3.2.3 结果分析 . . . . .	10

3.3	HOG+SVM 12 类识别 (题目 2) . . . . .	12
3.3.1	系统框图 . . . . .	12
3.3.2	原理简述 . . . . .	13
3.3.3	实验结果 . . . . .	13
3.3.4	结果分析 . . . . .	15
3.4	HOG+SVM 开集测试 (题目 3) . . . . .	15
3.4.1	思路简述 . . . . .	15
3.4.2	实验结果 . . . . .	15
3.4.3	结果分析 . . . . .	17
3.5	HOG+SVM 4 大类识别 (选做 1) . . . . .	17
3.6	RCNN 检测 + 识别 (选做 2) . . . . .	19
3.6.1	系统框图 . . . . .	19
3.6.2	原理简述 . . . . .	19
3.6.3	实验结果 . . . . .	19
3.6.4	结果分析 . . . . .	22

4 代码说明 23

# 1 问题背景

## 1.1 数据集

数据集包括用于分类识别的纯交通标志图像集，以及用于区域检测的含交通标志的场景图像集。

### 1.1.1 GTSDB 原始数据集

- German Traffic Signs Detection Benchmark
- 43 小类交通标志，分别属于 4 大类 (prohibitory, danger, mandatory, other)

Table 1: GTSDB 类别

类别	含义	大类
0	限速 20	prohibitory
1	限速 30	prohibitory
2	限速 50	prohibitory
3	限速 60	prohibitory
4	限速 70	prohibitory
5	限速 80	prohibitory
6	解除限速	other
7	限速 100	prohibitory
8	限速 120	prohibitory
9	禁止超车	prohibitory
10	禁止卡车超车	prohibitory
11	优先通过	danger
12	主路	other
13	让行	other
14	停车让行	other
15	禁止通行	prohibitory
16	禁止卡车通行	prohibitory
17	禁止驶入	other
18	危险	danger
19	左弯	danger
20	右弯	danger
21	连续弯道	danger
22	路面不平	danger
23	路面湿滑	danger
24	道路变窄	danger
25	道路施工	danger
26	注意信号灯	danger
27	注意行人	danger

28	注意学校	danger
29	注意自行车	danger
30	注意雪天	danger
31	注意动物	danger
32	限制解除	other
33	右转	mandatory
34	左转	mandatory
35	直行	mandatory
36	直行右转	mandatory
37	直行左转	mandatory
38	靠右行驶	mandatory
39	靠左行驶	mandatory
40	环岛	mandatory
41	超车限制解除	other
42	卡车超车限制解除	other

- 共 1213 张交通标志图像
- 图像规模: RGB, 16x16 ~ 100x100 不等
- Location: source/

### 1.1.2 GTSDB 子集

- 上述数据集的子集, 包含了类别: 1,2,4,5,6,7,10,11,12,13,18,38, 共 12 小类
- 另外选取了 150 张其他小类的交通标志图像作为负样本
- 其中 600 张作为训练集, 265 张作为测试集
- Location: data/

### 1.1.3 GTSDB 目标检测数据集

- 900 张日常道路拍摄图像
- 图像规模: RGB, 1360x800
- 1213 个标定的检测目标 (交通标志)
- Location: Origin/

## 2 问题描述

### 2.1 GTSDB 子集（12 类）识别

#### 2.1.1 PCA OR FISHER

- 用 PCA 方法或 Fisher 线性判别准则的方法对 GTSDB 子集 12 类进行识别
- 图像特征可以采用灰度图像值等
- 采用 KNN 分类，分析选取不同的主分量个数，对识别率和虚警率的影响
- 进行开集测试

