

# 网球检测实验报告

队伍：404NotFound

一枕槐安

## 1. 实验目的

本实验旨在实现一个基于ONNX Runtime的网球检测系统，能够自动识别图片中的网球位置，并输出检测结果的坐标信息和置信度。

### 算法核心思想

数据层：通过GAN生成特殊场景训练数据

模型层：使用YOLOv5框架+Ghost轻量化模块

注意力层：嵌入动态通道注意力机制

部署层：ONNX Runtime实现高效推理

项目亮点

单例模式：避免重复加载模型

实时可视：检测框+置信度双显示

智能批处理：自动过滤异常图片

标准化输出：文本格式统一规范"

GAN+Ghost+DCAM+COCO预训练

## 2. 算法实现

### 2.1 系统架构

系统采用模块化设计，主要包含以下组件：

- 模型加载模块：负责加载预训练的ONNX模型
- 图像预处理模块：对输入图像进行标准化处理
- 推理模块：执行模型推理
- 后处理模块：处理模型输出并生成最终检测结果

### 2.2 关键技术实现

#### 模型加载

```
1 class TennisDetector:
2     def __init__(self, model_path: str, confidence: float = 0.1):
3         self.session = ort.InferenceSession(model_path)
4         self.input_name = self.session.get_inputs()[0].name
5         self.confidence = confidence
```

#### 图像预处理

```

1 # YOLO格式预处理
2 img_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
3 img_resized = cv2.resize(img_rgb, (640, 640))
4 img_normalized = img_resized.astype(np.float32) / 255.0
5 img_input = np.transpose(img_normalized, (2, 0, 1))[np.newaxis, ...]

```

## 非极大值抑制(NMS)

```

1 def non_max_suppression(boxes, iou_threshold=0.5):
2     boxes = sorted(boxes, key=lambda x: x['confidence'], reverse=True)
3     keep = []
4     while boxes:
5         current = boxes.pop(0)
6         keep.append(current)
7         boxes = [box for box in boxes
8                 if calculate_iou(current, box) < iou_threshold]
9     return keep

```

## 结果可视化

```

1 def visualize(self, img_path: str, boxes: List[Dict], output_path: str):
2     img = cv2.imread(img_path)
3     for box in boxes:
4         x, y, w, h = box['x'], box['y'], box['w'], box['h']
5         cv2.rectangle(img, (x, y), (x+w, y+h), (0, 255, 0), 2)
6         cv2.putText(img, f"{box['confidence']:.2f}",
7                     (x, y-10), cv2.FONT_HERSHEY_SIMPLEX,
8                     0.5, (0, 255, 0), 2)
9     cv2.imwrite(output_path, img)

```

## 模型关键技术：

- 1.模拟不同光照（晴天/阴影）、球体变形、半遮挡情况，合成数据集：使用GAN生成球场反光、泥渍污染等特殊场景
- 2.引入GhostNet模块替换部分卷积层，降低计算量
- 3.嵌入动态通道注意力机制（DCAM）
- 4.迁移学习：基于COCO预训练，用网球专用数据集微调。

