基于计算机视觉的手势识别系统

目　录

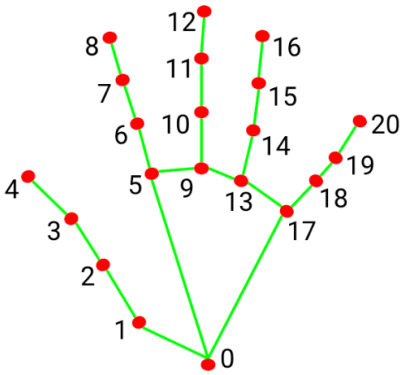
1 手势识别

1.1 Mediapipe 框架

MediaPipe 是谷歌在 2019 年开源的一种基于机器学习技术的手势识别算法，其特点是准确率高，对手部进行手势追踪，可根据一帧图像检测并推断出手部 21 个骨骼关键点，如图 1-1 所示。与当前的手势识别技术相比，谷歌的 MediaPipe 框架不仅可以使用台式机，还能使用手机来进行实时检测追踪，并支持多手追踪，做到手部遮挡识别。MediaPipe 使用深度学习方法，对骨骼关键点检测模型提取的关键点数据进行处理和分析，以实现手势识别。它也可识别基础手语，在 AR/VR 中应用于手势操控等。

MediaPipe 框架支持在实时视频流中进行骨骼关键点捕捉和手势识别。在数据集稀缺情况下，使用骨骼关键点具有较好的泛化性，并且较小的计算量就可进行识别。

本文使用了 Mediapipe 框架的 Hands 模块，其主要采集人手的骨骼关键点，包括各个手指的指骨关节以及每只手的腕关节的全部 21 个关键点，其中每个关键点都将输出 3 个值，分别为对应关键点的 x、y、z。



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

图 1-1 手部 21 个关键点

通常在使用 MediaPipe 时，需要使用 OpenCV 读取视频文件并将视频帧作为输入传递给 MediaPipe。MediaPipe 会对每一帧图像进行监测，如果检测到手形或类似手型的目标，则会推测这些目标的骨骼关键点的坐标位置，并输出相关信息，其中包含左右手的 label、每个关键点的 label、序号和坐标信息。

1.2 手势特征提取

手势特征可划分为距离和角度。通过判断手指弯曲情况、手指开合状态可以区分不同的手势。根据 Mediapipe 的 Hands 模块检测到的 21 个手部关键点二维坐标，计算手指间的距离、手指的角度、手指到手腕的距离比这三种几何特征，以实现对手指弯曲情况、手指开合状态的判断。

1.2.1 关键点间距离

为了消除手部与屏幕之间的距离对手势识别的影响，首先计算出一个距离元作为标定值，再以此距离元来衡量其它关键点之间的距离。距离元选择为手部关键点 5 和点 17 之间的欧式距离l_0，

$$l_0 = \sqrt{(x_5 - y_5)^2 + (x_{17} - y_{17})^2}$$

关键点间的距离l可用两关键点间的欧式距离l_1除以距离元l_0：

$$l = \frac{l_1}{l_0}$$

1.2.2 手指弯曲角度

手指弯曲角度为计算两条关键点连线间的夹角：

$$\theta = \arccos(l, l')$$

手掌角度θ_1为固定计算手部关键点 5 和点 17 的连线与 x 轴的夹角：

$$\theta_1 = \arctan((x_5, y_5), (x_{17}, y_{17}))$$

1.2.3 手指开合判断

如图 1-2 所示是判断大拇指开合情况的示意图。大拇指的开合由两向量 A、B 的夹角θ_0判断，θ_0的大小可反映大拇指弯曲程度。根据多次实验得出，角度θ_0以 153 度为阈值，大于或等于 153 度判断为张开，小于 153 度为闭合。

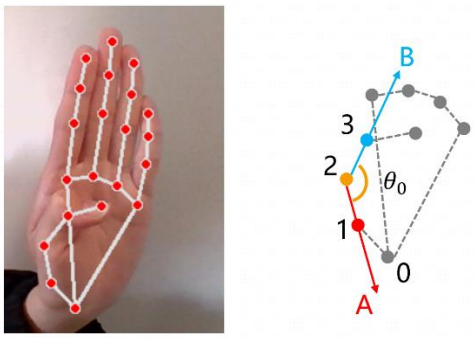


图 1-2 拇指开合情况示意图

对于其余四个手指的开合情况判断，以无名指为例，如图 1-3 所示是判断无名指开合情况的示意图，其开合由手指近节指关键点 13 和手指指头关键点 16 分别与手腕关键点的距离的$l_{0,13}$、$l_{0,16}$大小进行判断，$l_{0,13}$小于$l_{0,16}$则判断为手指张开，否则判断为闭合。

