

基于 MediaPipe 构建的手势交互系统

《人工智能课程》期末大作业



2022 年 6 月 12 日

基于 MediaPipe 构建的手势交互系统

摘 要

手势是一种自然而直观的人际交流模式，基于视觉的手势识别是实现新一代人机交互所不可缺少的一项关键技术。然而由于手势本身具有的多样性、多义性、以及时间和空间上的差异性等特点加之人手是复杂变形体及视觉本身的不适定性因此基于视觉的手势识别是一个极富挑战性的多学科交叉研究课题

同时，近年来随着机器视觉和深度学习的发展，涌现出一系列手势识别的算法和框架。为了探究基于 MediaPipe 的手势识别算法在人机交互中的应用，本文将手部不同手势与电脑系统的不同操作相对应，实现一个简单的手势交互系统。

采用 MediaPipe 对 15 种手势类型进行实时静态识别，生成指令，传输信号从而实现控制电脑鼠标键盘进行视频操控、图像绘制、ppt 切换等实际操作。结果显示，该系统具有良好的性能。

关键词：手势识别；MediaPipe；人机交互；计算机视觉；深度学习

Gesture interaction system built on MediaPipe

ABSTRACT

Gesture is a natural and intuitive mode of interpersonal communication, and vision-based gesture recognition is a key technology indispensable for the realization of next-generation human-computer interaction. However, because of the diversity, multiple meanings, and temporal and spatial variability of gestures, the human hand is a complex deformation and the discomfort of vision itself, vision-based gesture recognition is a challenging multidisciplinary interdisciplinary research topic.

Meanwhile, with the development of machine vision and deep learning in recent years, a series of algorithms and frameworks for gesture recognition have emerged. In order to explore the application of MediaPipe-based gesture recognition algorithms in human-computer interaction, this paper implements a simple gesture interaction system by corresponding different hand gestures to different operations of a computer system.

MediaPipe is used to statically recognize 15 gesture types in real time, generate commands, and transmit signals so as to control the computer mouse and keyboard for video manipulation, image drawing, ppt switching and other practical operations. The results show that the system has good performance.

Key words: Gesture recognition; MediaPipe; human-computer interaction; computer vision; deep learning

目录

1	项目简介	5
1.1	项目背景	5
1.2	项目目的	5
1.3	项目功能	5
2	项目方案设计	6
2.1	总体设计思路与总体架构	6
2.1.1	手势识别模型搭建部分思路说明	6
2.1.2	手势识别模型搭建部分总体架构	6
2.1.3	UI 界面部分图示	7
2.2	核心模型及其基本原理	8
2.3.2	功能模块设计说明	10
2.4	其他创新内容或优化算法	11
2.4.1	手势的检测与分割	11
2.4.2	多场景、多功能集成化	11
2.4.3	合理使用 html+css 搭建网页前端	11
3	实验过程	12
3.1	环境说明	12
3.2	源代码文件与主要函数清单	12
3.2.1	手势识别部分文件	13
3.2.2	功能实现	13
3.2.3	网页部分文件	13
3.2.4	主要函数清单	13
3.3	实验功能	14
3.3.1	视频控制	14
3.3.3	手势绘图	15
3.3.2	ppt 和 word 控制	16
4	总结	18
4.1	实验总结	18
4.2	实验中问题及解决方案	18
4.2.1	手指控制鼠标移动卡顿	18
4.2.2	手势检测速度过快导致动作重复发生	18
4.2.3	类似手势之间用户无意识切换带来的错误	18
4.3	后续改进方向	18
4.4	心得体会	19
5	参考文献	20
6	成员分工与自评	20

1 项目简介

1.1 项目背景

当今时代，处于第三波浪潮的 AI 正迎来加速发展时期，AI 技术不断突破，应用大规模爆发。计算机视觉作为人工智能行业的最大组成部分，无论是在技术成熟度、商业化进程，还是在市场增长速度、投融资热度等方面，自始至终都是人工智能领域最热门的行业之一。

人与计算机的交互活动越来越成为人们日常生活的一个重要组成部分。特别是最近几年随着计算机技术的迅猛发展研究符合人际交流习惯的新颖人机交互技术变得异常活跃也取得了可喜的进步。这些研究包括人脸识别、面部表情识别、唇读、头部运动跟踪、凝视跟踪、手势识别、以及体势识别等等。总的来说人机交互技术已经从以计算机为中心逐步转移到以人为中心是多种媒体、多种模式的交互技术。

基于视觉的手势识别研究正是顺应了这一潮流。手势交互是一种重要的自然交互方式，因而手势识别成为人机交互的一个重要分支领域。近些年来，国内外的研究机构曾成功开发了大量能够实现遥操作的设备和算法，目前存在一些将手势识别和遥操作结合的相关研究，大致可分为基于穿戴设备的手势识别和基于视觉的手势识别两大类。后者又可分为静态手势识别和动态手势识别两类，均以手势图像作为研究对象，区别在于静态研究时间节点上的图像，动态识别时序图像。

1.2 项目目的

本项目基于《人工智能课程》的教学内容，在此前认真完成 PA1A*实现八数码、PA2 剪枝算法五子棋、PA3 Horn 子句归结的基础上，选取了手势识别作为计算机视觉以期于对于人工课程知识的扩展认知。

本文针对传统的使用鼠标键盘等系列系统操作，提出一种基于 MediaPipe 机器视觉的手势识别用于操控系统的方法。给观看视频带来更好体验感与娱乐性的同时，对识别的准确性进行一定的优化，探究更人性化的人机交互形式。

