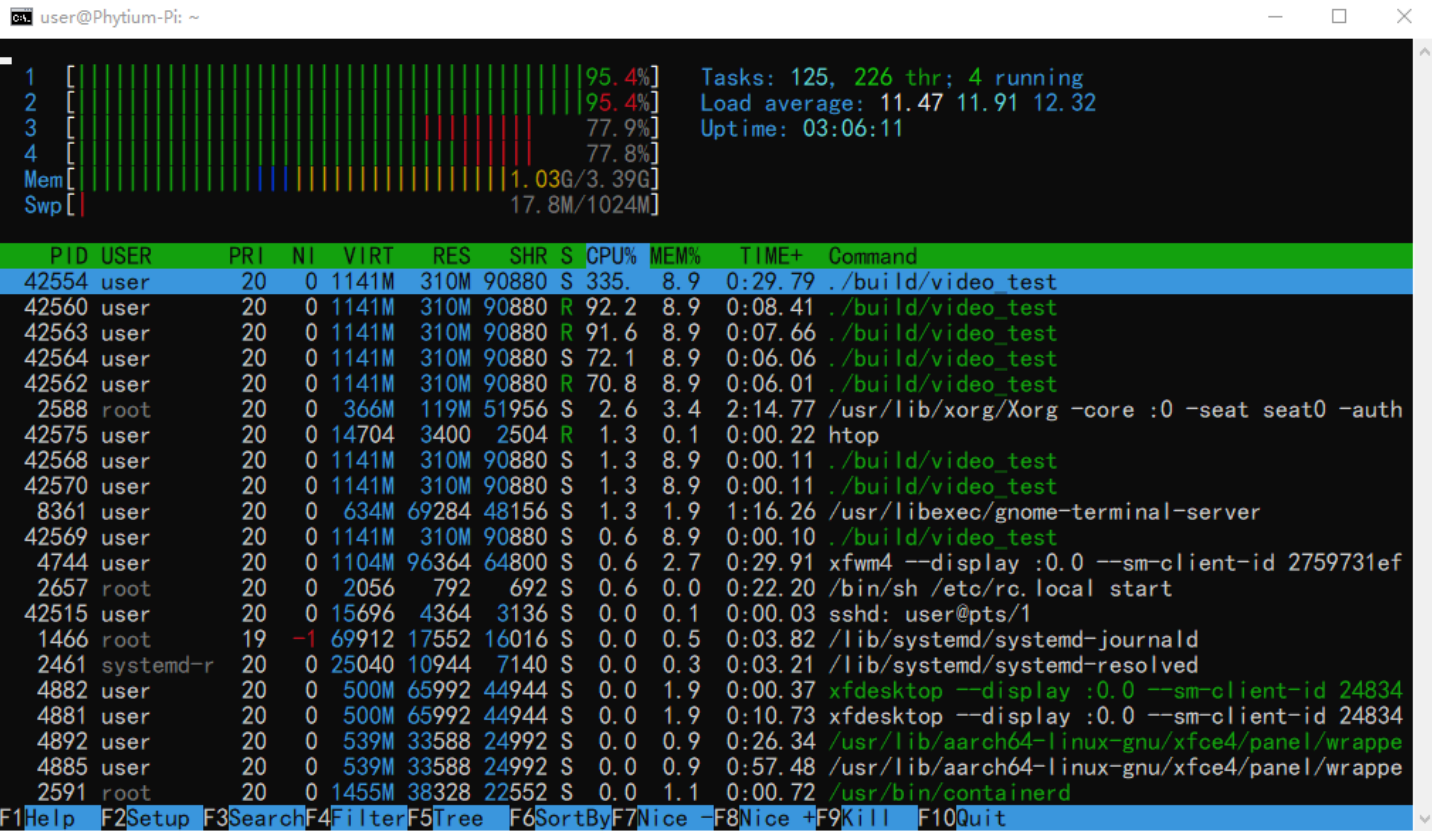


测试文档

图像识别

使用视频模式输入，平均cpu利用率为356%(四核理论值为400%)



平均推理时长为0.53s。

```
user@Phytium-Pi: ~/robot/yolo/HumanRecognition
```

```
Output shape: [1, 5, 8400]
Detected 1 objects after NMS
使用时间: 560毫秒
Output shape: [1, 5, 8400]
Detected 1 objects after NMS
使用时间: 533毫秒
Output shape: [1, 5, 8400]
Detected 1 objects after NMS
使用时间: 539毫秒
Output shape: [1, 5, 8400]
Detected 0 objects after NMS
使用时间: 523毫秒
Output shape: [1, 5, 8400]
Detected 1 objects after NMS
使用时间: 523毫秒
Output shape: [1, 5, 8400]
Detected 1 objects after NMS
使用时间: 528毫秒
Output shape: [1, 5, 8400]
Detected 1 objects after NMS
使用时间: 527毫秒
Output shape: [1, 5, 8400]
Detected 1 objects after NMS
使用时间: 528毫秒
Output shape: [1, 5, 8400]
Detected 1 objects after NMS
使用时间: 549毫秒
Output shape: [1, 5, 8400]
Detected 1 objects after NMS
使用时间: 529毫秒
```

