

# 《计算机视觉》实验报告

姓名：汪江豪 学号：22121630

## 实验六

### 一. 任务 1

- 1、在数据集 INRIADATA 上，使用 hog+svm 实现行人检测
- 2、模型参数调优，提升检测效果
- 3、画出 roc 曲线

#### a) 核心代码：

1. 先定义好 HOG 参数

```
# HOG 参数 (方向梯度直方图)
winSize = (64, 128) # HOG 窗口大小
blockSize = (16, 16) # HOG 块大小
blockStride = (8, 8) # HOG 块步长
cellSize = (8, 8) # HOG 单元格大小
nbins = 9 # HOG 方向数
```

2. 提取 hog 特征

```
def extract_hog(img_path):
    img = cv2.imread(img_path, cv2.IMREAD_COLOR) # 读取图片
    if img is None:
        raise ValueError('无法读取图片: {}')
    img = cv2.resize(img, winSize) # 调整图片大小
    features = hog.compute(img) # 计算 HOG 特征
    return features.flatten() # 返回展平特征向量
```

3. 加载数据集

```
def load_data(pos_dir, neg_dir, max_samples=None):
    X, y = [], []
    pos_imgs = [os.path.join(pos_dir, f) for f in os.listdir(pos_dir) if
    f.endswith('.png') or f.endswith('.jpg')]
    neg_imgs = [os.path.join(neg_dir, f) for f in os.listdir(neg_dir) if
    f.endswith('.png') or f.endswith('.jpg')]
```

```

# 只提取前 max_samples 个样本
if max_samples:
    pos_imgs = pos_imgs[:max_samples]
    neg_imgs = neg_imgs[:max_samples]

    for img_path in tqdm(pos_imgs, desc='加载正样本'):
        X.append(extract_hog(img_path))
        y.append(1)
    for img_path in tqdm(neg_imgs, desc='加载负样本'):
        X.append(extract_hog(img_path))
        y.append(0)
return np.array(X), np.array(y)

```

#### 4. SVM 参数调优

```

best_auc = 0 # 初始 AUC
best_C = 1.0 # 初始 C 参数
Cs = [0.01, 0.1, 1, 10, 100] # C 参数范围, 值越大, 拟合程度越高
for C in Cs:
    clf = LinearSVC(C=C, max_iter=10000) # 创建 SVM 分类器
    clf.fit(X_train, y_train) # 训练模型
    y_score = clf.decision_function(X_test) # 决策函数值
    fpr, tpr, thresholds = roc_curve(y_test, y_score) # 计算 ROC 曲线
    roc_auc = auc(fpr, tpr) # 计算 AUC
    print(f'C={C}, AUC={roc_auc:.4f}')
    if roc_auc > best_auc:
        best_auc = roc_auc
        best_C = C
print(f'最佳参数: C={best_C}, 最优 AUC={best_auc:.4f}')

```

#### 5. 采用最优参数绘制 AOC 曲线

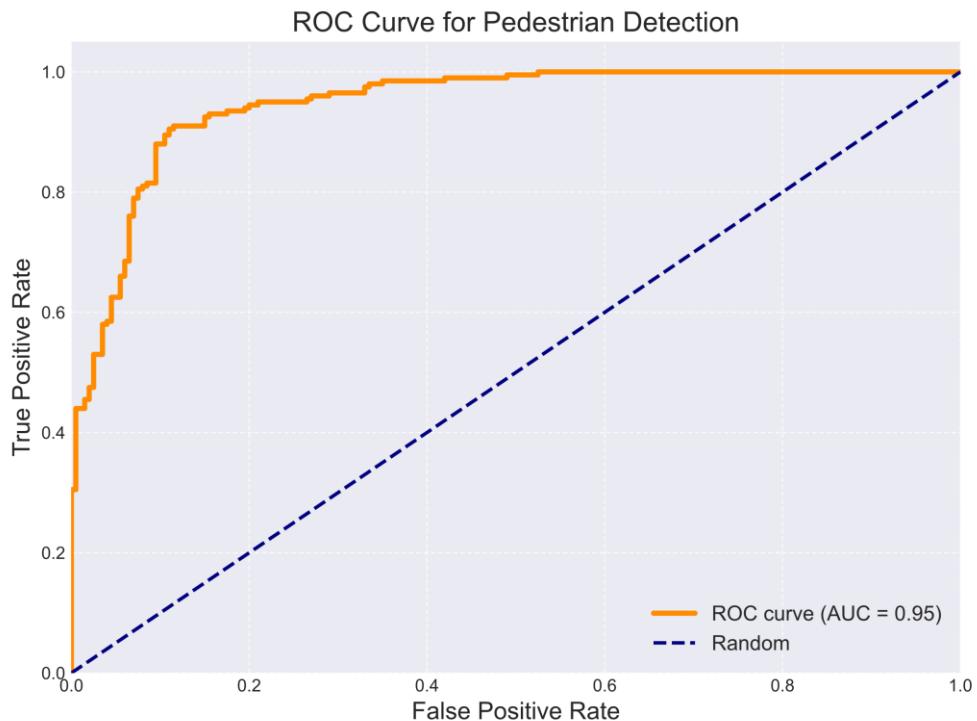
```

# 用最优参数训练最终模型、
clf = LinearSVC(C=best_C, max_iter=10000)
clf.fit(X_train, y_train)
y_score = clf.decision_function(X_test)
# 绘制 ROC 曲线
fpr, tpr, thresholds = roc_curve(y_test, y_score)
roc_auc = auc(fpr, tpr)
plt.figure(figsize=(8, 6), dpi=120)
plt.style.use('seaborn-v0_8-darkgrid') # 使用 seaborn 风格
plt.plot(fpr, tpr, color='darkorange', lw=3, label=f'ROC curve (AUC = {roc_auc:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--',
label='Random')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])