1.3 项目功能

通过实时摄像头捕捉手势操作，利用框架对 15 种手势类型进行实时静态识别，生成指令，传输信号从而实现控制电脑鼠标键盘进行视频操控、图像绘制、无接触控制 ppt 放映、切换等操作等实际操作。

2 项目方案设计

2.1 总体设计思路与总体架构

总体设计思路可以分为手势识别模型搭建以及具体功能实现的设计两大部分，两部分之间的交互如图 1 所示：

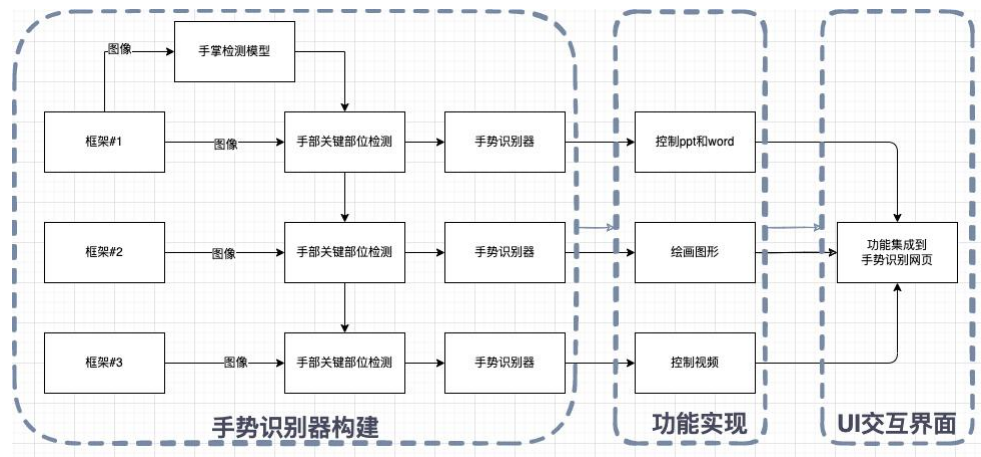


图1 总体架构示意图

2.1.1 手势识别模型搭建部分思路说明

基于手势识别系统的搭建需要经过三步:手势检测、手势跟踪、手势识别，基于 MediaPipe 的手势识别同样包含手掌检测模型、手部关键部位模型、手势识别器三个部分，其中：手掌检测模型，用于从图像中识别手掌，定义手部边界区域；手部关键部位模型，用于对手部边界区域检测并返回 3D 手部关键点；手势识别器，则用根据手部关键点定义并识别各种手势。