YOLOV8模型测试指标代码

YOLOV8模型测试指标代码

```
1  # evaluate_enhanced_fixed_v4.py
2  from ultralytics import YOLO
3  import matplotlib.pyplot as plt
4  from pathlib import Path
5  import time
6  import shutil
7  import torch
8  import yaml
9  import numpy as np
10 import matplotlib
11
12 def evaluate_model():
13     # ===== Font Configuration =====
14     try:
15         # Try using Source Han Sans
```

```

16     plt.rcParams['font.sans-serif'] = ['Source Han Sans SC', 'SimHei',
    'Microsoft YaHei']
17     plt.rcParams['axes.unicode_minus'] = False
18 except:
19     # Fallback to system default font
20     matplotlib.rcParams['font.family'] = 'sans-serif'
21     print("⚠ Chinese font configuration failed, using system default
font")
22
23     # 1. Initial configuration
24     model_path = 'runs/detect/best.pt'
25     data_yaml = 'datasets/inspection/my_data.yaml'
26
27     try:
28         # 2. Device detection
29         device = '0' if torch.cuda.is_available() else 'cpu'
30         print(f"🔧 Using device: {'GPU' if device == '0' else 'CPU'}")
31
32         # 3. Load model and dataset config
33         model = YOLO(model_path)
34         with open(data_yaml) as f:
35             data_cfg = yaml.safe_load(f)
36
37         print(f"✅ Model loaded successfully: {Path(model_path).name}")
38         print(f"📁 Dataset classes: {data_cfg['names']}")
39
40         # 4. Perform evaluation
41         print("🚀 Evaluating model...")
42         metrics = model.val(
43             data=data_yaml,
44             split='test',
45             batch=8,
46             conf=0.5,
47             iou=0.6,
48             device=device,
49             plots=True,
50             half=False if device == 'cpu' else True
51         )
52
53         # 5. Create save directory
54         timestamp = time.strftime("%Y%m%d_%H%M%S")
55         save_dir = Path(f"eval_results_{timestamp}")
56         save_dir.mkdir(exist_ok=True)
57
58         # 6. Calculate F1 score
59         def calculate_f1(p, r):
60             return 2 * p * r / (p + r + 1e-16)

```

```

61
62     # 7. Get test set sample count (fallback method)
63     test_img_dir = Path(data_yaml).parent / 'images/test'
64     num_test_images = len(list(test_img_dir.glob('*.*'))) if
test_img_dir.exists() else 'Unknown'
65
66     # 8. Prepare results text
67     results_text = [
68         f"Evaluation time: {timestamp}",
69         f"Model path: {model_path}",
70         f"Dataset config: {data_yaml}",
71         f"mAP50: {metrics.box.map50:.4f}",
72         f"mAP50-95: {metrics.box.map:.4f}",
73         f"Precision (mp): {metrics.box.mp:.4f}",
74         f"Recall (mr): {metrics.box.mr:.4f}",
75         f"F1-score: {calculate_f1(metrics.box.mp, metrics.box.mr):.4f}",
76         f"Test set size: {num_test_images}",
77         "\nPer-class results:"
78     ]
79
80     # Add detailed per-class metrics
81     for i, c in enumerate(metrics.box.ap_class_index):
82         cls_name = data_cfg['names'].get(c, str(c))
83         p = metrics.box.p[i]
84         r = metrics.box.r[i]
85         f1 = calculate_f1(p, r)
86         ap50 = metrics.box.ap50[i]
87         ap = metrics.box.ap[i]
88
89         results_text.append(
90             f"  {cls_name}({c}): "
91             f"AP50={ap50:.4f} "
92             f"AP={ap:.4f} "
93             f"P={p:.4f} "
94             f"R={r:.4f} "
95             f"F1={f1:.4f}"
96         )
97
98     # Print to console
99     print("\n📊 Evaluation results:")
100    print("\n".join(results_text[3:])) # Skip first 3 info lines
101
102    # Save to file
103    with open(save_dir/'evaluation_results.txt', 'w', encoding='utf-8') as
f:
104        f.write("\n".join(results_text))
105