```

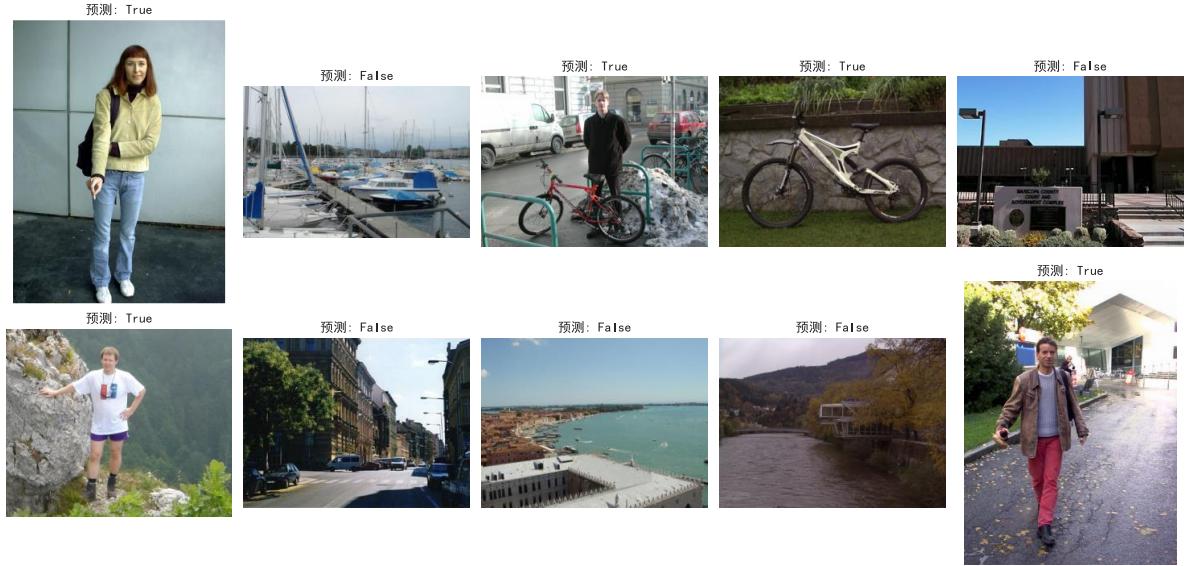
```
plt.xlabel('False Positive Rate', fontsize=14)
plt.ylabel('True Positive Rate', fontsize=14)
plt.title('ROC Curve for Pedestrian Detection', fontsize=16)
plt.legend(loc='lower right', fontsize=12)
plt.grid(True, linestyle='--', alpha=0.7)
plt.tight_layout()
plt.savefig('roc_curve.svg')
plt.show()
```

b) 实验结果截图



曲线下方面积为 0.95，表明该行人检测模型在区分行人和飞行人方面表现出色，有较强的辨别能力。能较好地平衡真阳性率和假阳性率，能较为准确地检测出行人，同时误判的情况相对较少。

随机抽取10张测试图片的预测结果



### c) 实验小结

本次实验旨在利用方向梯度直方图（HOG）特征和支持向量机（SVM）算法实现行人检测。通过对行人与非行人图像数据的处理，提取 HOG 特征并训练 SVM 模型，最终在测试集上进行行人检测验证。

在使用 SVM 算法对提取的 HOG 特征进行训练，经过参数调优，模型在测试集上取得了一定的检测准确率。实验中通过调整 SVM 的参数 C，找到了相对较优的模型配置，提高了模型的泛化能力和检测精度。

在最终的测试集随机测试结果中，仍然存在部分预测错误现象，我推测是这些图片的 HOG 特征很可能与行人的相似，后期可以考虑其他方法，比如深度学习方式训练，来降低这种错误。