

手势数字识别系统

基于深度学习和计算机视觉的实时手势数字识别

2024/11/26

报告人：肖坤

目录

1. 项目概括

- 系功能与目标
- 技术路线选择
- 整体架构设计

2. MediaPipe实现方案

- 手部关键点检测
- 基于规则的手势识别

3. 自定义实现方案

- 方案设计思路
- 系统架构设计
- 实现效果对比

4. 视觉定位技术

- YCrCb肤色检测
- 形态学处理流程
- ROI提取方法
- 实验效果分析

5. 深度学习模型

- 网络架构设计
- 数据处理流程
- 训练策略优化
- 模型评估结果

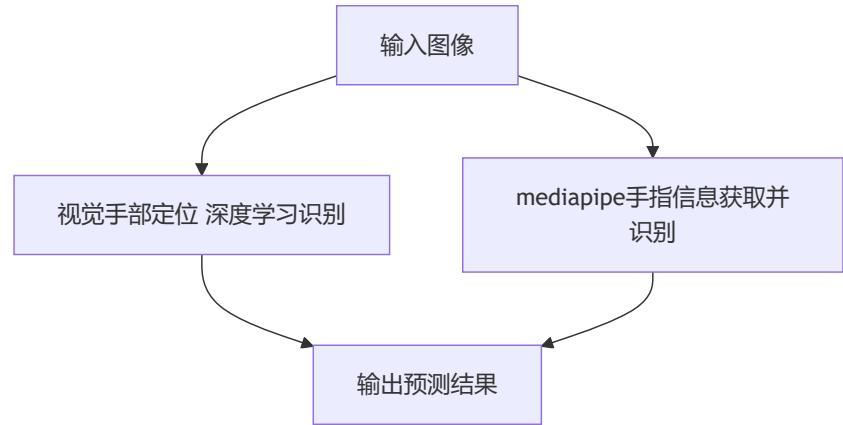
6. 系统总结与展望

- 当前存在的问题
- 改进方向
- 发展前景

系统架构

- UI界面：PyQt6
- 手势识别器接口：抽象基类
- MediaPipe手势识别器
- 自定义手势识别器
- 视频流处理
- 可视化展示

数据流程



将自定义的手势识别器和MediaPipe的识别器组合在一起，对比二者的识别结果，得到最终的识别结果

MediaPipe

Google开源的手部跟踪解决方案

检测21个3D手部关键点，支持多手检测和实时跟踪

应用原理简介

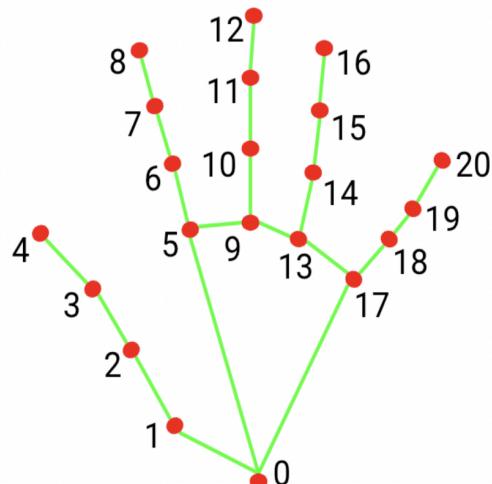
4类关键点：

指尖点 (Fingertips)

远节指关节 (DIP)

近节指关节 (PIP)

掌指关节 (MCP)



- | | |
|-----------------------|-----------------------|
| 0. WRIST | 11. MIDDLE_FINGER_DIP |
| 1. THUMB_CMC | 12. MIDDLE_FINGER_TIP |
| 2. THUMB_MCP | 13. RING_FINGER_MCP |
| 3. THUMB_IP | 14. RING_FINGER_PIP |
| 4. THUMB_TIP | 15. RING_FINGER_DIP |
| 5. INDEX_FINGER_MCP | 16. RING_FINGER_TIP |
| 6. INDEX_FINGER_PIP | 17. PINKY_MCP |
| 7. INDEX_FINGER_DIP | 18. PINKY_PIP |
| 8. INDEX_FINGER_TIP | 19. PINKY_DIP |
| 9. MIDDLE_FINGER_MCP | 20. PINKY_TIP |
| 10. MIDDLE_FINGER_PIP | |

mediapipe判断规则

手指状态判断

- 拇指特殊处理:
 - 使用MCP、PIP和指尖三点 仅考虑共线性 ($\text{mse} < 0.001$)
- 其他手指:
 - 使用PIP、DIP和指尖三点 考虑共线性和位置关系