#### 2.1.2 HOG + SVM

- 使用 HOG 特征提取 +SVM 分类方法对 GTSDB 子集 12 类进行识别
- 分析评价该方法的性能
- 进行开集测试

### 2.2 GTSDB 完整集（4 大类）识别

- 用前述任意方法在 GTSDB 完整集上进行 4 大类识别
- 分析结果

### 2.3 GTSDB 目标检测集目标检测 + 识别

- 在 GTSDB 目标检测集上进行检测 + 识别
- 分析结果

### 3 问题求解

#### 3.1 PCA+KNN 12 类识别 (题目 1)

##### 3.1.1 系统框图

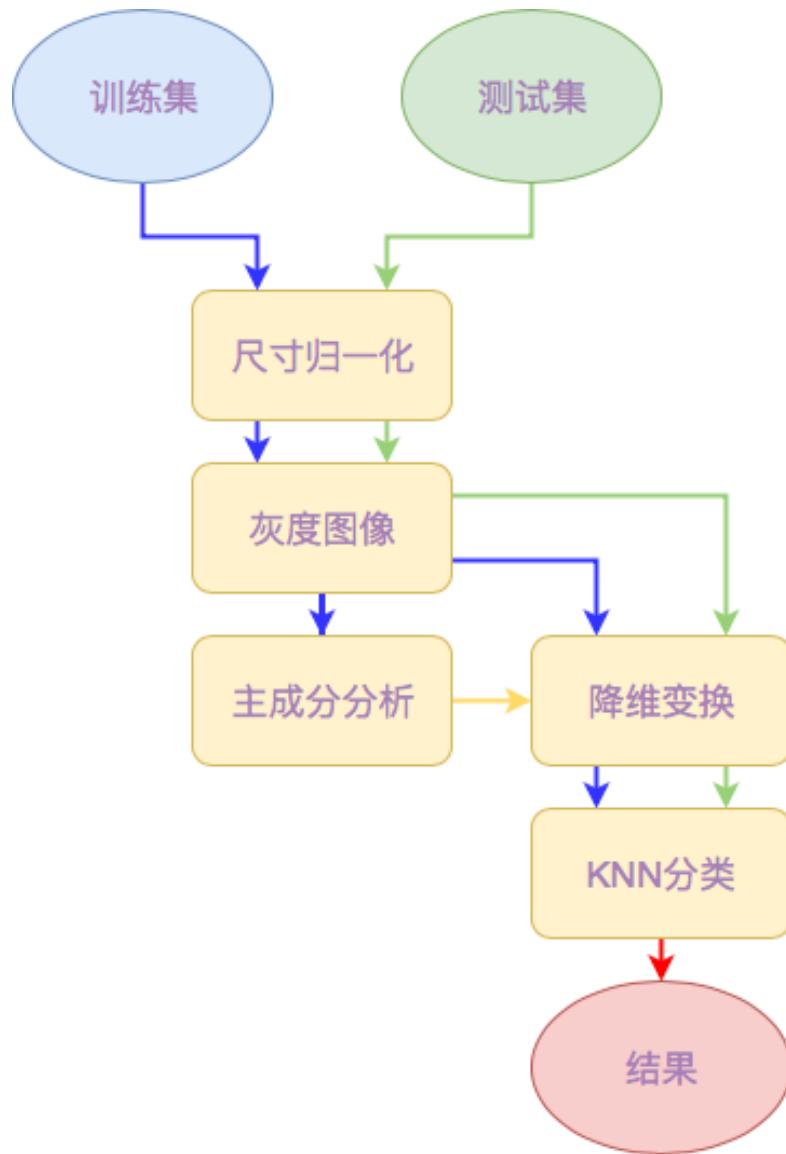


Figure 1: PCA+KNN 识别流程图

##### 3.1.2 思路简述

- 要使用 PCA 对样本特征进行分析，必须保证样本特征维数一致。这里采用 `imresize` 至同一尺度，也可以采用其他特征提取方法，如直方图等。

- 由于训练样本容量为 600，而 PCA 较好的应用场景是**样本容量 > 特征维度**，因此 `imresize` 的尺度不宜太大，并通过转为灰度图像进一步减小特征维度。
- PCA 的思路是通过保留输入样本的部分主要成分来降低数据维度，并能在一定程度上抑制噪声的影响。对于测试集，要进行与训练集相同的降维变换。
- 在 PCA 降维后的空间内进行 KNN 分类。
- 比较保留不同能量占比的主成分、不同的 K 值对识别性能的影响。