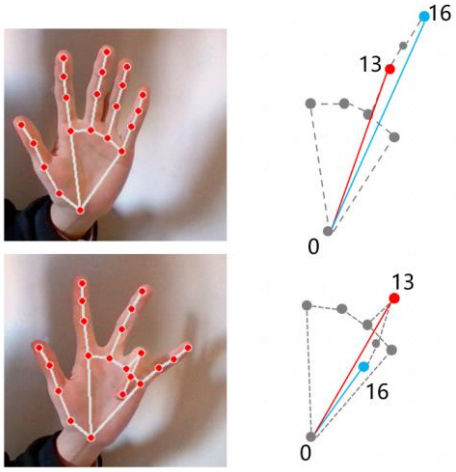


图 1-3 无名指开合情况示意图

为了让后续对手势的判断更便捷，将手指的开合两种状态转化为二元特征 0 和 1，最终一只手的手指开合状态可用一个长度为 5 的一维数组表示，例如，当手势为“剪刀”时，只有食指和中指张开，其它手指均闭合，则其对应状态可表示为[0,1,1,0,0]。

2 交互系统设计

本文设计出一个手势交互系统，对不同的手势，映射到计算机设备上的不同指令。程序流程图如图 2-1 所示。

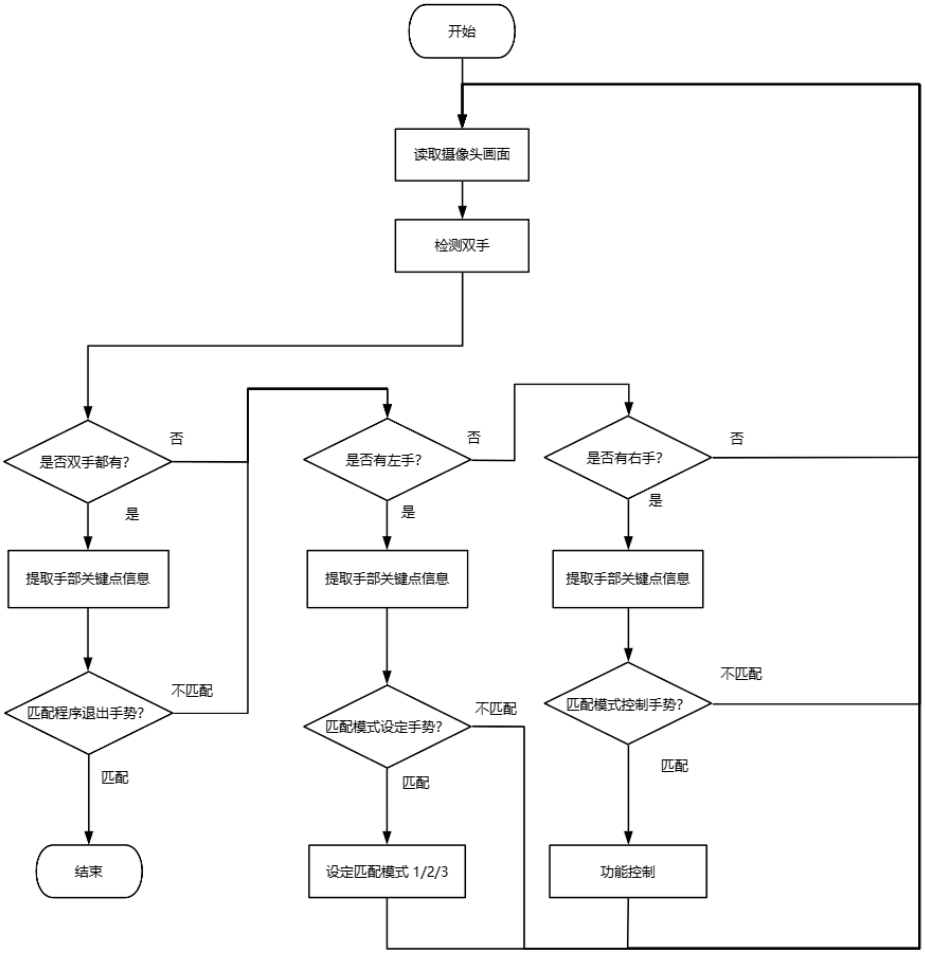


图 2-1 程序流程图

本文共设计了三个交互模式，对应手势如表 2-1 所示。

表 2-1 三种交互模式

模式功能	左手手势状态	模式功能
亮度音量控制	[0,1,0,0,0]	亮度音量控制
光标控制	[0,1,1,0,0]	光标控制
手语字母识别	[0,1,1,1,0]	手语字母识别

