

# 上海大学 2023 ~ 2024 学年 课程报告成绩评价表

课程名称： 《模式识别》 课程编号： 08306089

报告名称： 基于 HOG+SVM 与 YOLO 的路标识别分类

姓 名： 郑力铖 学 号： 21122873

报告评语：

--

报告成绩：

方案设计（20 分）		验收（20 分）		书面报告（60 分）			总分
可行性 (10 分)	创新性 (10 分)	规范性 (10 分)	演示效果 (10 分)	规范性 (20 分)	完整性 (20 分)	科学性 (20 分)	

任课教师：

评阅日期：        年        月        日

# 基于 HOG+SVM 与 YOLO 的路标识别分类

郑力铖 (21122873)

**摘要:** 作为交通系统的基本要素, 交通标志提供了关于驾驶员、行人等的道路状况的重要信息, 来降低事故风险。随着计算机视觉和人工智能的快速发展, 交通标志识别算法已被应用于先进的驾驶员辅助系统和自动驾驶系统, 以帮助驾驶员和自动驾驶车辆准确获取道路信息。然而在实际应用中, 交通标志识别仍然具有挑战性。本文对比了经典的机器学习方法 HOG+SVM, 和通用视觉识别模型 yolov5 和 yolov8 方法, 探讨在自动驾驶场景下的路标识别方法。

**关键词:** 图像识别, 支持向量机, 深度学习, 神经网络。

## 1 引言

### 1.1 问题提出

作为高级驾驶员辅助系统 (ADAS) 和自动驾驶系统 (ADS) 的重要组成部分, 交通标志识别 (TSR) 技术可以帮助驾驶员和自动驾驶汽车获取重要的道路信息 [1]。交通标志识别 TSR 是一种为自动驾驶车辆识别道路上的交通标志的智能识别系统, 是物体检测的一个子任务, 类似于人脸识别。

TSR 系统通常由检测和分类两个阶段组成。检测通常使用形状和颜色来识别交通标志的特征, 从而从自然场景中提取交通标志。更好的分类能更准确地识别的内容检测到的交通信号。交通标志的识别度, 对于避免道路事故具有重大意义 [2]。本文通过对比不同方法的实际应用效果, 旨在为自动驾驶系统选择最合适的路标识别方法提供参考。考虑到实时性、准确性和鲁棒性等因素, 读者将能够更好地了解何种方法在特定条件下更为适用。

### 1.2 求解方案分析

一般来说, 交通标志识别有两种检测方法, 一种是传统方法 (人工定制的特征), 另一种是深度学习方法 (学习具有深度神经网络的特征)。对于传统方法而言, 整个实现流程通常包括的提取区域兴趣 (包括交通标志), 提取特征 (例如 HOG), 以及然后将这些特征发送到分类器 (例如 SVM)。然而, 传统方法在多类任务中的表现的性能较差。近年来, 深度学习方法在许多任务中表现出优异的性能, 例如图像分类和检测。对于对象检测, 深度学习方法 (例如 YOLO[3], Faster R-CNN[4]) 在主流基准上表现良好, 准确度和速度都较传统方法有较大优势。

早在 2005 年, 方向梯度直方图 (HOG) 就被提出, HOG 用来提取图片特征, 并和支持向量机 (SVM) 结合, 最早用来进行行人检测。HOG+SVM 这种经典的机器学习方法在过去取得了一定的成功, 但随着技术的进步, 人们开始寻求更加高效、准确的解决方案。近年来, YOLO 算法以其高速度和高精度迅速走红。因此, 本文深入研究了 yolov5 和 yolov8 这两种先进的目标检测模型。这两种模型都是基于 yolo 模型的优化, 利用深度学习技术, 通

过神经网络层次结构和先进的目标检测算法，在图像中快速准确地识别路标。相较于传统方法，它们在处理复杂场景和多类别标识时表现更为出色。

### 1.3 论文概述

本文的其余部分的组织如下。在第二章我们介绍了梯度直方图与支持向量机(HOG+SVM)，以及对象检测算法 YOLO 的算法概述。在第三章我们实现了 HOG+SVM，YOLOv5 和 YOLOv8 在数据集上的实验代码，介绍了算法的总体框架，并做了一部分改进。第四章分别对 HOG+SVM，和 YOLO 的两种算法进行了对比，同时在 YOLOv5 和 YOLOv8 间的实验结果做了对比。最后在第五章做了讨论和结论。