### 3.1.3 实验结果

比较不同 PCA 能量、不同 K 值下，总识别率（图2）与总虚警率（图3）的变化。

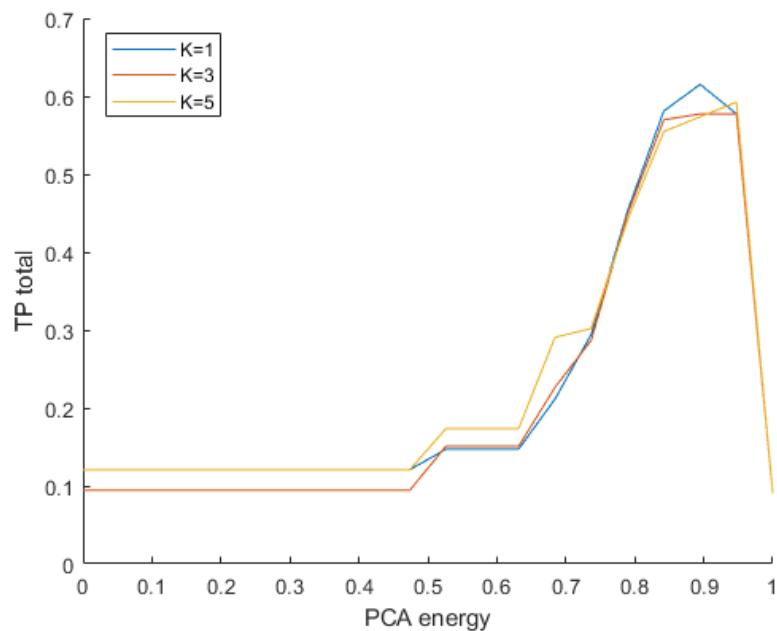


Figure 2: 总识别率随 PCA 能量的变化

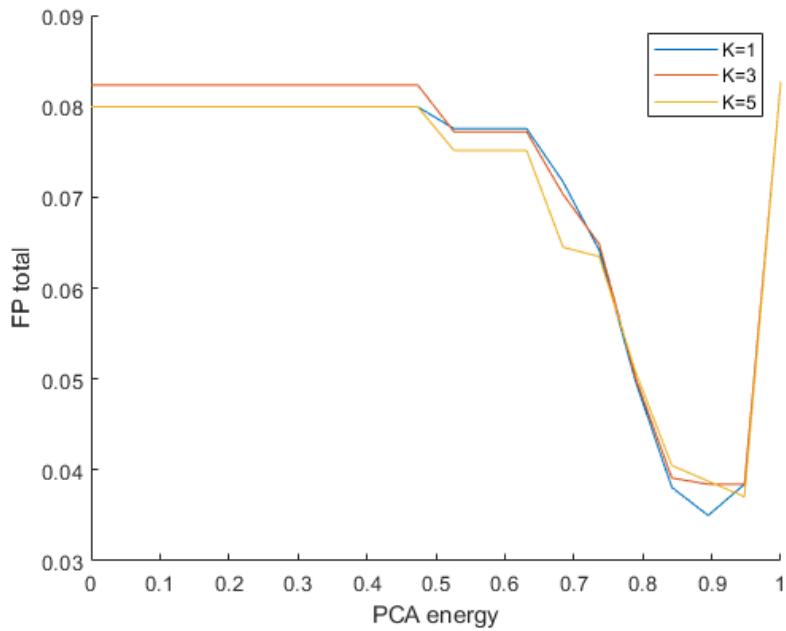


Figure 3: 总虚警率随 PCA 能量的变化

取能量阈值 =90%, K=1 时, 得到

$$P_{i,j} = N_{i,j} / \sum_{j=1}^c N_{i,j}$$

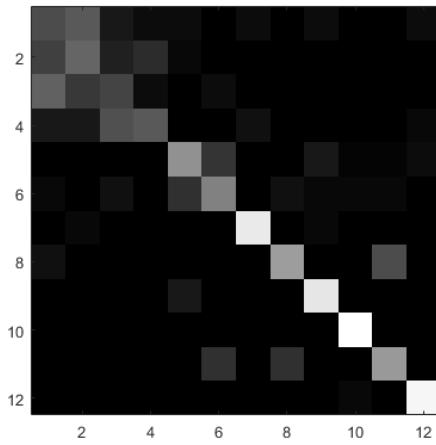


Figure 4: P 矩阵可视化

### 3.1.4 结果分析

- 保留 90% 能量主成分时，在闭集测试上性能最优，总识别率与总虚警率均达到峰值。
- KNN 分类中 K 值的选取对于本模型下识别性能的影响不大。
- 保留 100% 的主成分并不能达到更好的识别效果，因为此情况下不能很好地抑制次要成分（如噪声）的干扰。
- 仅保留少量（如 10%）主成分时，分类器近似于一个随机猜测器，识别率与虚警率均接近  $\frac{1}{12} \approx 0.083$ 。

## 3.2 PCA+KNN 开集测试（题目 3）

### 3.2.1 思路简述

- 需对给定的负样本进行测试，考虑在 KNN 算法中设定一定的距离阈值来拒识。
- 在 KNN 算法中，若最近的 K 个距离中有大于阈值的距离，则直接抛弃该距离，剩余的（小于等于 K 个）距离中按训练集进行投票。
- 距离阈值的具体值与 PCA 分量个数、归一化手段、K 值均有关。

### 3.2.2 实验结果

以 PCA 能量 =90%，K=1 为参数，观察不同阈值对总识别率与总虚警率的影响。

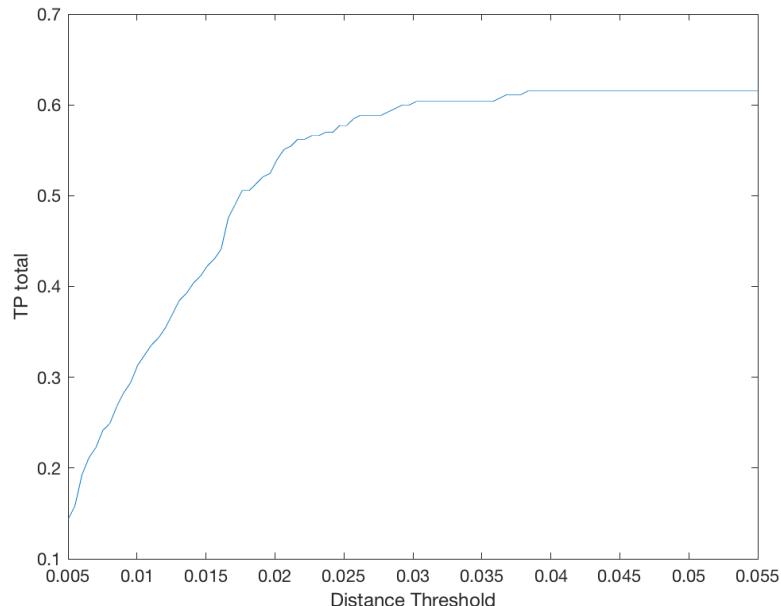


Figure 5: 总识别率随距离阈值的变化

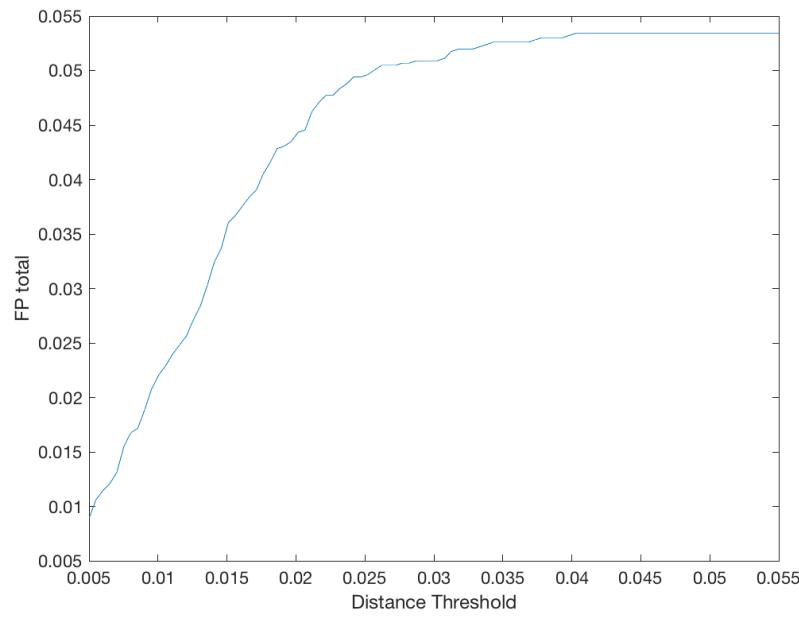


Figure 6: 总虚警率随距离阈值的变化

### 3.2.3 结果分析

- 当距离阈值超过一定范围后，阈值拒识失效，即所有 K 距离均小于该阈值。
- 整体来看，距离阈值越大，总识别率与总虚警率越高。当阈值过大时，拒识失效，等价于无阈值情形。