交互系统的窗口界面如图 2-2 所示，包含画面实时帧率、当前设定模式和实时检测到的手型等文字说明。

图 2-2 系统界面设计效果

2.1 模式选择

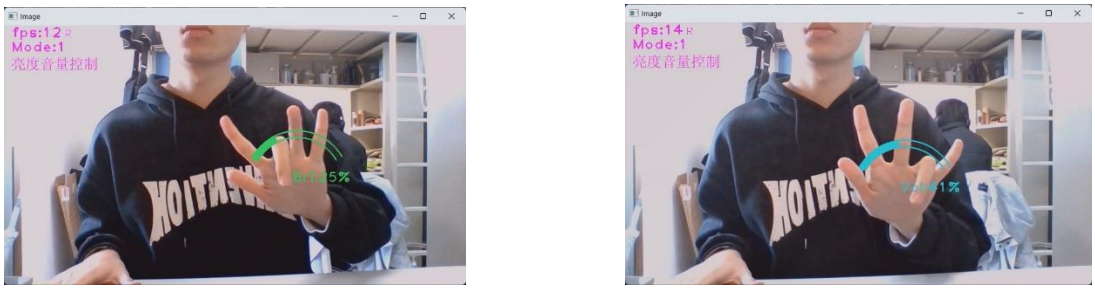
模式选定与切换由左手手势控制，模式内具体功能控制由右手手势控制。模式 1、2、3 的选择分别对应左手的食指张开，食指拇指张开，食指拇指中指张开。切换模式时需保持模式对应的手势 3 秒钟不变，以避免非设定模式情况下对系统的误触，在画面中将设定模式时的 3 秒钟可视化为一个环形进度条，效果如图 2-3 所示。

图 2-3 模式设定效果展示

2.2 亮度音量控制

模式 1 为亮度音量控制，控制笔记本电脑显示屏亮度和媒体音量，右手手掌角度θ_1与亮度和音量成线性关系，手掌角度θ_1有效范围为$[-40°, 80°]$，对应控制亮度和音量的范围为$[0,100]$。定义亮度控制手势状态为$[1,1,0,1,1]$，音量控制手势状态为$[1,1,1,0,1]$。

为了让控制给出有效的反馈，本文将亮度和音量的实时设置的数值可视化，显示在摄像头返回的画面上，效果如图 2-4 所示。



（a）亮度控制　　（b）音量控制

图 2-4 亮度音量控制效果

2.3 光标控制

模式 2 为光标控制，模拟笔记本电脑鼠标的移动和点击，定义光标控制手势状态为[0,1,1,0,0]。

当手在摄像头的画面边缘移动时，系统可能会因画面中的手部不完全而无法检测到手的存在，为避免手在画面边缘时对控制的影响，以及保证手在控制光标移动时能达到计算机屏幕边缘，设定一个小于摄像头画面大小的控制区域，控制区域与画面边缘之间的距离为边界宽度，根据多次实验，将边界宽度设置为 80。

控制光标的移动时，食指和中指的指尖坐标的中心值c作为输入，根据控制区域和显示屏的像素比进行线性映射，实现光标在整个屏幕区域的移动。

为实现光标的按下和释放两种状态，此模式根据食指和中指的指尖间的距离$l_{8,12}$区分，当距离$l_{8,12}$小于 0.4 时，鼠标左键按下，大于 0.9 时，鼠标左键释放。左键按下的同时可以控制光标的移动，实现对不同应用程序窗口的拖拽和文件的移动。效果如图 2-5 所示。

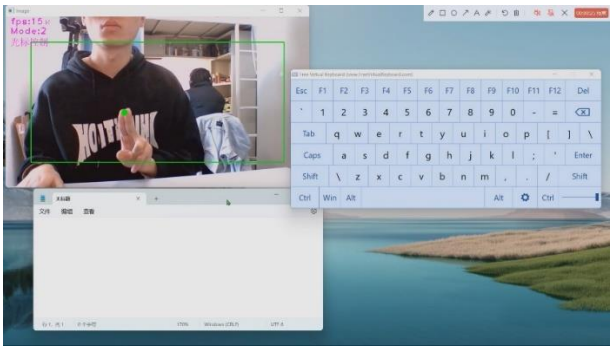


图 2-5 光标控制效果

此外，为实现模拟键盘外设输入字符的功能，系统添加了虚拟键盘小程序，当系统处于模式 2，且右手手势状态为[0,1,1,0,0]时，系统会打开虚拟键盘小程序，当光标聚焦在文本框上，通过点击虚拟键盘上的字符，即可输入相应的字符，效果如图 2-6 所示。