## 2 相关算法概述

### 2.1 梯度直方图 (HOG)

梯度直方图 (Histogram of oriented gradient, 简称 HOG) [5] 是应用在计算机视觉和图像处理领域，用于目标检测的特征描述器。这项技术是用来计算局部图像梯度的方向信息的统计值。在 HOG 特征描述符中，梯度方向的分布，也就是梯度方向的直方图被视作特征。图像的梯度 (x 和 y 导数) 非常有用，因为边缘和拐角 (强度突变的区域) 周围的梯度幅度很大，并且边缘和拐角比平坦区域包含更多关于物体形状的信息。方向梯度直方图 (HOG) 特征描述符常和线性支持向量机 (SVM) 配合使用，用于训练高精度的目标分类器。

### 2.2 支持向量机 (SVM)

支持向量机 (Support Vector Machine, 常简称为 SVM, 又名支持向量网络) [6] 是在分类与回归分析中分析数据的监督式学习模型与相关的学习算法。给定一组训练实例，每个训练实例被标记为属于两个类别中的一个或另一个，SVM 训练算法创建一个将新的实例分配给两个类别之一的模型，使其成为非概率二元线性分类器。SVM 模型是将实例表示为空间中的点，这样映射就使得单独类别的实例被尽可能宽的明显的间隔分开。然后，将新的实例映射到同一空间，并基于它们落在间隔的哪一侧来预测所属类别。

### 2.3 对象检测算法 YOLO

YOLO (You Only Look Once) [3] 是一种流行的对象检测和图像分割模型，由华盛顿大学的 Joseph Redmon 和 Ali Farhadi 开发。它的主要特点是在一次前向传递中，同时完成对象定位和分类。相较于传统的对象检测方法，YOLO 在速度和准确性上取得了很大的突破。这使得它适用于实时目标检测任务，如视频分析和自动驾驶。

3 算法实现描述

- 3.1 算法总体框架 (>500 字)
- 3.2 改进一及分析 (>500 字)
- 3.3 改进二及分析 (>500 字)

4 实验描述

- 4.1 实验数据和实验方案 (>500 字)
- 4.2 实验一及结果分析 (>500 字)

表 1: HOG+SVM 的路标识别多指标分析

	精确率	召回率	F1 得分
直行标志	0.3706	0.3251	0.3464
左转标志	0.5889	0.3869	0.4670
右转标志	0.3750	0.4565	0.4118
禁止鸣笛	0.4999	0.5682	0.5319
人行横道	0.8095	0.5965	0.6869
禁止通行	0.7288	0.7049	0.7167