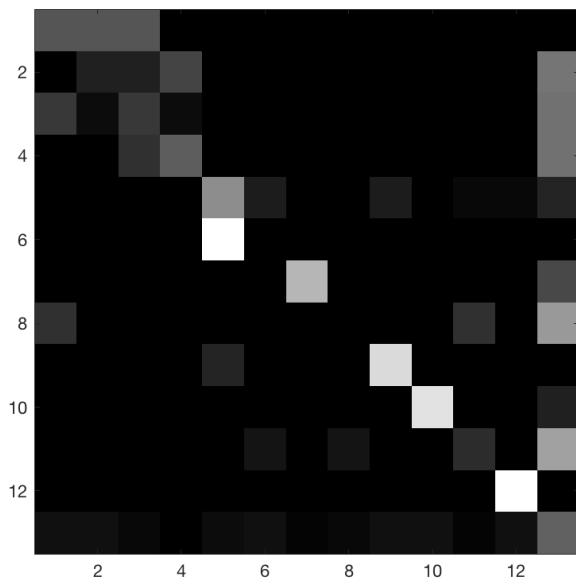


Figure 7: 当距离阈值设为 0.0085 时， $P$  矩阵的可视化。对于负样本来说，可以保证一定的拒识率 (0.39)，但不可避免地同时拒识了正测试样本。

### 3.3 HOG+SVM 12 类识别（题目 2）

#### 3.3.1 系统框图

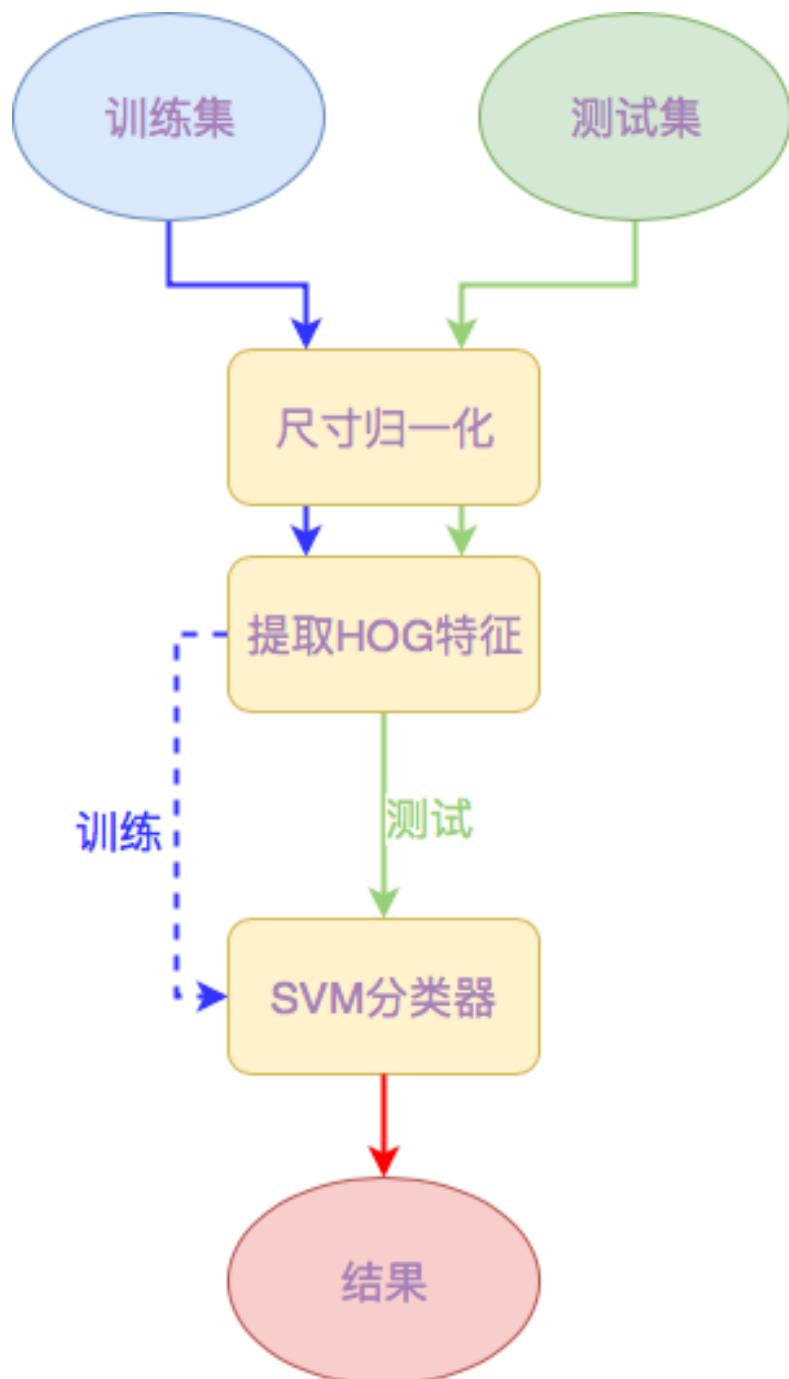


Figure 8: HOG+SVM 识别流程图

### 3.3.2 原理简述

- 对尺寸归一化后的图像进行 HOG 特征提取，获得图像的 HOG 特征空间表示。
- 由于是多类分类问题，而 SVM 适用于 2 类识别问题，需要构造一组 SVM，每个 SVM 只处理一种两类识别问题。
- 用 Error-correcting Output Codes 的方式构造一组 SVM，对于 12 类识别问题，共有  $12 \times 11 / 2 = 66$  个 SVM。
- 参考：<https://ww2.mathworks.cn/help/stats/fitcecoc.html>