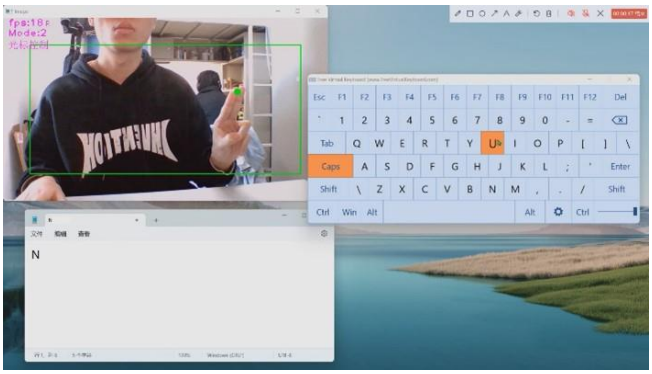


图 2-6 虚拟键盘字符输入效果

2.4 手语字母识别

模式 3 为手语字母识别，此模式参照《汉语手指字母方案》，实现对 A-Z、CH、SH、ZH 和 NG 共 30 个字母的识别。为区分不同字母对应的手势，本系统先通过手指的开合状态对字母进行分组，分组情况如表 2-2 所示。

表 2-2 手势与手语字母对应分组表

手指开合状态	对应字母
[1, 0, 0, 0, 0]	A, C, O, Q, SH, S
[0, 1, 1, 1, 1]	B
[0, 0, 0, 0, 0]	D, M, N
[0, 0, 1, 1, 1]	E
[0, 1, 0, 0, 0]	G, I, J
[0, 1, 1, 0, 0]	F, H, V, X
[1, 1, 1, 0, 0]	K
[1, 1, 0, 0, 0]	L, R
[0, 1, 0, 0, 1]	T, Z
[0, 1, 1, 0, 1]	ZH
[1, 1, 1, 1, 1]	P, CH, U
[1, 0, 0, 0, 1]	Y
[0, 1, 1, 1, 0]	W
[0, 0, 0, 0, 1]	NG

分组后，对于一种手指开合状态对应一个手语字母的组别，可直接识别出手语字母；对于一种手指开合状态对应多个手语字母的组别，需要进一步对手指指式区分，以手指开合状态为[1, 0, 0, 0, 0]为例，其对应 A, C, O, Q, SH, S 共 6 种可能的字母，其对应指式如图 2-7 所示。

图 2-7 六种手语字母对应指式

首先根据中指和无名指指尖距离分类，若中指与无名指尖相贴（距离小于 0.45），则可能为 A、C、O、S，相反，二者不相贴时，则可能为 Q、SH，对于 A、C、O、S，根据拇指与食指的距离可分离出 A 或 S、C、O 三组，再根据手掌的朝向（向上或向右）进一步分辨出 A 和 S，对于 Q 和 SH，同样根据拇指与食指的距离区分，至此，同一组手指开合状态的 6 中手语字母被区分开。以此类推，其它包含两个及以上手语字母的组别依据类似的规则也可区分开。

当系统识别出手语字母时，会在画面左上角显示识别出的手语字母，同时在画面左下角给出字母指式参考图片，效果如图 2-8 所示。

图 2-8 手语字母识别效果

2.5 系统退出

除前文的三种功能模式外，本文还设计了一个退出手势用于退出此交互系统。当左右手的手指开合状态均为[1,1,1,1,1]，且保持手势不变 5 秒，系统会自动退出，效果如图 2-9 所示。退出程序优先级最高，在任何模式下都可激活退出程序。

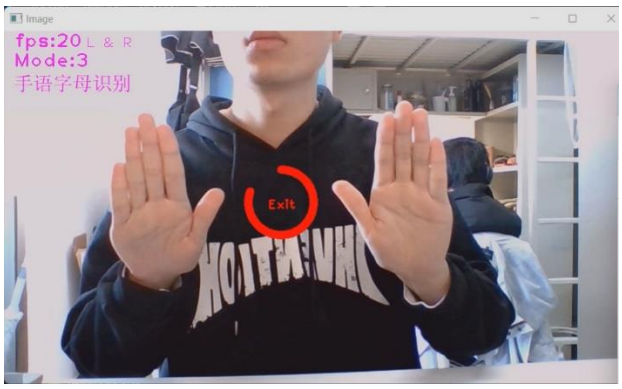


图 2-9 退出程序效果

3 系统测试及分析

为测试系统对手势识别的效果，本文选择使用模式 3 中的 30 中手语字母对应的手势对系统进行测试。作者邀请两位同学拍摄记录 30 种不同的手势，对每张图像裁剪，仅保留手部区域，再对手势图像通过高斯模糊、调亮、调暗、提高对比度的形式扩充数据集，最终得到 2100 张手势图像。数据集如图 3-1 所示。