共线性判断

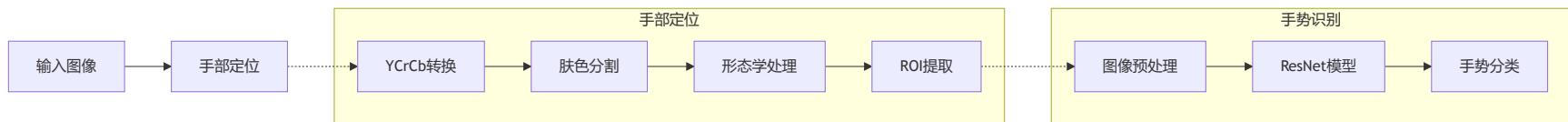
```
def calculate_collinearity(p1, p2, p3):  
    # 最小二乘法拟合直线  
    points = np.array([[p1.x, p1.y], [p2.x, p2.y], [p3.x, p3.y]])  
    x = points[:, 0]  
    y = points[:, 1]  
    A = np.vstack([x, np.ones(len(x))]).T  
    m, c = np.linalg.lstsq(A, y, rcond=None)[0]  
    # 计算点到直线距离均方误差  
    distances = (y - (m * x + c)) ** 2  
    return np.mean(distances)
```

自定义识别方案 - 总体设计

设计思路

- 采用传统CV+深度学习的混合方案
- 分为两个主要阶段：
 1. 手部定位：基于肤色的传统视觉方法
 2. 手势识别：基于深度学习的分类模型

技术路线



自定义识别方案 - 手部定位原理

YCrCb颜色空间原理

- 颜色空间转换：
 - Y：亮度分量 (Luminance)
 - 范围：0-255
 - 表示图像的明暗程度
 - Cr：红色色度分量 (Red Chrominance)
 - 范围：133-173 (肤色)
 - 表示红色偏移度
 - Cb：蓝色色度分量 (Blue Chrominance)
 - 范围：77-127 (肤色)
 - 表示蓝色偏移度

形态学处理原理

1. 闭运算 (Close)
 - 先膨胀后腐蚀
 - 填充小孔洞
 - 连接断开区域
2. 开运算 (Open)
 - 先腐蚀后膨胀
 - 去除噪点
 - 平滑边界
3. 膨胀 (Dilate)
 - 扩目标区域
 - 增强连通性

自定义识别方案 - 手部定位实现

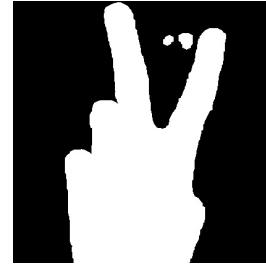
1. 原图



3. 掩码



5. 膨胀结果



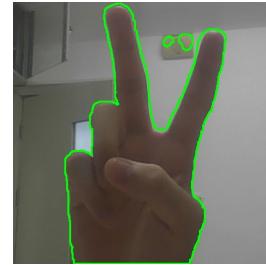
2. YCrCb 转换



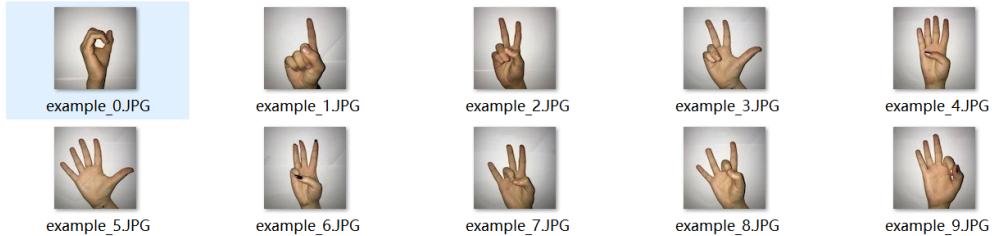
4. 开闭结果



6. 最终结果



自定义识别方案 - 深度学习模型



数据集处理

- Sign Language Digits Dataset
 - 土耳其数字手势
 - 2062张手势图片
 - 灰度图像 (64x64)