2.1.2 手势识别模型搭建部分总体架构

手势识别模型搭建部分的总体架构图如图 2：

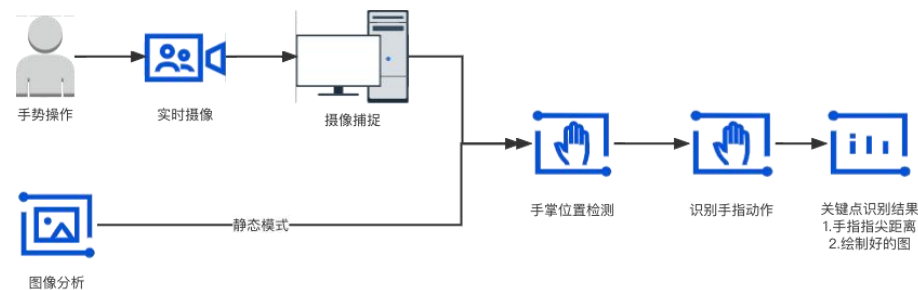


图 2 手势识别模型搭建部分的总体架构图示

2.1.3 UI 界面部分图示

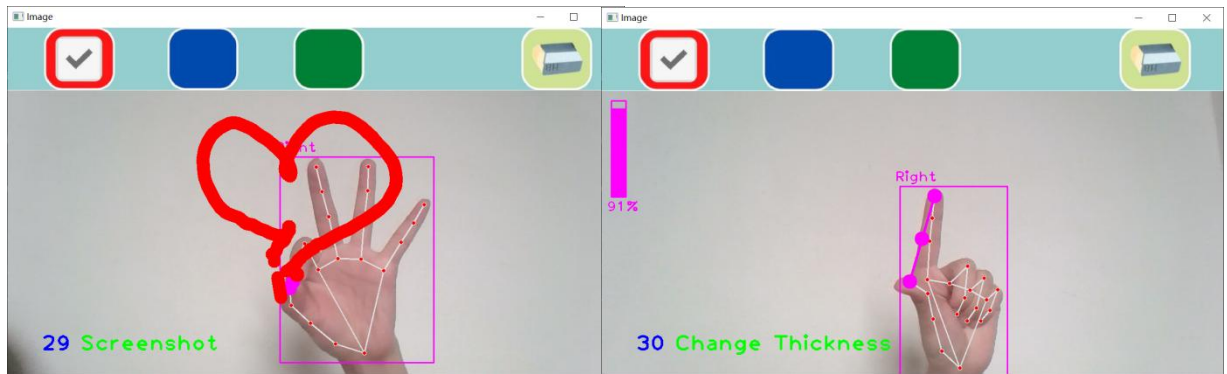


图 3 绘图功能

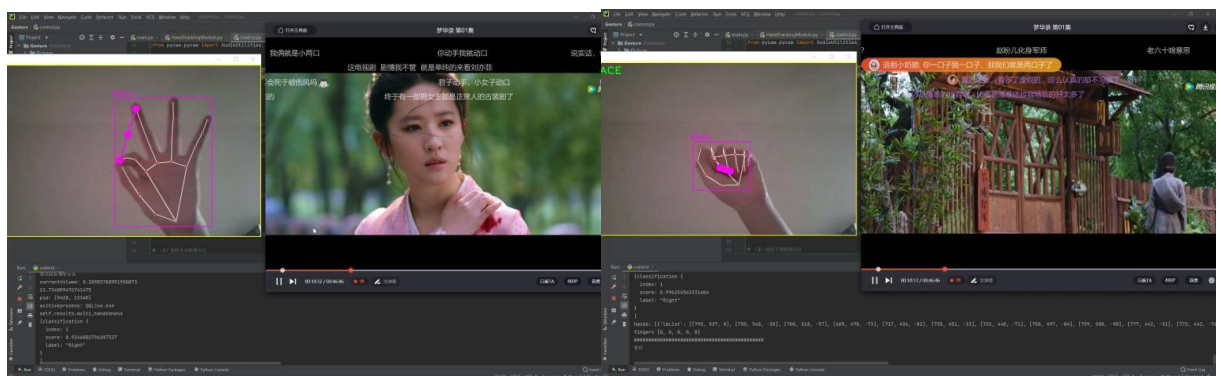


图 4 视频播放功能

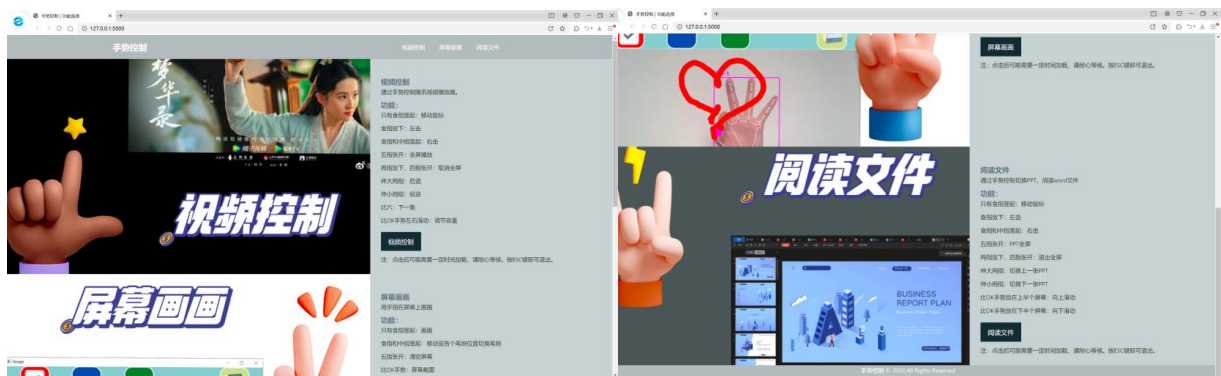


图 5 网页集成所有功能

2.1.4 UI 界面总体架构

UI 界面的总体架构图如图 6:

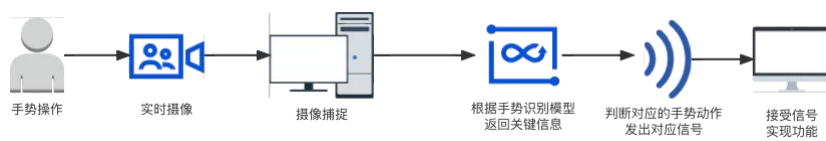


图 6 UI 界面总体架构

2.2 核心模型及其基本原理

2.2.1 手掌检测模型

为了检测初始手部位置，我们设计了一个针对移动实时使用优化的单次检测器模型，其方式类似于 MediaPipe Face Mesh 中的人脸检测模型。检测手是一项非常复杂的任务：我们的 lite 模型和完整模型必须在相对于图像帧具有较大尺度跨度 ($\sim 20x$) 的各种手尺寸上工作，并且能够检测被遮挡和自遮挡的手。尽管面部具有高对比度模式，例如在眼睛和嘴巴区域，但手部缺乏此类特征使得仅从它们的视觉特征可靠地检测它们变得相对困难。相反，提供额外的上下文，如手臂、身体或人的特征，有助于准确的手部定位。

我们的方法使用不同的策略解决了上述挑战。首先，我们训练手掌检测器而不是手检测器，因为估计手掌和拳头等刚性物体的边界框比检测有关节的手指的手要简单得多。此外，由于手掌是较小的物体，因此非极大值抑制算法即使在双手自遮挡情况下也能很好地工作，例如握手。此外，手掌可以使用方形边界框（ML 术语中的锚点）建模，忽略其他纵横比，因此将锚点的数量减少 3-5 倍。其次，编码器-解码器特征提取器用于更大的场景上下文感知，即使对于小物体也是如此（类似于 RetinaNet 方法）。最后，通过上述技术，我们在手掌检测中实现了 95.7% 的平均精度。

2.2.2 手部地标模型

在对整个图像进行手掌检测后，我们的后续手标模型通过回归对检测到的手部区域内的 21 个三维手指关节坐标进行精确的关键点定位，也就是直接坐标预测。该模型学习了一个一致的内部手部姿势表征，甚至对部分可见的手和残疾的手也很稳健。