```

```

106         # 9. Visualize metrics
107         plt.figure(figsize=(12, 6))
108         plt.title(f"Model Evaluation Metrics\n{Path(model_path).name}", pad=20)
109         plt.text(0.02, 0.5,
110                 "\n".join(results_text[3:]), # Only show metrics portion
111                 fontfamily='monospace',
112                 fontsize=10,
113                 bbox=dict(facecolor='ghostwhite', alpha=0.8))
114         plt.axis('off')
115         plt.savefig(save_dir/'metrics_summary.png', bbox_inches='tight',
116                   dpi=200)
117         plt.close()
118
119         # 10. Copy auto-generated plots
120         val_dir = Path("runs/detect/val/")
121         if val_dir.exists():
122             for img in val_dir.glob("*.png"):
123                 shutil.copy(img, save_dir/f"val_{img.name}")
124
125             print(f"\n📁 Results saved to: {save_dir.resolve()}")
126
127         except Exception as e:
128             print(f"\n❌ Evaluation failed: {str(e)}")
129             if hasattr(e, 'args'):
130                 print("Details:", *e.args)
131
132     if __name__ == "__main__":
133         evaluate_model()

```

YOLOV8模型测试指标说明

测试概述

本测试旨在评估YOLO目标检测模型在特定图像识别任务中的性能。测试使用独立测试集对模型进行全面评估，计算多种指标以衡量模型的检测精度、召回率和综合性能。

核心评估指标

1. mAP (Mean Average Precision)

- mAP50：在IoU阈值为0.5时的平均精度均值
 - 衡量模型在宽松重叠要求下的检测精度
- mAP50-95：在IoU阈值从0.5到0.95（步长0.05）的平均mAP
 - 综合评估模型在不同重叠要求下的性能

2. Precision (精确率)

- 计算公式: $TP / (TP + FP)$
- 含义: 模型预测为正例的样本中, 真正为正例的比例, 衡量模型预测结果的准确性, 避免误报

3. Recall (召回率)

- 计算公式: $TP / (TP + FN)$
- 含义: 实际为正例的样本中, 被模型正确预测的比例, 衡量模型发现目标的能力, 避免漏检

4. F1-score (F1分数)

- 计算公式: $2 * (Precision * Recall) / (Precision + Recall)$
- 含义: 精确率和召回率的调和平均数, 综合评估模型准确性和覆盖能力的平衡指标

各类别详细指标

1. AP50 (Average Precision at IoU=0.5)

- 特定类别在IoU=0.5时的平均精度
- 反映模型对该类别的检测能力

2. AP (Average Precision)

- 特定类别在多个IoU阈值下的平均精度
- 更全面评估类别检测性能

3. 类别级Precision和Recall

- 针对每个类别的精确率和召回率
- 识别模型在不同类别上的性能差异

测试集信息

- 测试集样本数: 评估使用的独立测试图像数量
- 重要性: 确保评估结果具有统计意义
- 建议: 测试集应足够大且代表实际应用场景

评估可视化

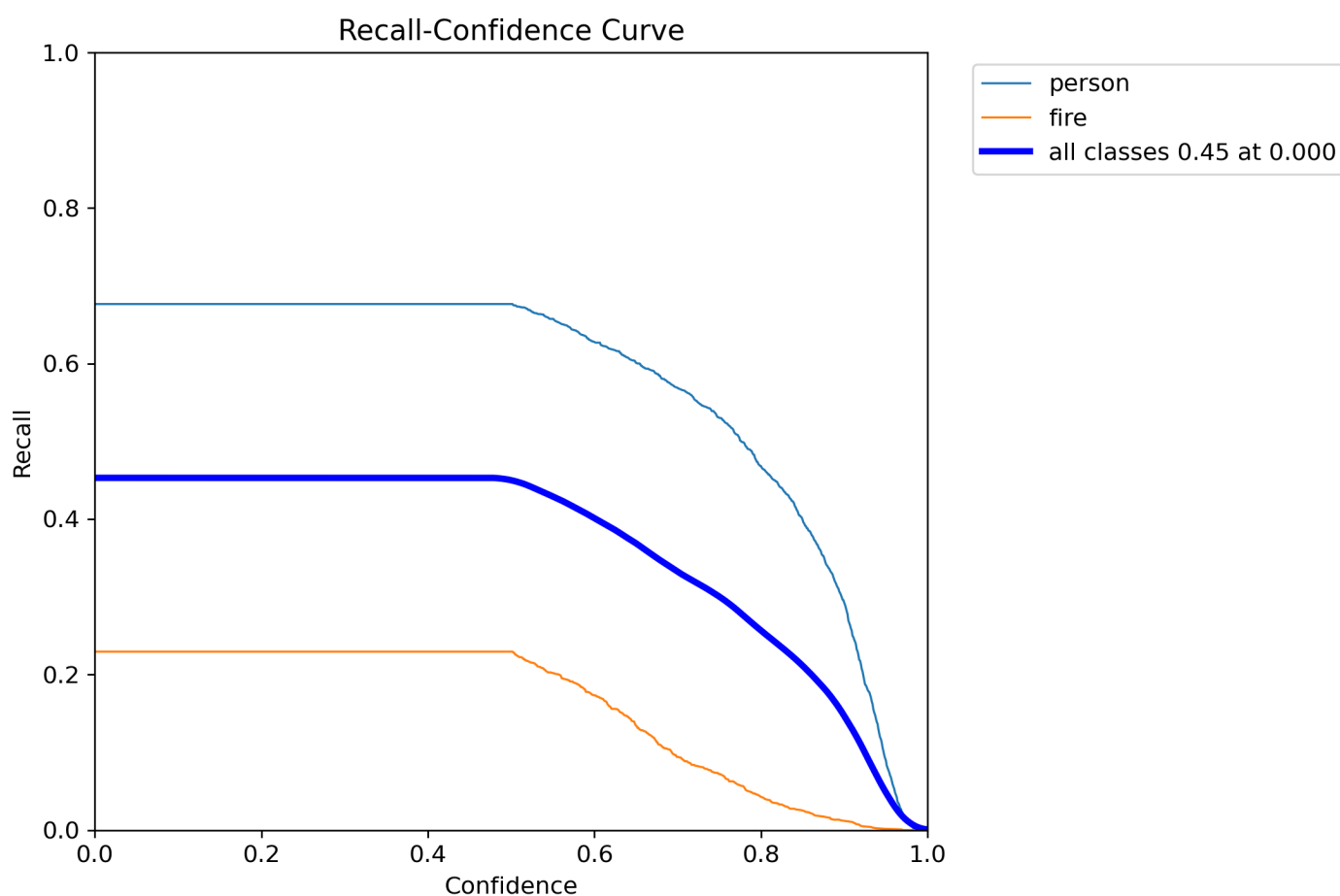
- 测试生成的图表包括:
 - PR曲线: 展示不同置信度阈值下的精确率-召回率平衡
 - 混淆矩阵: 直观显示各类别的预测结果分布
 - 检测示例: 可视化模型在实际图像上的检测效果

