

# 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: 基于卷积神经网络与 SVM 的路标识别

姓 名: 郑力铨 学 号: 21122873

报告评语:

[illegible]

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

# 基于卷积神经网络与 SVM 的路标识别

郑力钺 (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 提出问题 (300-500 字)

### 1.1.1 三级标题

[illegible]

### 1.1.2 三级标题

[illegible]

## 1.2 求解方案分析

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，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。

在自动驾驶场景下，路标识别至关重要。本文通过对比这些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。

1.3 论文概述 (200 字)

正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文。正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文。正文正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文。

2 相关算法概述

2.1 梯度直方图与支持向量机 (HOG+SVM)

正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文正文。正文正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文，如图1所示。

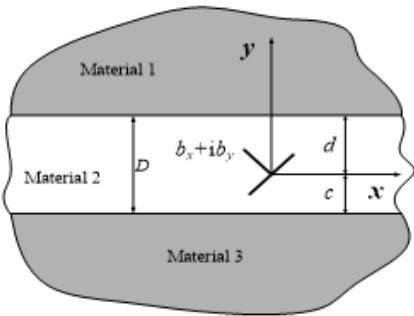


图 1: 位于三层材料体系中的位错示意图

2.2 对象检测算法 YOLO

YOLO (You Only Look Once) 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。

3 算法实现描述

正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文正文。正文正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文。



## 5 结论 (500 字)

正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文。正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文。正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文。

## 6 学习体会和建议 (300 字)

正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文。正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文。正文正文正文正文正文正文正文正文正文正文正文，正文正文正文正文正文正文正文正文正文正文正文。

# A 附录

## 1、图模板

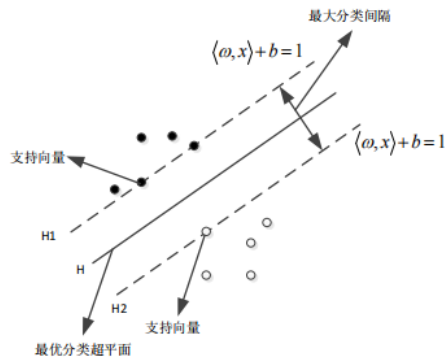


图 2: SVM 模型原理图

## 2、表模板

表 1: 最优算法的多指标分析

	精确率	召回率	F1 得分
石块	0.94	0.96	0.95
金属	0.92	0.97	0.95
塑料	0.96	0.89	0.93

## 3、公式模板

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

## 4、伪代码模板

<b>Algorithm 1</b> K 近邻算法步骤
<b>Input:</b> 训练数据集; 待预测数据;
<b>Output:</b> 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; <b>return</b> 数量最多的标签作为样本的预测值;

## 5、代码模板

```

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()

```