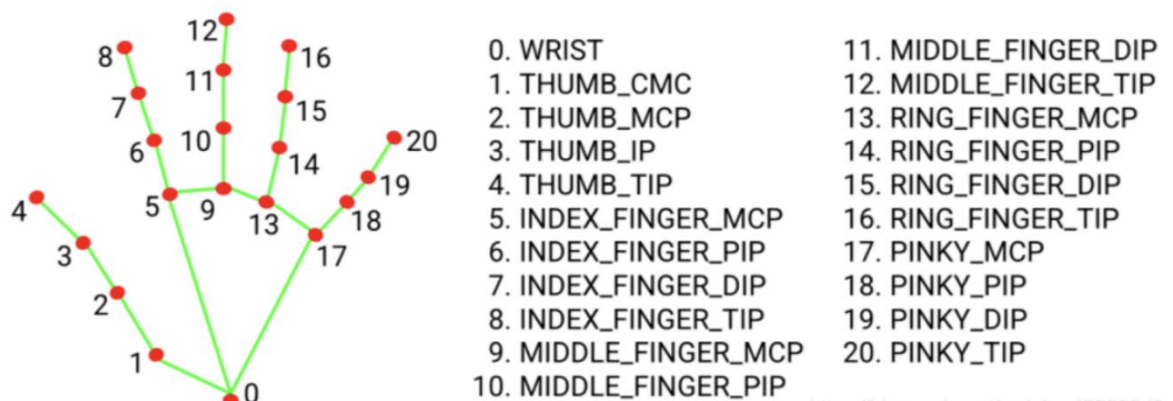


图 7 21 个三维手指关节坐标进行精确的关键点

为了获得真实数据，我们用 21 个三维坐标手动注释了约 3 万张真实世界的图像，如下图所示（我们从图像深度图中获取 Z 值，如果它存在于每个相应的坐标）。

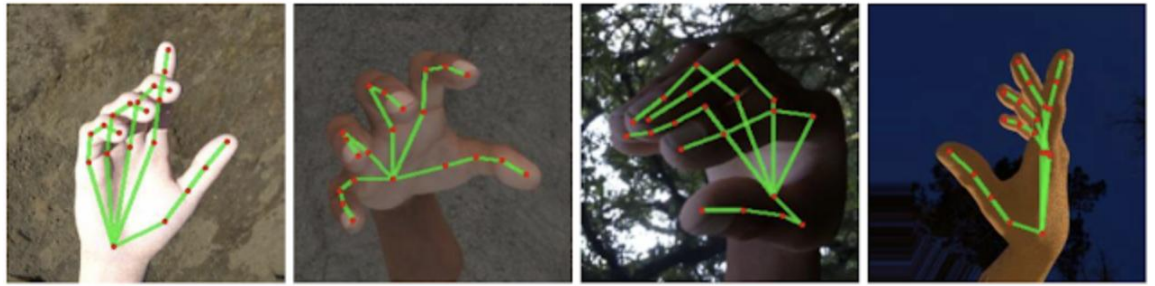


图 8 真实手势图像（包含不同肤色、手势等等）

为了更好地覆盖可能的手部姿势，并提供关于手部几何性质的额外监督，我们还在各种背景上渲染了一个高质量的合成手部模型

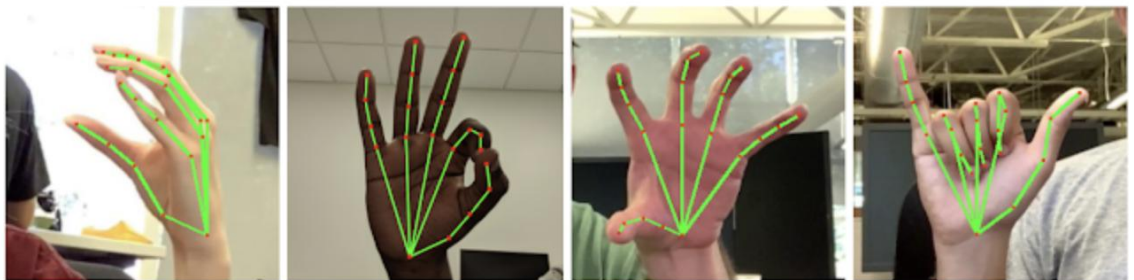


图 9 渲染背景后的手势图像（包含不同肤色、手势等等）

并将其映射到相应的三维坐标上，映射图如下所示。从而建立起了基于 3D 模型的手势建模，更适合计算机实时处理

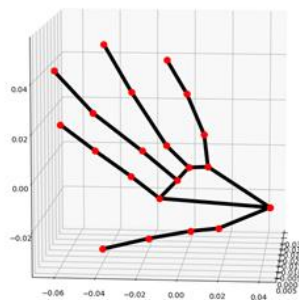


图 10 2D 到 3D 映射示意图

2.3.1 参数说明

【模式设定】

mode 图像模式

如果设置为 `false`，则解决方案将输入图像视为视频流。它将尝试在第一个输入图像中检测手，并在成功检测后进一步定位手标志。在随后的图像中，一旦检测到所有 `max_num_hands` 手并定位了相应的手部标志，它就会简单地跟踪这些标志，而不调用另一个检测，直到它失去对任何一只手的跟踪。这减少了延迟，是处理视频帧的理想选择。如果设置为 `true`，则手部检测会在每个输入图像上运行，非常适合处理一批可能不相关的静态图像。默认为 `false`。

【识别目标-手的个数】

`max_num_hands` 要检测的最大手数。默认为 1。`MULTI_HANDEDNESS` 检测到/跟踪的手的惯用手（即是左手还是右手）的集合。每只手由 `label` 和组成 `score`。`label` 是一串值，要么"Left" 要么"Right"。`score` 是预测的惯用手的估计概率并且总是大于或等于 0.5（并且相反的惯用手的估计概率为 $1 - \text{score}$ ）。

【MIN_DETECTION_CONFIDENCE】

手部检测模型中的最小置信度值被认为是成功的检测，范围为[0.0, 1.0]，默认为 0.5

【MIN_TRACKING_CONFIDENCE】

来自标志跟踪模型的最小置信度值（将其设置为更高的值可以提高解决方案的稳健性，但代价是更高的延迟）。其中手部检测仅在每个图像上运行，如果 `static_image_mode` 为 true 则忽略。范围为[0.0, 1.0]，默认为 0.5。