YOLOV8模型测试指标结果

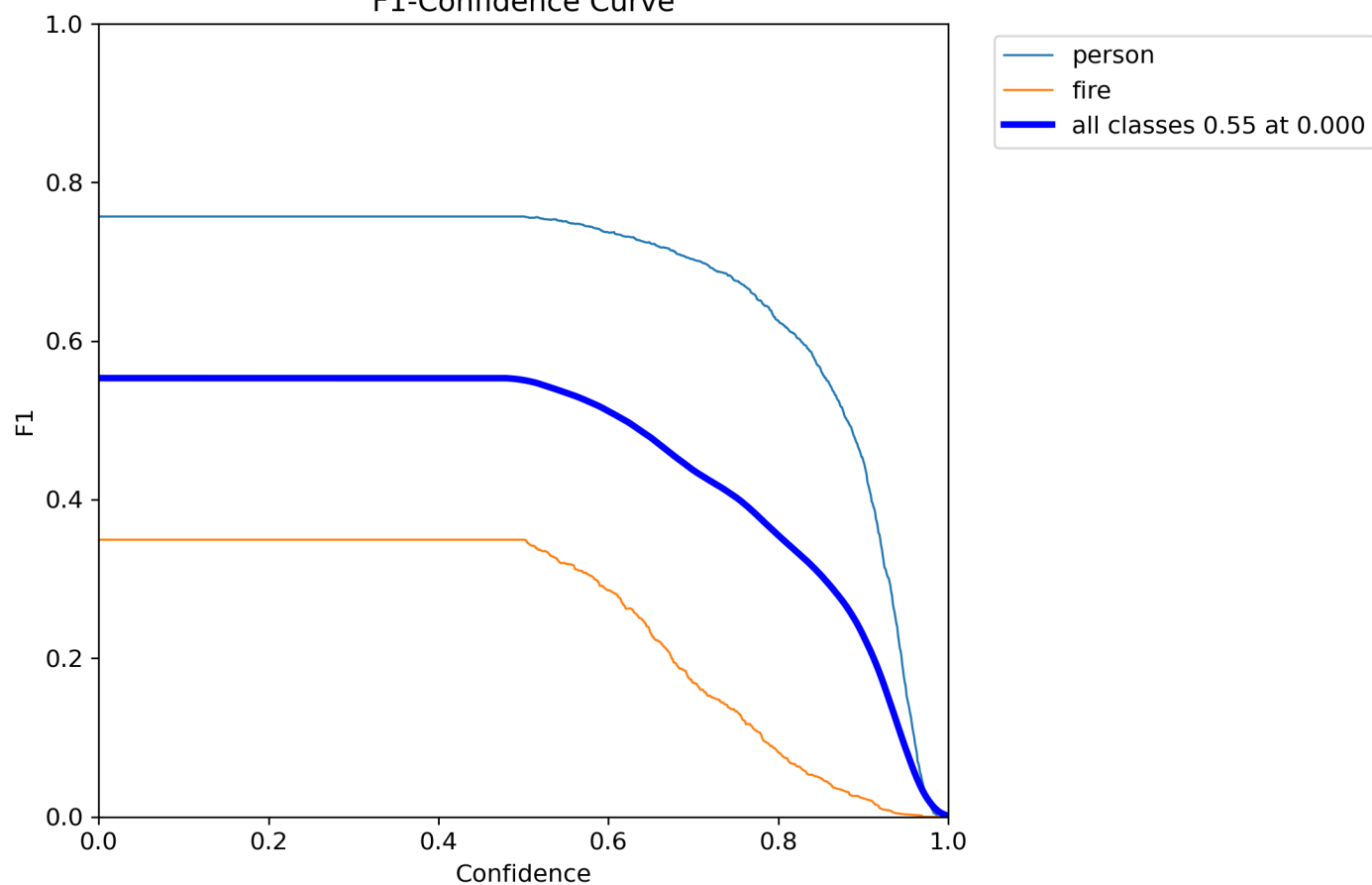
1. YOLOV8模型测试指标结果数据展示

```
1 Evaluation time: 20250625_174135
2 Model path: runs/detect/best.pt
3 Dataset config: datasets/inspection/my_data.yaml
4 mAP50: 0.6318
5 mAP50-95: 0.4735
6 Precision (mp): 0.7959
7 Recall (mr): 0.4530
8 F1-score: 0.5774
9 Test set size: 907
10
11 Per-class results:
12   person(0): AP50=0.7878 AP=0.6419 P=0.8592 R=0.6765 F1=0.7570
13   fire(1): AP50=0.4759 AP=0.3051 P=0.7325 R=0.2294 F1=0.3494
```