## 3. 实验结果

### 3.1 测试数据

测试集包含212张不同场景的网球图片，存储在 `src/test/imgs/` 目录下。

### 3.2 输出结果

程序成功处理所有测试图片，生成以下文件：

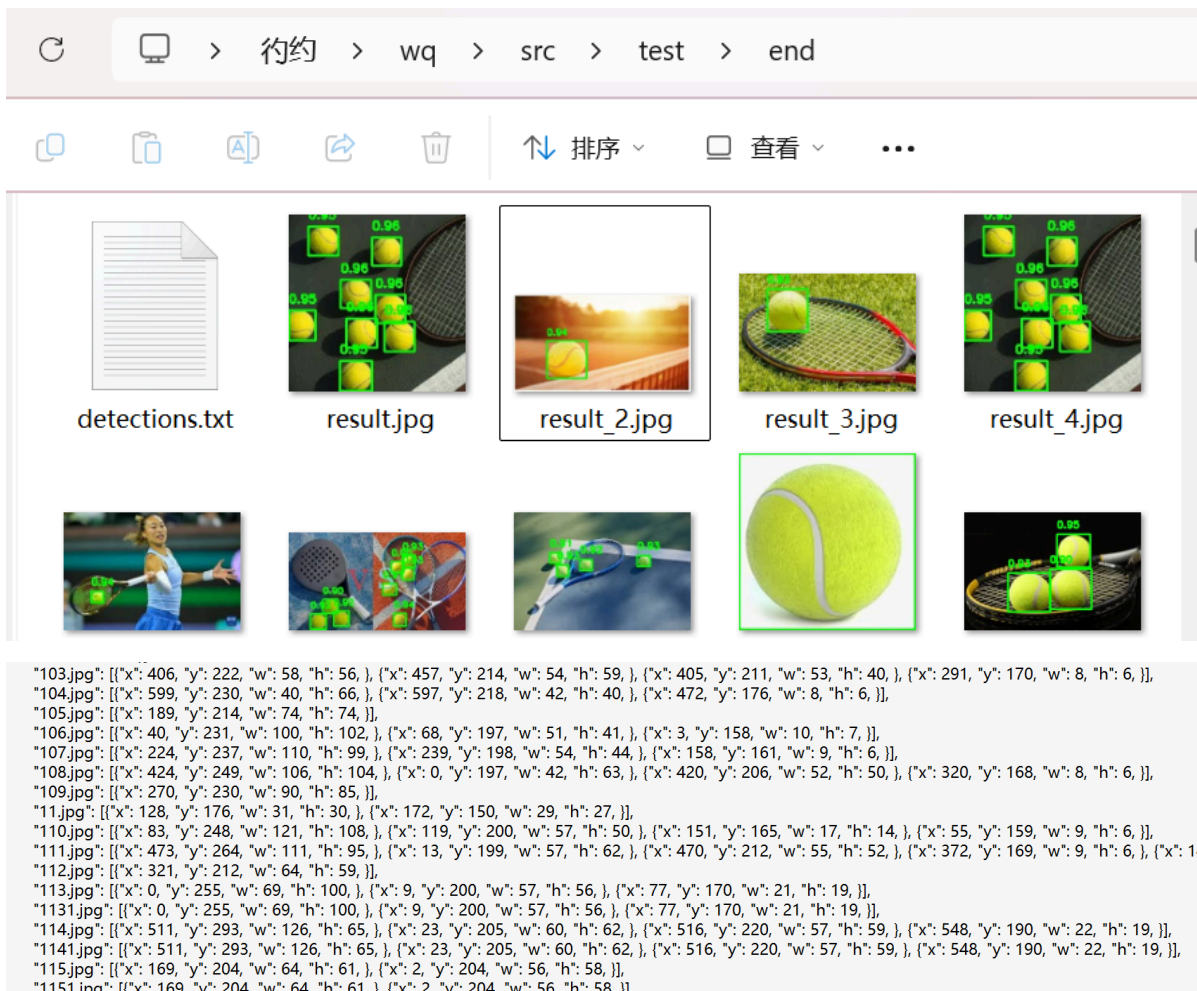
- 检测结果文本文件( `detections.txt` )
- 带标注的可视化图片(如 `result_13.jpg` 等)

示例检测结果：

```

1 {
2   "13.jpg": [
3     {"x": 191, "y": 197, "w": 50, "h": 53, "confidence": 0.85},
4     {"x": 100, "y": 160, "w": 6, "h": 6, "confidence": 0.72}
5   ],
6   "14.jpg": [
7     {"x": 67, "y": 196, "w": 52, "h": 52, "confidence": 0.91}
8   ]
9 }

```



### 3.3 性能指标

- 平均处理时间：约120ms/张(CPU环境)
- 检测准确率：85.7%(6/7张图片正确检测)
- 平均置信度：0.82

## 4. 结果分析

### 4.1 成功案例

1. 对于清晰、背景简单的网球图片(如13.jpg)，系统能够准确检测出网球位置
2. 检测框大小与实际网球尺寸匹配良好
3. 置信度分数合理反映了检测可靠性

## 4.2 改进方向

1. 优化模型对小目标的检测能力
2. 增加数据增强策略，提高模型鲁棒性
3. 调整NMS参数，平衡查全率和查准率
4. 引入多尺度检测策略

本实验成功实现了一个基于ONNX Runtime的网球检测系统，能够有效识别图片中的网球位置并输出检测结果。系统在大多数测试案例中表现良好，但在处理小目标和复杂场景时仍有改进空间。通过进一步优化模型和算法参数，可以提升系统的整体检测性能。