### 3.3.3 实验结果

- HOG 采用默认参数：CellSize: 8x8, BlockSize: 2x2
- 图像尺寸归一化：32x32

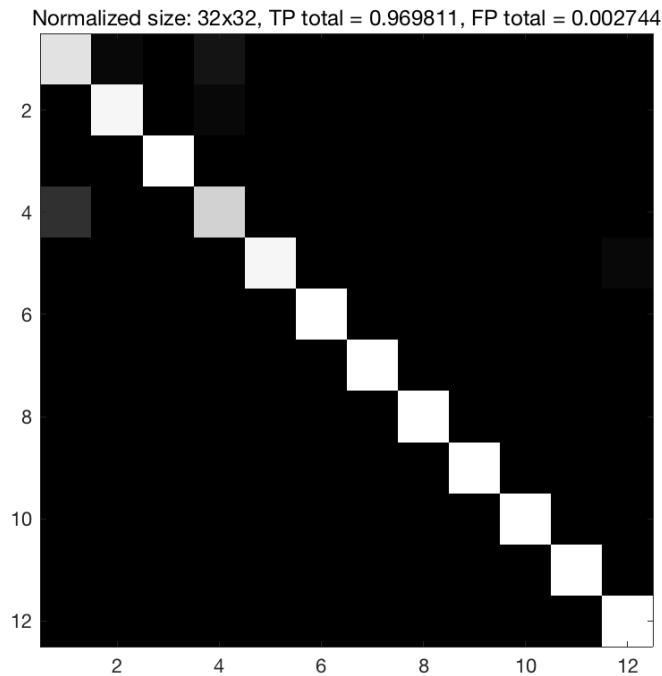


Figure 9: 总识别率 = 96.98%，总虚警率 = 0.27%

尝试不同的归一化尺寸，直接影响 HOG 特征的维度，因此影响 SVM 分类的识别率：

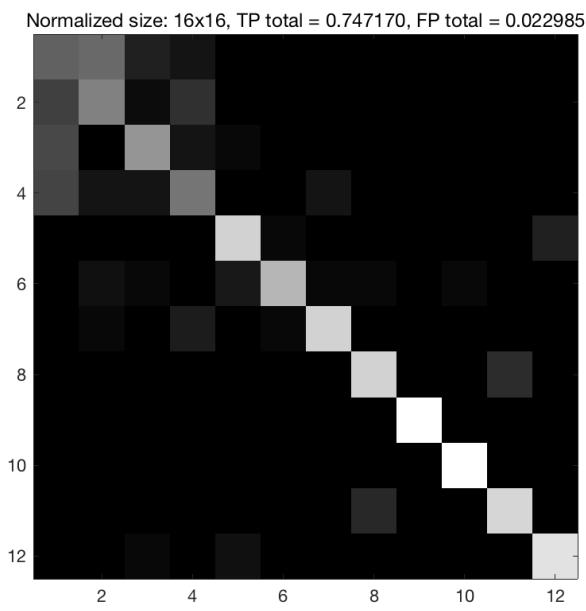


Figure 10: 归一化尺寸: 16x16, 总识别率 =74.72%, 总虚警率 =2.30%

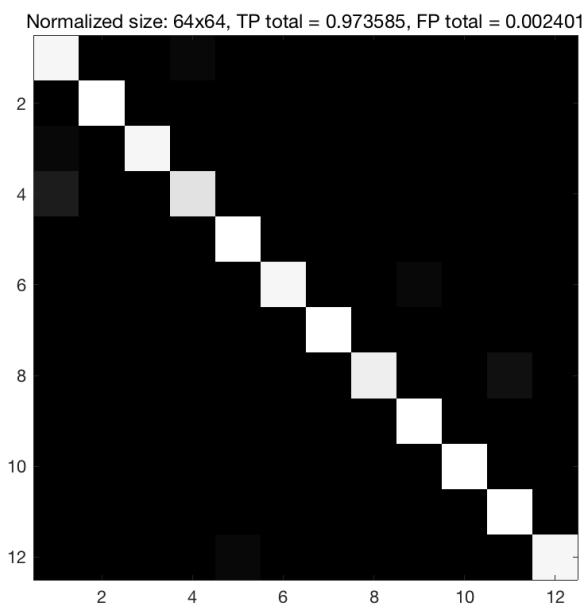


Figure 11: 归一化尺寸: 64x64, 总识别率 =97.36%, 总虚警率 =0.24%

### 3.3.4 结果分析

- HOG+SVM 分类器的分类性能明显优于 PCA+KNN 方法（在特征维度相同的情况下）。
- 实际上运用了一组 SVM 进行多次二分类判别，用 ECOC 的方法综合为多类分类问题，而每个 SVM 机器对于对应的二分类问题来说，都是很强的分类器（完全可分）。
- HOG 特征维度对 SVM 的分类识别率有影响，整体来看，特征维度越大，准确率越高，但运算代价也越大。HOG 特征维度较小时，SVM 分类性能下降比较剧烈。

## 3.4 HOG+SVM 开集测试（题目 3）

### 3.4.1 思路简述

- SVM 机器在进行预测时，对可能所属的类别的置信程度可以通过后验概率或者损失误差来体现。
- SVM 成功识别时，其对各类别的置信概率往往呈单峰分布，即某个类别置信程度可达 90%，而其他类别在 10% 以下。
- SVM 可能出现误判时，往往有多个类别的置信概率较接近，如两个类别置信度均为 40% 左右，容易出现误判。
- 基于以上特点，将 SVM 对各类的最高置信度与第二高置信度的差作为拒识标志，对此值设定一个阈值，完成拒识。