图 1: 质量较好的 HOG+SVM 检测结果

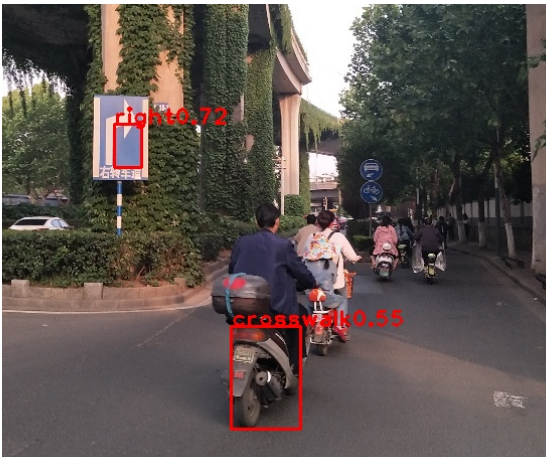


图 2: 明显错检与 roi 不一致问题

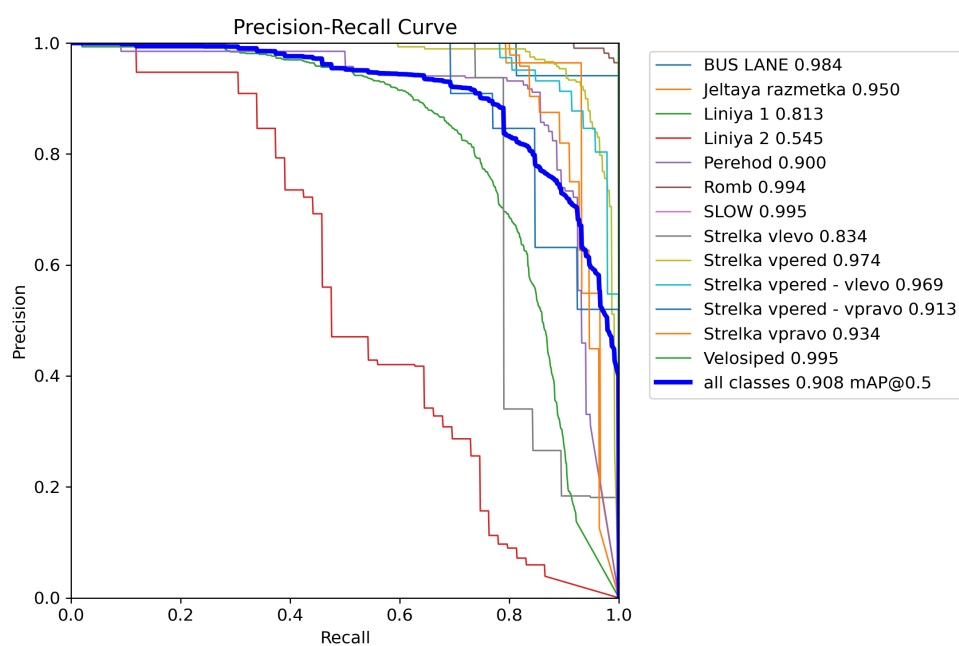


图 3: YOLOv5 的准确度-召回率图, YOLOv8 与之类似。

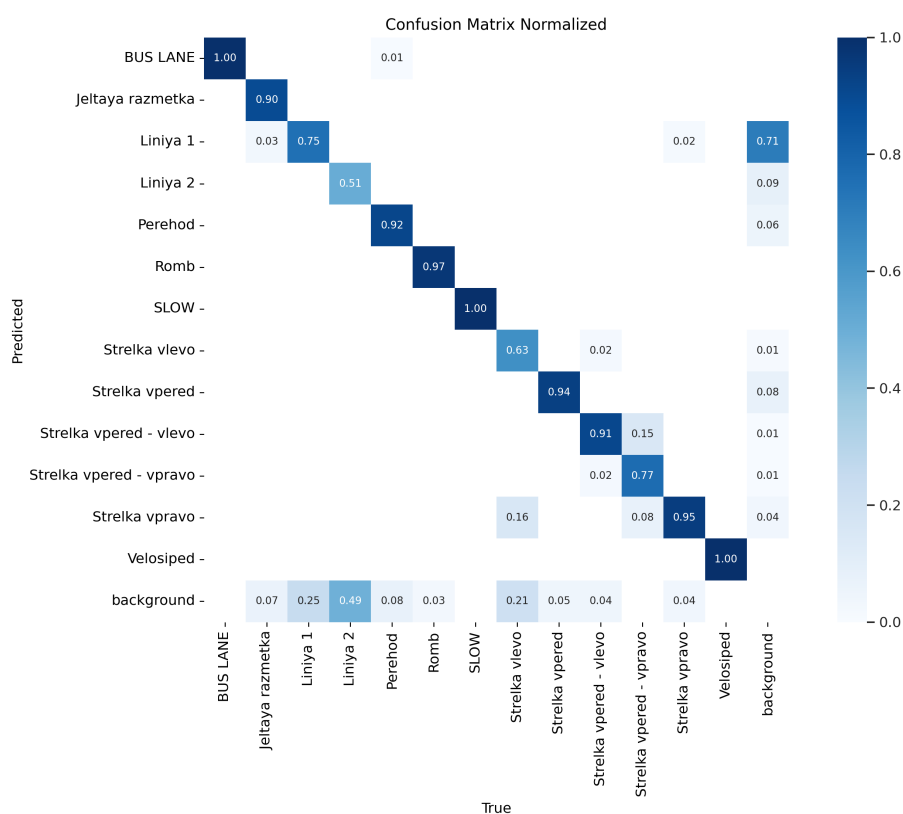


图 4: YOLOv5 的归一化混淆矩阵图, YOLOv8 与之类似。

### 4.3 实验二及结果分析 (>500 字)

## 5 结论 (500 字)

## 6 学习体会和建议

本学期在模式识别课上，方昱春老师深入地介绍了一些模式识别的理论基础，与不同算法模型和技术，以及它们在一些不同场景中的应用。建立起一定的理论基础后，通过研讨和小组作业等形式，上手实际项目，让我更好地理解算法模型的工作原理、优缺点以及如何调整参数以提高性能。这培养了我解决问题的能力，有助于更好地理解问题，从而选择合适的方法，并在实践中更加灵活地应用知识。希望今后能在课程中加入更多前沿领域的最新科研成果分享，并与小组作业有机结合起来，让同学们更好地感受到新成果带来的进步，并能有机会启发同学们在今后参与进前沿的计算机科研的工作中去的热情。

## 参考文献

- [1] J. Chen, K. Jia, W. Chen, *et al.*, “A real-time and high-precision method for small traffic-signs recognition,” *Neural Computing & Applications*, vol. 34, pp. 2233–2245, 2022.
- [2] R. Zhang, Z. He, H. Wang, F. You, and K. Li, “Study on self-tuning tyre friction control for developing main-servo loop integrated chassis control system,” *IEEE Access*, vol. 5, 2017.
- [3] “Redmon j, divvala s, girshick r, farhadi a (2016) you only look once: unified, real-time object detection, pp 779–788.”
- [4] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: towards real-time object detection with region proposal networks,” *IEEE Trans Pattern Anal Mach Intell*, vol. 39, 2016.
- [5] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*, vol. 1, pp. 886–893 vol. 1, 2005.