2.3.2 功能模块设计说明

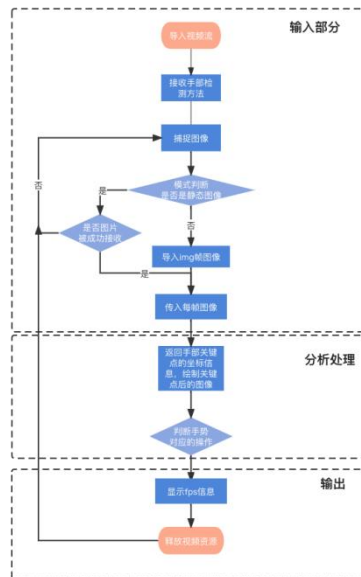


图 10 功能模块设计流程图

【手势判断】

通过计算欧式距离（三维空间两点之间的距离）判断手指有没有弯曲（计算四个手指是否弯曲看手指的指尖距离是否大于关节距离）来判断手势

【鼠标位置监测】

该模块的主要功能是获取用户的操作，主要是在鼠标单击事件发生时，获取鼠标在窗口中的点击位置，并判断此时用户所选择的操作，并将对应操作传入内核部分，调用对应的程序。

【信息显示】

该模块主要是根据用户的输入（通过对摄像流中手势信息的识别）调动对应内核功能程序运行，在显示实时摄像头界面显示信息（左右手、手势信息等）

【渲染功能】

该模块的主要功能是实现程序对用户的反馈，主要负责网页的初始化、选择画笔粗细的反应、场景的重渲染等。

2.4 其他创新内容或优化算法

2.4.1 手势的检测与分割

手势检测与分割的准确度直接影响手势识别的种类、效果和速度。通过颜色信息检测手势，利用 MediaPipe 的 ML 管道，由手掌检测模型在完整图像上运行并返回定向的手边界框；由手部界标模型将该模型在由手掌检测器定义的裁剪图像区域上操作并返回高保真 3D 手部关键点。将精确裁剪的手部图像提供给手部界标模型可以极大地减少对数据增强例如旋转、平移和缩放的需求，并且可以使网络将其大部分功能专用于坐标预测精度。

2.4.2 多场景、多功能集成化

在实现手势识别的基础上，思考手势识别的实用型应用场景，主要实现了视频控制、屏幕画画、文件阅读三大功能，使得用户可以通过摄像头捕捉较远距离看视频、画画、读文献、切换 PPT 等人性化需求功能，从而能够合理利用手势识别的功能，为用户提供便利性和满足感。同时将这些功能利用网页前端集成化，提高人机交互感。

2.4.3 合理使用 html+css 搭建网页前端

采取样式与结构分离的方式，做到了 CSS 和 HTML 的完美分离，减少 html 中的代码，如果需要修改页面的样式，则只需要修改 CSS 文件，当多个页面共用同一个相同的样式时，只会在首个页面访问时需要下载 CSS 文件，随后其他页面访问则只需要下载页面本身，CSS 文件可直接从缓存中取得，不用再次从服务器端下载，从而提高了页面加载的速度，减少页面加载时间，提高性能效率，同时提高代码简洁性，网页布局结构更加清晰。并且合理使用 CSS 选择器，如标签选择器、类选择器、ID 选择器等实现多种样式以及特有属性的设置。

3 实验过程

3.1 环境说明

内核部分开发时的环境说明如表 2 所示：

表 2 内核部分开发环境说明

操作系统	Windows10
开发语言	python
开发环境及具体版本	Python3+opencv+MediaPipe
核心使用库	<pywin32>、<autopy>等 具体在 requirements.txt 中给出

UI 部分开发时的环境说明如表 3 所示：

表 3 UI 部分开发环境说明

操作系统	Windows10
开发语言	html+css
开发环境及具体版本	VSCode

为保证正常编译，我们将环境配置所需要的库放在 requirements.txt 中，执行下列指令配置环境

```
gestureIdentify % pip install -r requirements.txt
```

3.2 源代码文件与主要函数清单

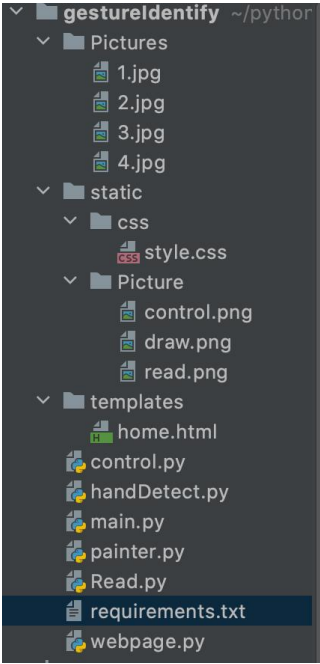


图 11 各文件清单

3.2.1 手势识别部分文件

(1) handDetect.py 存储手势识别模型相关的函数实现

3.2.2 功能实现

(2) read.py 存储文件操作相关的函数实现

(3) control.py 存储视频操作相关的函数实现

(4) painter.py 存储屏幕绘制相关的函数实现

3.2.3 网页部分文件

(5) style.css 存储网页格式定义

(6) home.html 存储网页相关实现

(7) webpage.py 存储内核到交互界面衔接的函数实现

3.2.4 主要函数清单

此处仅给出主要函数清单，具体实现见.py 文件。

(1) main.py:

```
def __init__(self, mode=False, maxHands=2, detectionCon=0.5, minTrackCon=0.5):
```

```
    """
```

```
    捕捉图像
```

```
:param mode: 在静态模式会对没一张图片进行检测：比较慢
```

```
:param maxHands: 检测到手的最大个数
```

```
:param detectionCon: 最小检测阈值
```

```
:param minTrackCon: 最小追踪阈值
```

```
    """
```

```
def findHands(self, img, draw=True, flipType=True):
```

```
    """
```

```
    在图中寻找手掌范围
```

```
:param img: 需要检测的图片
```

```
:param draw: 是否输出结果
```

```
:return: 检测图片结果
```

```
    """
```

```
def fingersUp(self, myHand):
```

```
    """
```

```
    识别多少手指的张合，返回坐标表，（多手识别的时候区分左右手）
```

```
:return: 返回手指状态列表，例如[0,1,0,0,0]代表只有食指竖起
```

```
    """
```