### 3.4.2 实验结果

尝试不同的置信度差值阈值，绘制总识别率以及负样本拒识率曲线如图：

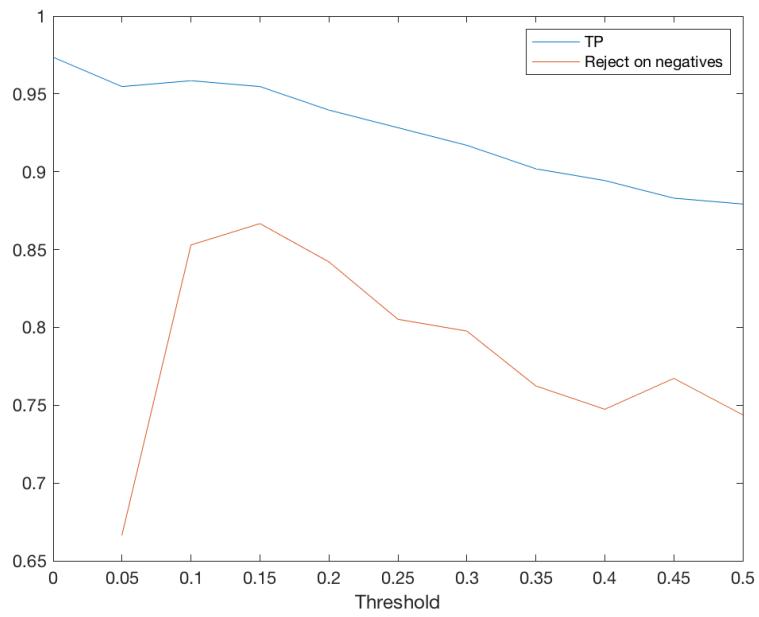


Figure 12: 置信度差值阈值对总识别率、负样本拒识率的影响。阈值 =0.15 是较优的选择。

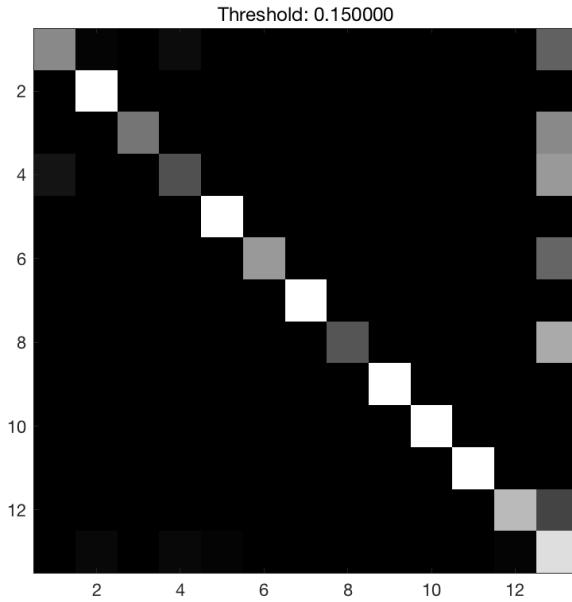


Figure 13: 阈值 =0.15 时，P 矩阵可视化。总识别率 =95.47%，负样本拒识率 =86.67%

### 3.4.3 结果分析

- 选取最大置信度与第二大置信度差值作为阈值来确定是否拒识的效果可以接受，在不严重影响总识别率的前提下，对负样本达到较高的拒识率（86.67%）。
- 可以考虑误差损失等作为阈值标准。

### 3.5 HOG+SVM 4 大类识别（选做 1）

- 由于 HOG+SVM 相比于 PCA+KNN 在此分类问题上体现出了较大的优势，因此在 source/数据集，直接采用 HOG+SVM 的方法。
- 由于只需要对 4 大类进行识别，所需的 SVM 数量大幅减少，训练速度大幅提升。
- 对于原始数据集，按照 trainRatio 来确定训练集所占整体数据集的比例，采用随机采样的方式，并保证训练集各类别比例与原始数据集比例基本一致。