- [6] M. Hearst, S. Dumais, E. Osuna, J. Platt, and B. Scholkopf, “Support vector machines,” *IEEE Intelligent Systems and their Applications*, vol. 13, no. 4, pp. 18–28, 1998.

## A 附录

### 1、图模板

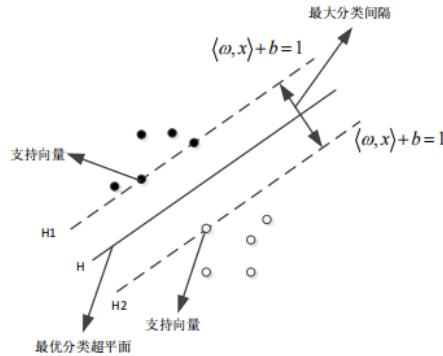


图 5: SVM 模型原理图

### 3、公式模板

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2} \|\mathbf{w}\|^2 \\ \text{s.t.} \quad & y_i (\mathbf{w}^T \mathbf{x}_i + b) \geq 1, \quad i = 1, 2, \dots, m \end{aligned} \quad (1)$$

### 4、伪代码模板

---

**Algorithm 1** K 近邻算法步骤

---

**Input:** 训练数据集; 待预测数据;

**Output:** 预测数据的类别;

- 1: 加载数据;
  - 2: 初始化 K 值;
  - 3: 计算预测样本与训练集中的每一个样本的距离;
  - 4: 将距离和索引添加到有序集合中;
  - 5: 对距离按从小到大排序方式对距离和索引的有序集合进行排序;
  - 6: 从排序的集合中选择前 K 条数据;
  - 7: 获得选的 K 条数据的标签;
  - 8: 计算每一种标签的样本数量; **return** 数量最多的标签作为样本的预测值;
- 

### 5、核心代码

```
1 #调整图片尺寸到统一大小，并扁平化为一维数据
2 def image_to_feature_vector(image, size=(128, 128)):
3
4     return cv2.resize(image, size).flatten()
5
6 #提取图像在HSV颜色空间上的颜色直方图，将直方图扁平化，
7 #作为特征向量返回
8 def extract_color_histogram(image, bins=(32, 32, 32)):
9     hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)
10    hist = cv2.calcHist([hsv], [0, 1, 2], None, bins,
```

```
11         [0, 180, 0, 256, 0, 256])
12     if imutils.is_cv2():
13         hist = cv2.normalize(hist)
14     else:
15         cv2.normalize(hist, hist)
16     return hist.flatten()
```