(2) read.py

def read(flag) #实现文件操作相关的函数实现

(3) control.py

def videoControl(flag) #实现视频操作

(4) painter.py

def painter(flag) #实现屏幕绘制

3.3 实验功能

3.3.1 视频控制

视频控制部分可以通过手势动作实现如下功能：进度条回退、进度条快进、全屏、取消全屏、调整音量、跳转到下一集、移动鼠标、暂停和继续播放。详细的功能执行操作如下：

①进度条回退。当伸出大拇指时，执行键盘的左方向键功能，使进度条回退。

②进度条快进。当伸出小拇指时，执行键盘的右方向键功能，使进度条快进。

③全屏。当五指全都竖起时，执行键盘的 Alt+F 键功能，实现全屏播放。

④取消全屏。当四指竖起时，执行键盘的 Esc 键功能，退出全屏播放。

⑤调整音量。当大拇指食指相触，其余三指竖起呈 ok 手势时，通过手的左右移动进行视频音量的调整（向左移动音量减小，向右移动音量增大）。

⑥跳转到下一集。当大拇指和小拇指竖起，呈比六手势时，执行键盘的 Alt+右方向键功能，跳转到下一集。

⑦移动鼠标。当只有食指竖起时，通过跟踪食指位置的实现手指控制鼠标移动。

⑧暂停和继续播放。当五指握拳时，执行键盘的空格键功能，实现视频的暂停播放和继续播放功能。

各操作的手势示例图如下：



图 12 视频控制界面手势操作示意图

3.3.3 手势绘图

绘图部分可以通过手势动作实现如下功能：切换笔刷、调整粗细、画图、擦除、清屏、截图。详细的功能执行操作如下：

- ①调整笔刷粗细。当中指弯下食指在上时，进行调整笔刷粗细操作。
- ②切换笔刷。当食指和中指竖起时，进行切换笔刷操作。
- ③擦除。当切换笔刷为橡皮擦选项时，竖起食指，类似绘图操作进行擦除操作。
- ④绘图。当只有食指竖起时，进行绘图操作，通过指尖位置控制绘图位置。
- ⑤截屏。当食指弯曲其余手指竖起时，进行截图操作。
- ⑥清屏。当五指张开时，进行清屏操作。