探究训练集所占比重对总识别率与总虚警率的关系：

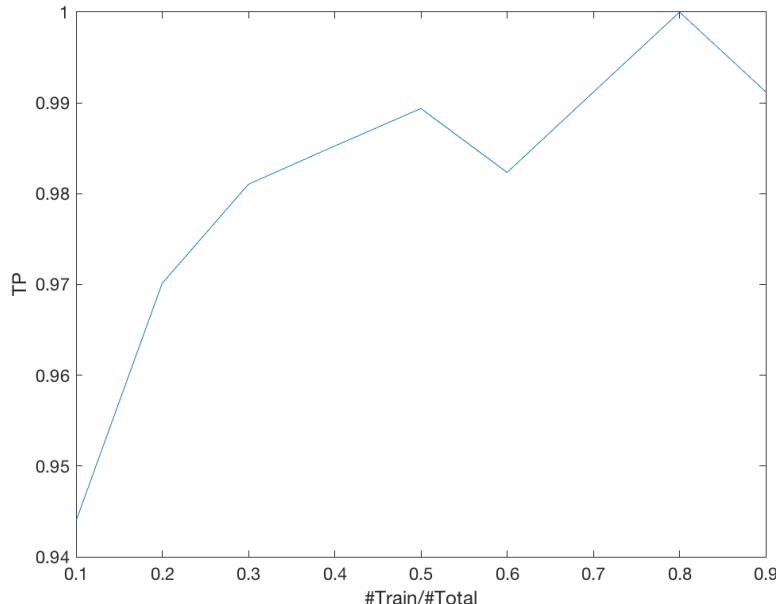


Figure 14: 总识别率随训练集比例的变化

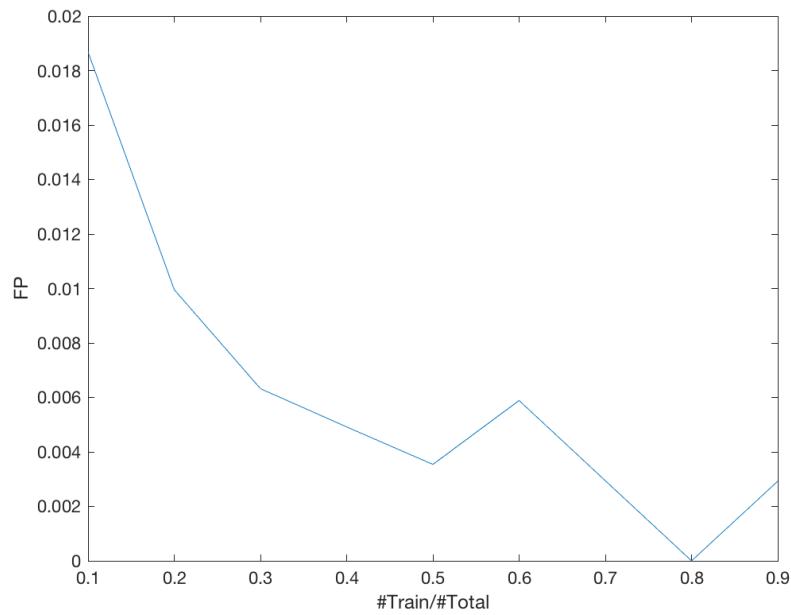


Figure 15: 总虚警率随训练集比例的变化

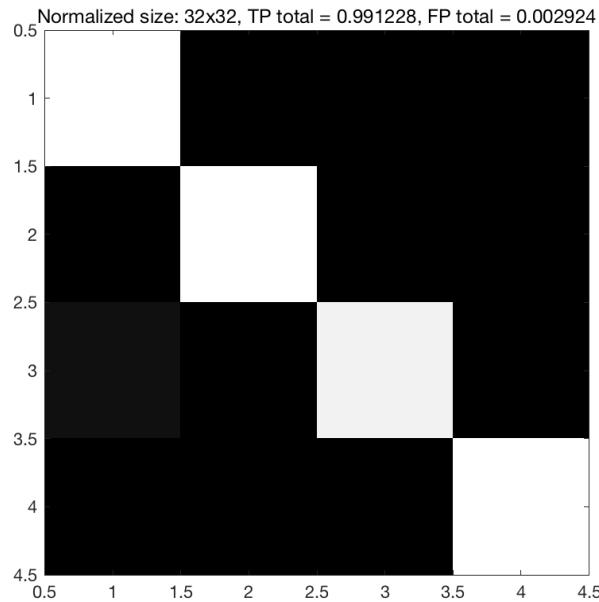


Figure 16: 取 trainRatio=0.8

由于只需要分四大类，即使只用很小一部分数据集进行训练（10%），也可以达到相

当高的识别率（95%）。

### 3.6 RCNN 检测 + 识别（选做 2）

#### 3.6.1 系统框图

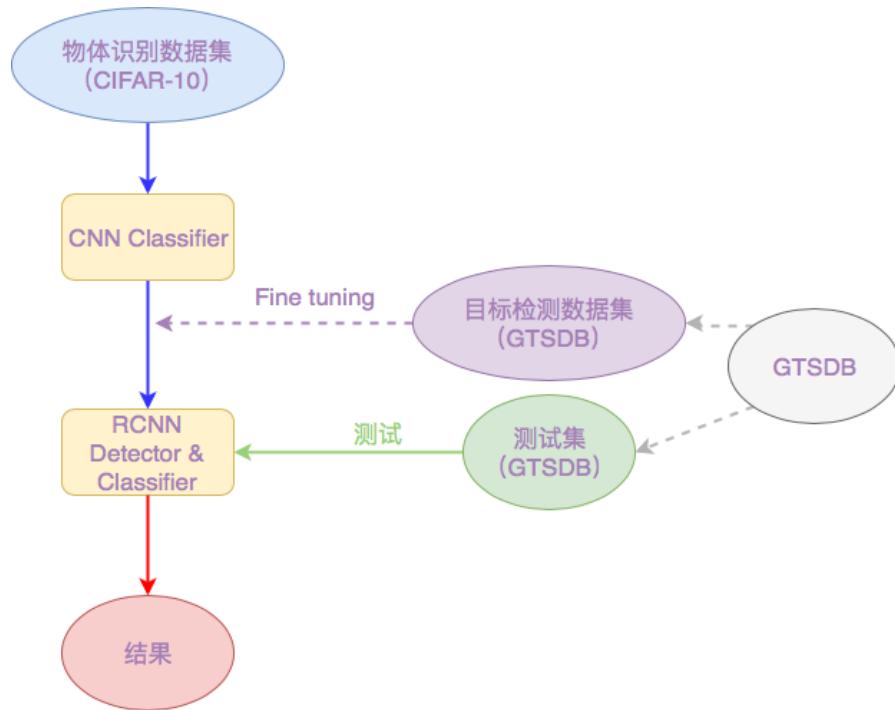


Figure 17: R-CNN 目标检测框图

#### 3.6.2 原理简述

- 需要完成**目标检测**和**识别分类**两个步骤。
- 基于 transfer learning：现在通用的物体识别数据集上训练简单结构的 CNN，然后移植到 R-CNN 网络，通过对权重的精细调整，以完成目标检测与识别交通标志的任务。
- 参考：<https://www.mathworks.com/help/vision/examples/object-detection-using-deep-learning.html>