数据增强策略

```
transforms.Compose([
    transforms.RandomRotation(15),      # 随机旋转
    transforms.RandomAffine(
        degrees=0,
        translate=(0.1, 0.1),           # 平移范围
        scale=(0.9, 1.1),              # 缩放范围
    ),
    transforms.RandomHorizontalFlip(p=0.3),
])
```

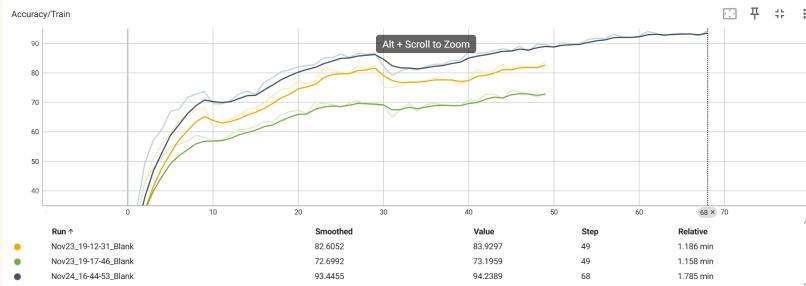
自定义识别方案 - 训练过程

网络结构

- 改进的ResNet架构
- SE注意力模块
- 混合精度训练

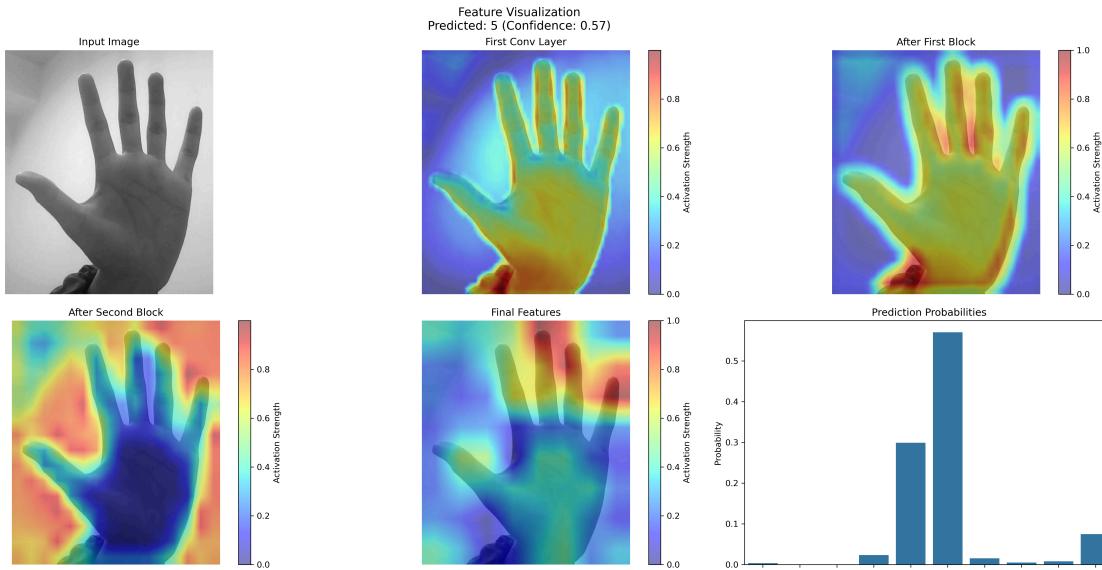
训练策略

- 优化器: Adam($lr=0.001$)
- 学习率调度: 余弦退火
- 早停机制 ($patience=15$)



模型在训练和验证时的表现(tensorboard)

模型特征分析



红色为关注度最高 蓝色为关注度最低

特征提取分析

浅层特征捕捉边缘和轮廓信息

深层特征捕捉手势结构

关键区域（手指）反应较强

confidence柱状图也显示了模型对每个类别的置信度

可以看到 模型对4和5的置信度较高
这样可以看出 模型对手指的识别效果较好

演示

系统总结与展望

当前存在的问题

1. 肤色检测的局限性:

- 人脸等其他肤色区域也会被错误检测
- 对光照条件敏感，容易受环境影响
- 复杂背景下误检率增加

2. 手势识别的挑战:

- 相似手势（如7和8）难以区分
- 视角变化导致识别不稳定
- 部分遮挡情况下准确率下降

改进方向

1. 技术改进:

- 使用更加鲁棒的肤色检测方法
- 使用更加复杂的手势数据集
- 在数据增强阶段优化数据集 增强模型泛化能力

谢谢聆听