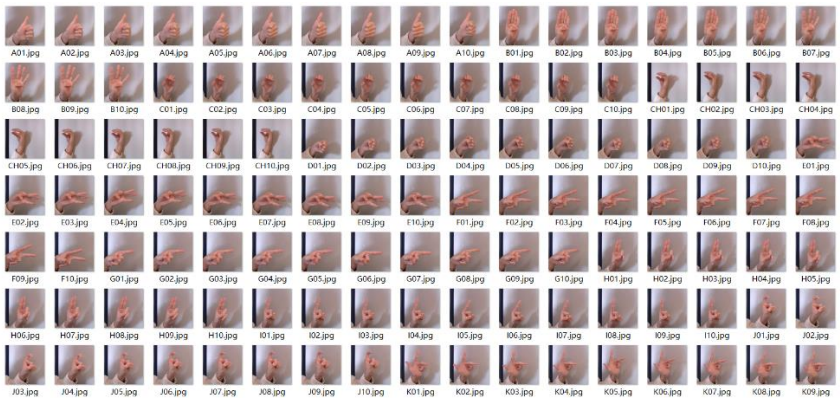


图 3-1 手语字母识别测试数据

随后使用此系统对 2100 张手势图像进行识别并保存识别结果，将识别结果统计汇总，最终得到各手势识别的混淆矩阵，如图 3-2 所示。

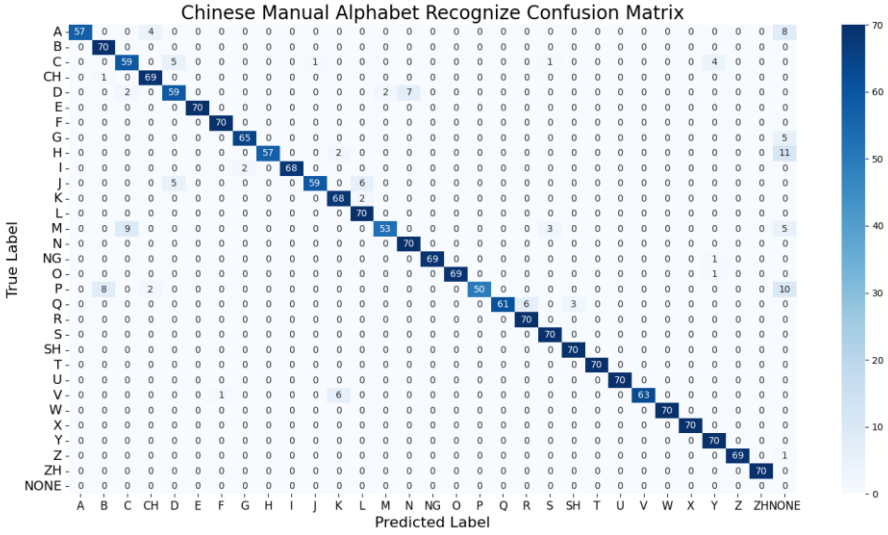


图 3-2 汉语手指字母识别结果混淆矩阵

采用准确率 ACC、精确度 P、召回率 R 和 F1 综合评价指标来评估手势识别的算法性能：

$$ACC = \frac{TP + TN}{TP + FP + FN + TN}$$

$$P = \frac{TP}{TP + FP}$$

$$R = \frac{TP}{TP + FN}$$

$$F1 = \frac{2AR}{A + R}$$

其中，TP 为真正类，FN 为假负类， FP 为假正类，TN 为真负类。

最终计算得准确率 ACC 为 94.3%，精确度 P 为 93.08%，召回率 R 为 91.05%，F1 为 91.8%。

4 结论

本文研究并设计了一个基于计算机视觉的手势识别系统，使用 Mediapipe 框架识别手势，并在此之上设计了人机交互功能，实现计算机设备的亮度音量控制、光标控制与手语字母识别功能，并对系统的识别效果进行测试。根据测试结果可知，此系统能很好地完成设计的功能，且对手语字母有不错的识别效果。但仍存在一些可改进的地方，如手语字母识别会出现无法匹配的情况，以及整体识别率有待进一步提升。