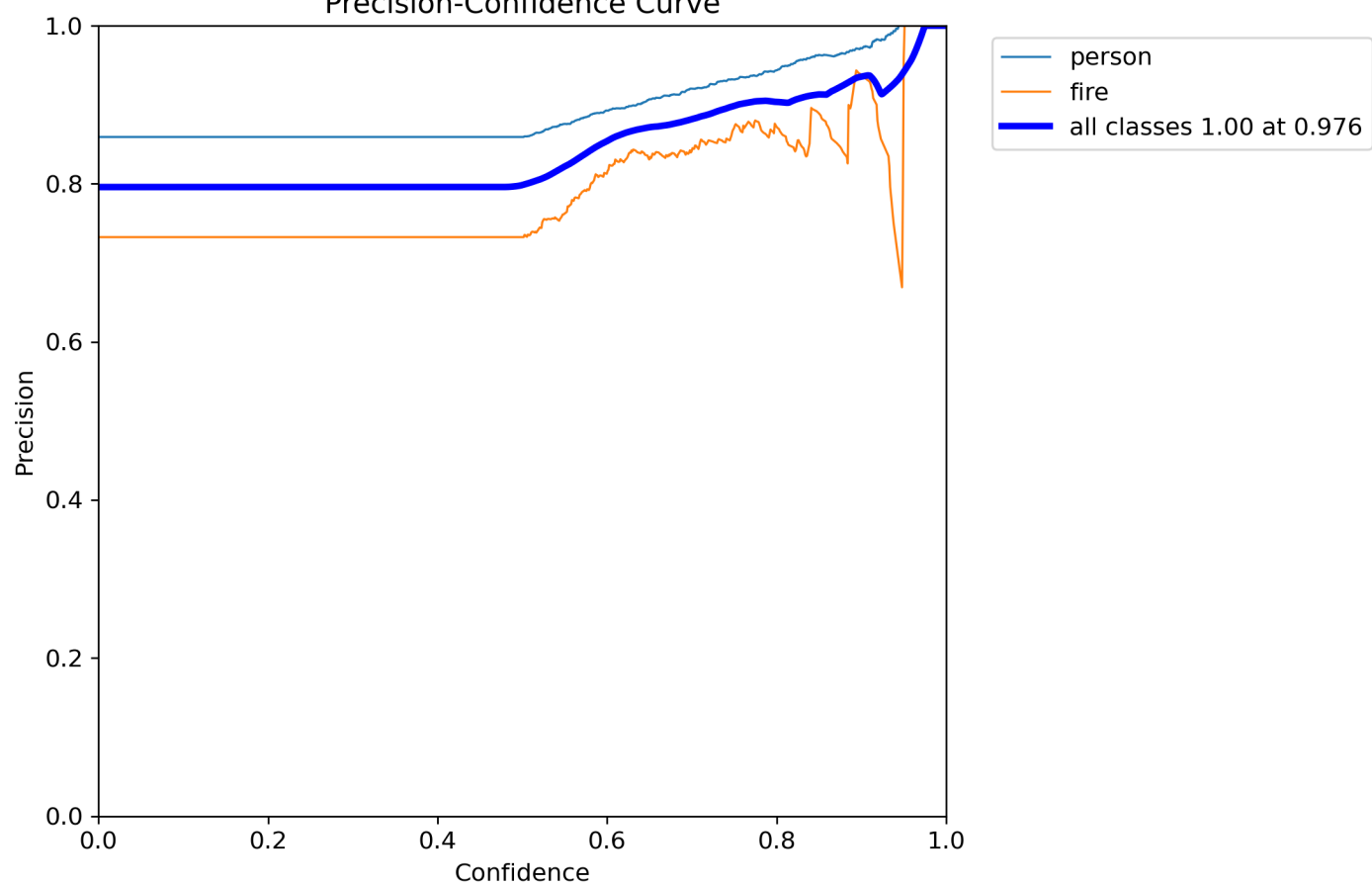
2. PR曲线

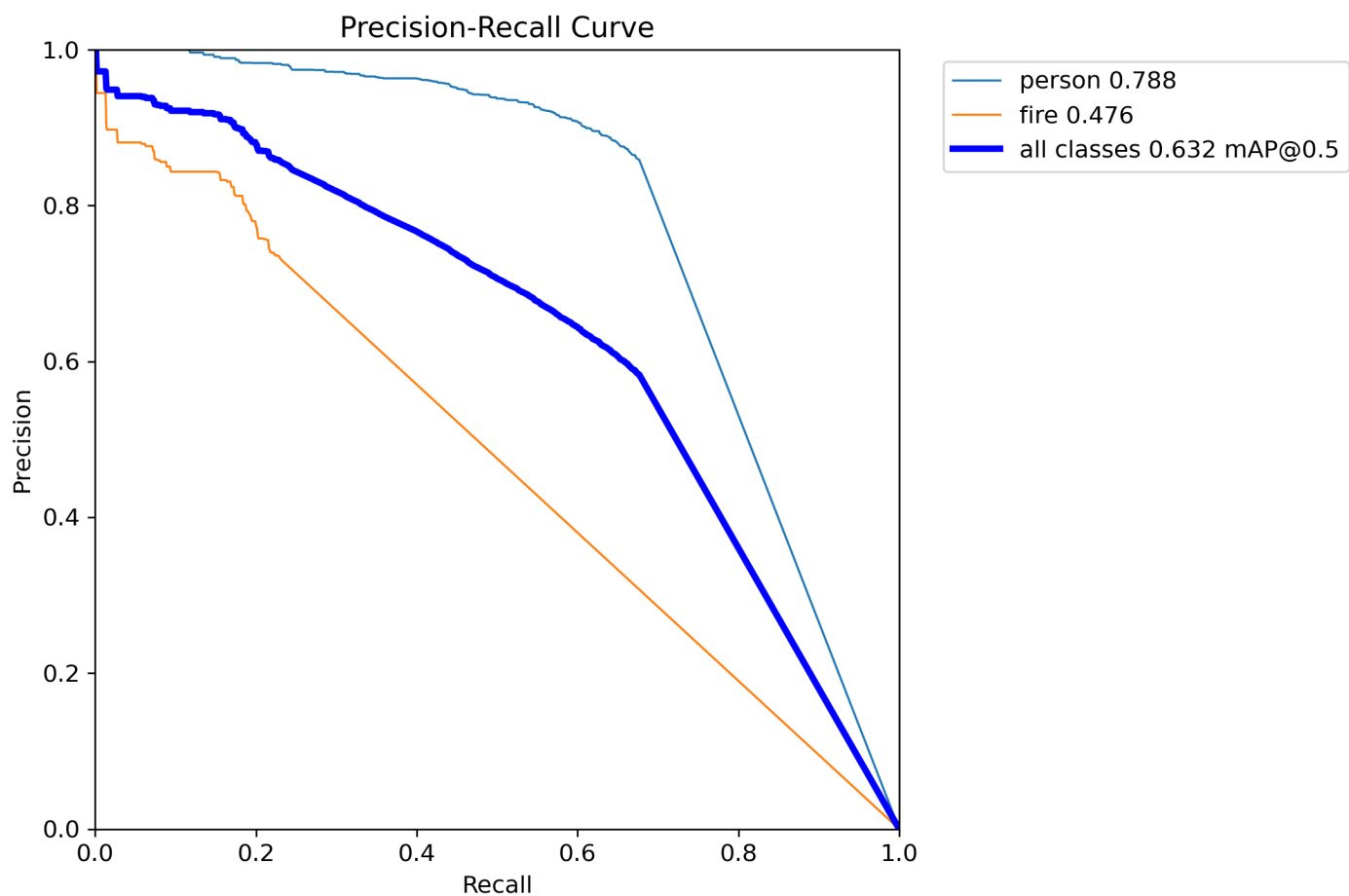


F1-Confidence Curve

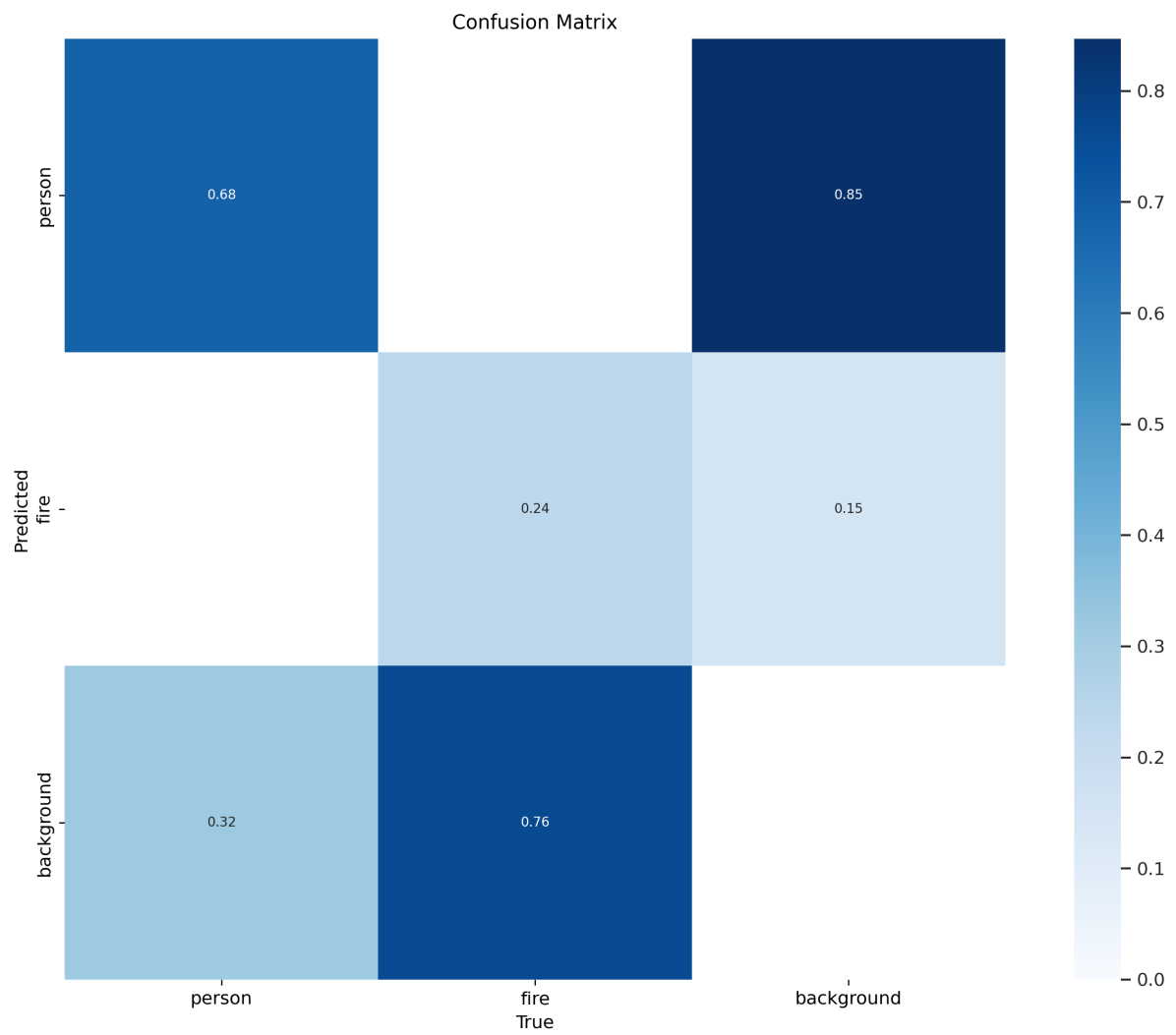


Precision-Confidence Curve

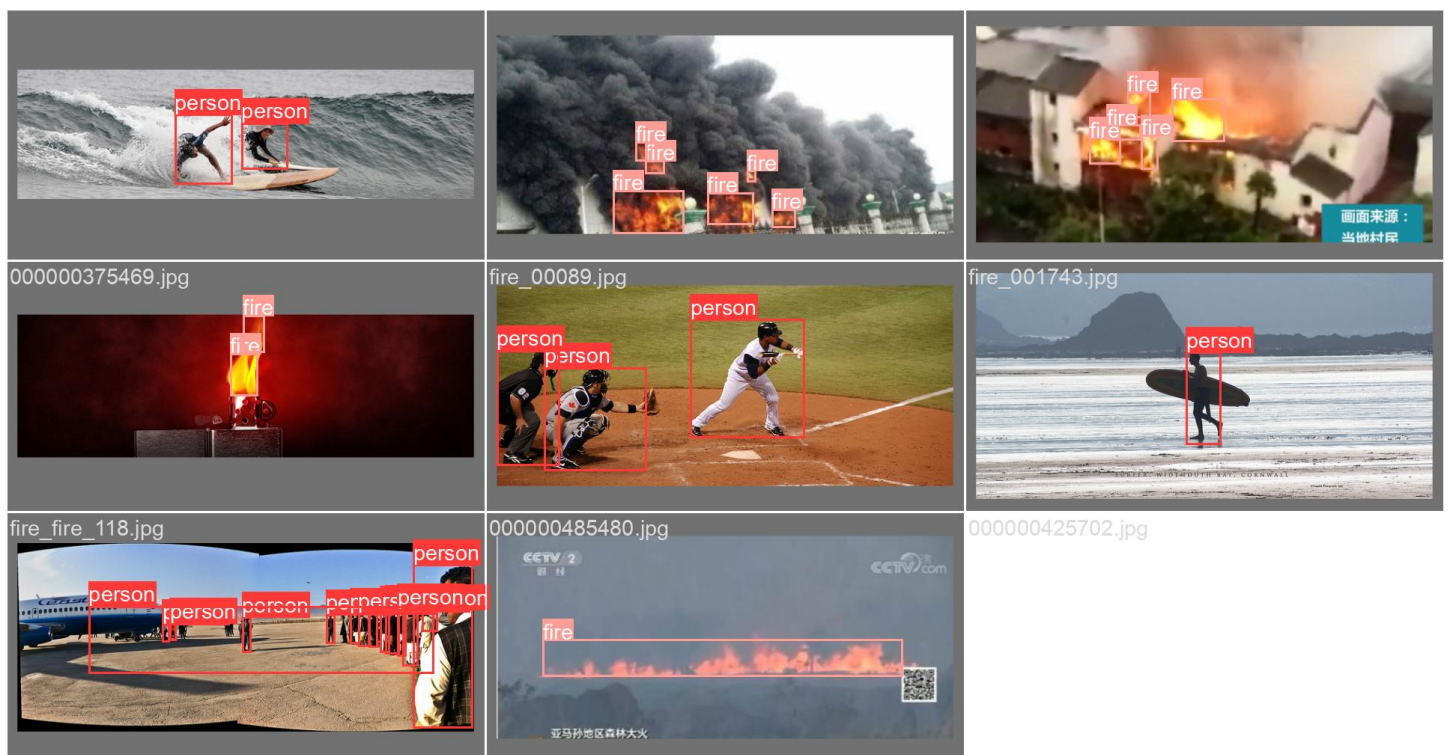


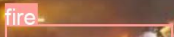
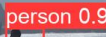
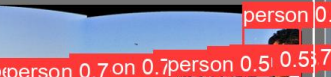


3. 混淆矩阵



4. 检测示例