各操作的手势示例图如下：

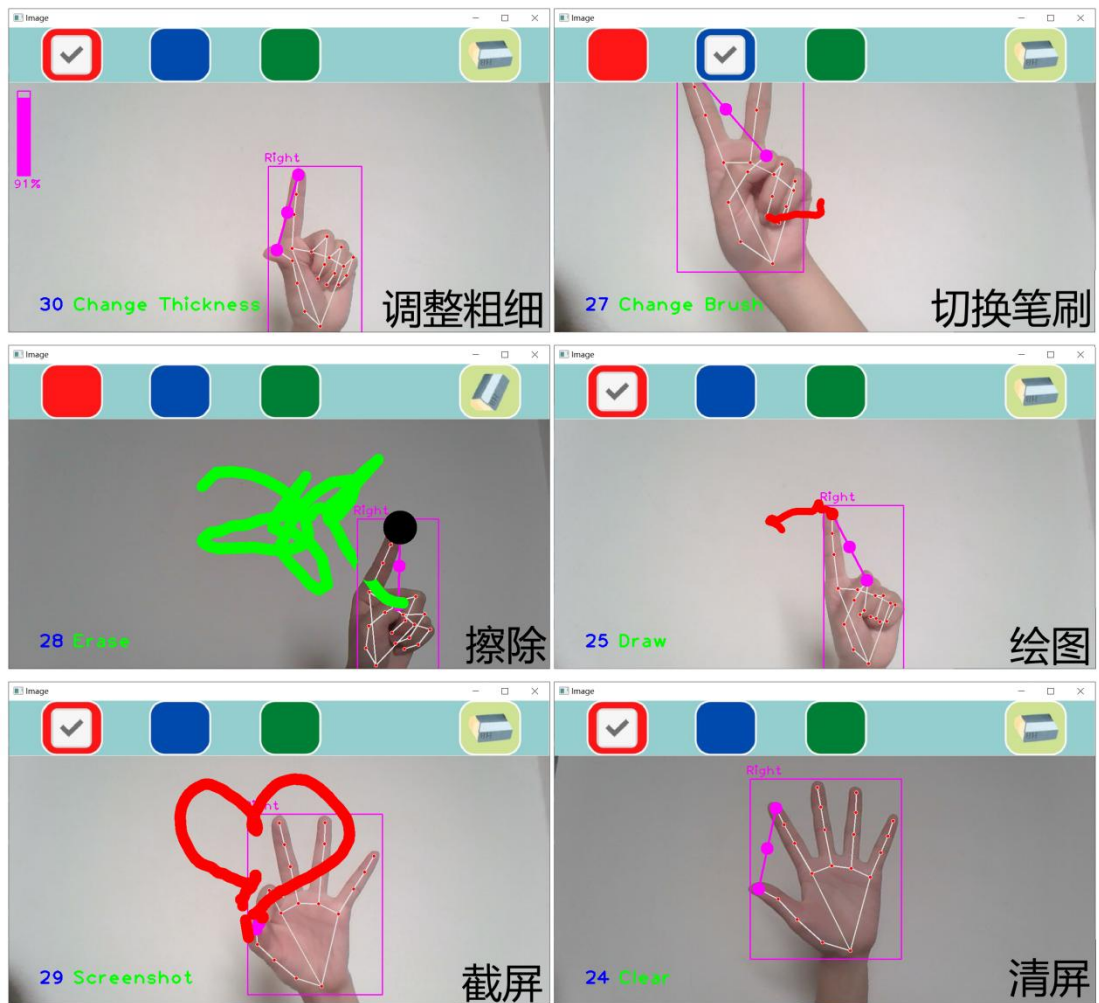


图 13 绘图界面手势操作示意图

3.3.2 ppt 和 word 控制

ppt 和 word 控制部分可以通过手势动作实现如下功能：全屏、取消全屏、跳转到上一页、跳转到下一页、滑动条向上滚动、滑动条向下滚动、右击鼠标、左击鼠标。详细的功能执行操作如下：

- ①取消全屏。当四指竖起时，执行键盘的 Esc 键功能，取消全屏。
 - ②全屏。当五指全都竖起时，执行键盘的 shift+F5 键功能，实现全屏。
 - ③跳转到上一页。当伸出大拇指时，执行键盘的上方向键功能，到达上一页。
 - ④跳转到下一页。当伸出小拇指时，执行键盘的下方向键功能，到达下一页。
 - ⑤滑动条向上滚动。当大拇指食指相触，其余三指竖起呈 ok 手势时，手向上移动。
 - ⑥滑动条向下滚动。当大拇指食指相触，其余三指竖起呈 ok 手势时，手向下移动。
 - ⑦右击鼠标。当食指和中指竖起呈比耶手势时，执行鼠标的单击右键功能，实现右击鼠标。
 - ⑧左击鼠标。当五指握拳时，执行鼠标的单击左键功能，实现左击鼠标。
- 各操作的手势示例图如下：