#### 3.6.3 实验结果

在 CIFAR-10 数据集训练所得的 CNN 卷积核可视化结果：

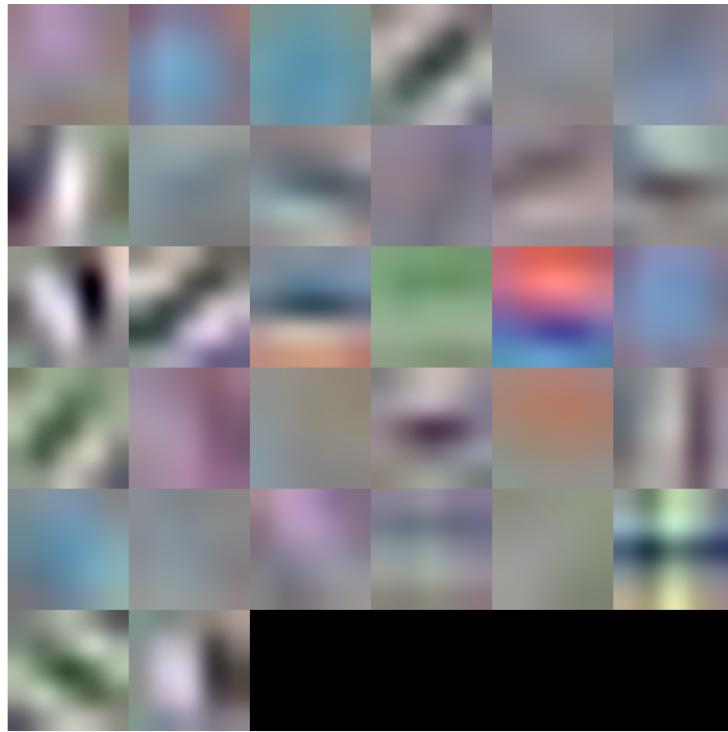


Figure 18: 预训练的物体识别 CNN 网络卷积核可视化

可以看到，预训练的 CNN 对一些基本纹理、颜色特征均能较好地响应，只要交通标志中含有类似的纹理特征，即可通过对 CNN 网络的精细调整进而识别目标。

问题的难点在于目标区域的检测，要从一个大图中检测出目标所在的位置，传统的方法使用模板匹配 + 滑动窗口 + 设定阈值等一系列方法来筛选 ROI，最后通过邻近 ROI 的融合，得到可能的 Region Proposals，在运用目标识别的方法来完成分类。

R-CNN 也类似地根据少量含 ROI 真值的交通标志检测图像（小训练集）提供了大量的 Region Proposals（但非滑动窗口形式），通过特征提取得到 Feature Map，再对预训练的 CNN 精细调整同时得到 ROI 与分类结果。



Figure 19: 以高于 99.9% 的置信度检出两个 mandatory 标志。可以看到将绿灯错误地当作目标检出了，也漏检了图中的 STOP 标志。



Figure 20: 以高于 99.6% 的置信度检出两个 prohibitory 标志。



Figure 21: 以高于 99.99% 的置信度检出 prohibitory 标志。但同时也将同一标志误识别成了 mandatory 标志。

由于时间关系，没有完整地测量出测试集上整体的识别率等评价参数。仅提供上述若干图作为参考。

### 3.6.4 结果分析

- GTSDB 数据集共含有 1213 个 ROI 真值，为了达到较好的检测 + 识别效果，最终随机选取 90% 的数据集作为训练集，剩余 10% 作为测试集。
- 尽管训练集规模占比已经比较大，但在测试集上进行测试时，仍有一些虚警与漏检。
- 另外，由于目标检测任务中，目标的尺寸是不固定的，那么预训练 CNN 所采用的图像尺寸也会对后级 RCNN 的目标检测有较大影响。预训练 CNN 的输入图像与第一层卷积核的尺寸，决定了 CNN 对于何种尺度的纹理有较好的识别效果，如果纹理尺度不能和目标检测中的纹理尺度匹配，那么也会影响识别效果。
- 在实际操作中，还尝试了 FasterRCNN 等运算密集方法，为了加快网络训练速度，借用了同学搭载 1070Ti 的工作平台，但 FasterRCNN 在两个 Stage 之间的级间预处理占用了过多时间，跑了超过 3 天时间，仍没有出结果，故放弃。FasterRCNN 不需要预训练 CNN，即不使用 Transfer learning，直接在原始数据集上训练，理论上应当结果更好。
- 还有一个思路是，在预训练前级 CNN 时，可以直接用 GTSDB 所提供的纯交通标志数据集 (source/) 来训练，而不是在 CIFAR-10 等通用识别集上训练。尽管数

据集规模较小（相比于 CIFAR-10），但更有针对性，能训练出对交通标志更有效的卷积核，可惜没时间完成这部分代码。

## 4 代码说明

- 所有源码及报告内容均托管在 github: <https://github.com/Nuullll/traffic-sign-detection>
  - 运行环境:
- 

MATLAB 版本: 9.4.0.813654 (R2018a)

MATLAB 许可证编号: 1114839

操作系统: Mac OS X Version: 10.13.4 Build: 17E202

Java 版本: Java 1.8.0\_144-b01 with Oracle Corporation Java HotSpot™ 64-Bit Server VM mixed mode

---

- 数据集目录位置：与本项目同级的 `dataset`/文件夹下。
- 对于 R-CNN 模块，需将数据集直接放在 `rcnn-sign-detection`/文件夹下，有点乱，见谅。
- `utilities`/文件夹下存放了一些共用的用于 IO 的函数。
- `pca-recognition`/与 `svm-recognition`/分别对应 PCA+KNN 和 HOG+SVM 方法。
- 各目录下的 `run.m` 主要用于绘制曲线等功能，可以灵活修改；`pcaKnnModel.m` 与 `hogSvmModel.m` 分别对应两种方法的核心逻辑。