A photograph of a person walking on a beach, carrying a surfboard. A red bounding box is drawn around the person, and a red label 'person 0.9' is placed above it. The background shows the ocean and mountains.

A large crowd of people is gathered on a city street, many holding up smartphones to record. In the background, a building has a large red sign that reads "fire fine" and another sign that says "911 119". The scene appears to be a public demonstration or protest.

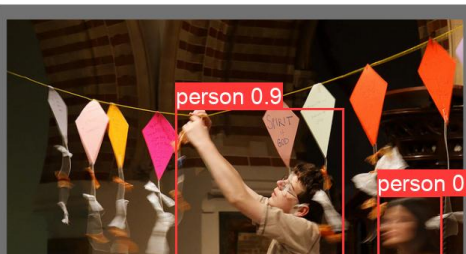


fire_001755.jpg





fire_detection_img(108).jpg



000000235778.jpg



fire_detection_18657d3a-2eb2-40e1-b80e-...



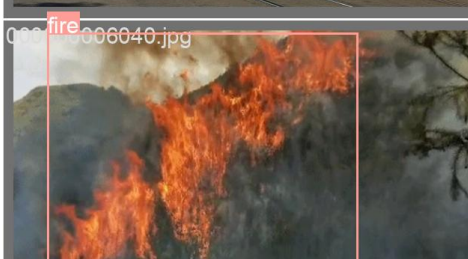
fire_001669.jpg



fire_001755.jpg



fire_00724.jpg



fire_detection_aadca7e8-bf65-47f7-a27a-0...



fire_detection_pic(83).jpg



fire_detection_small(126).jpg



5. 评估摘要

Model Evaluation Metrics best.pt

```
mAP50: 0.6318
mAP50-95: 0.4735
Precision (mp): 0.7959
Recall (mr): 0.4530
F1-score: 0.5774
Test set size: 907
```

Per-class results:

```
person(0): AP50=0.7878 AP=0.6419 P=0.8592 R=0.6765 F1=0.7570
fire(1): AP50=0.4759 AP=0.3051 P=0.7325 R=0.2294 F1=0.3494
```