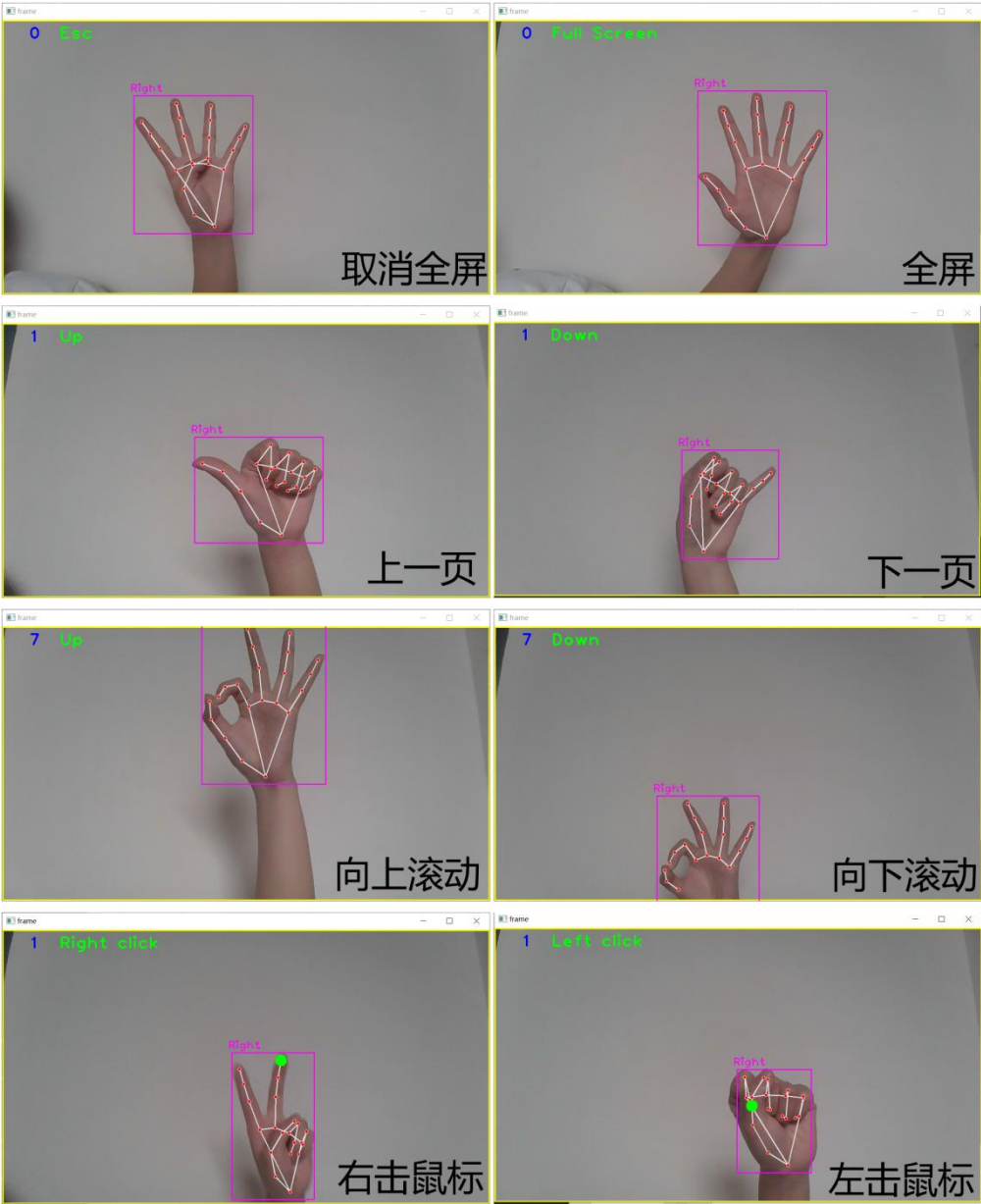


图 14 文件操作手势操作示意图

4 总结

4.1 实验总结

通过本次实验，我们对手势识别人机交互系统有了更深入的了解。通过广泛的查阅资料，我们深入研究了手势识别模型的建立与应用，基于 MediaPipe 实现了对视频的控制、屏幕绘画、文件阅读操作，使得人机交互模式更加便捷，实现了更直接更自然、和谐的人机接口。

为了有较好的效果呈现和用户体验，我们组员特意学习使用 html+css 完成对网页的搭建，使得功能页面简洁美观，操作简洁，有一定的交互性，功能完备。将功能集成在一个网页之中，便于用户选择也更加美观，网页的学习与应用使得程序具有更大的潜在实际应用意义价值。

4.2 实验中问题及解决方案

4.2.1 手指控制鼠标移动卡顿

在利用手势识别控制鼠标移动的过程中发现鼠标在移动过程中会一直晃动，导致操作体验感非常差。

后来自定义平滑系数 smooth，将当前鼠标所在位置坐标(cLocx,cLocy)与之前记录的鼠标位置坐标(pLocx,pLocy)建立关系式 $cLocx = pLocx + (x3 - pLocx) / smooth$ ， $cLocy = pLocy + (y3 - pLocy) / smooth$ ，其中(x3,y3)为当前食指在电脑屏幕范围上的位置，这样就会使得鼠标移动有较为平滑的移动效果。

4.2.2 手势检测速度过快导致动作重复发生

在利用手势识别控制功能的过程中发现由于视频帧率较高，在比一个手势后会重复读入，导致程序认为用户反复给予某一指令，导致动作会重复发生，功能紊乱。

在部分功能读入一次后延时一定时间，避免重复读入相同的指令，反复操作导致混乱。例如鼠标点击的动作，在左击后延时 0.5 秒，防止重复点击。

4.2.3 类似手势之间用户无意识切换带来的错误

在利用手势识别在屏幕上画画的过程中发现切换笔刷粗细和画画动作手势类似，切换笔刷粗细是通过计算拇指和食指的距离调节笔刷粗细，画画动作是食指竖起进行画画。用户在画画过程中可能会有无意识的拇指伸出的动作，导致程序判定为切换笔刷粗细，影响画画过程。

设定变量 frame 记录某一动作的累计帧数，通过记录用户目前的手势和之前的手势，若一致则 frame+1，那么当 frame 大于某一个设定的常量时，认定为用户给予对应的指令，因此根据不同动作、不同操作的特点设置该常量值，例如认定切换笔刷粗细的动作的累计帧数大于 10，才可认为是要切换笔刷粗细，这样可以有效规避可能发生的无意识操作。

4.3 后续改进方向

(1) 本次网页的设置由于时间限制并没有搭建在服务器上，后续考虑在学习 web 网页设计的同时完成对网页搭建的进一步学习，省去用户对环境的配置操作，使得操作更加便捷高效，从而使得功能更加的用户友好。

(2) 静态图片的识别速度还可加速, 考虑往后采集数据集, 使用数据增强代码, 用大数据集去优化算法, 采用精确率、召回率和 F1 综合评价指标来评估从而不断提高手势识别的算法性能

4.4 心得体会

通过此次实验我们学习手势识别模型、html+css 网页搭建等实现了基于手势识别的简单交互系统网页, 对这些方面的知识点有了深一步的认识。同时我们注意到要根据实际情况对代码进行优化, 不断的设计小功能丰富用户体验, 同时要根据问题本身特点对不同情况代码进行不断的调整, 保证程序的正确性与效率。

在 UI 界面, 我们学习使用 html+css 进行网页设计, 编程能力有进一步提升, 能够设计出美观有一定交互的用户友好的程序界面。

综合来说, 本小组成功完成了本次项目, 且在完成项目的过程中不断学习, 丰富功能、充实自我。系统完成了基于手势识别的功能实现, 并进行一定的改进拓展。正如鼠标并没有取代键盘一样, 手势识别系统并不是为了取代键盘, 相反手势识别将增强现有的人机交互模式。并且促成一些新的应用, 例如手语识别等研究多模态交互技术已逐渐成为研究人员的共识。英语的语音识别与唇语理解结合起来已成为不少人研究的课题, 使用口语与手势接口进行分子结构设计的研究也是虚拟现实应用的一个较为成功的例子。可以看到无论是小方向的手势识别还是大方向的计算机视觉现在已经有了许多成熟的结果, 并在解决人工智能的有关问题中得到了广泛的运用。在往后我们也会更加努力深入地去拜读借鉴先进学者的理论与实践, 进行更加深入的研究加深对专业知识的理解并提高自己的专业能力。

5 参考文献

- [1] 《人工智能：一种现代的方法（第 3 版）》，Stuart J.Russell, Peter Norvig 著
- [2] 《手部 21 个关键点检测+手势识别-[MediaPipe]》
https://blog.csdn.net/weixin_45930948/article/details/115444916
- [3] 《MediaPipe 中文官方文档》
- [4] 《OpenCV 中文官方文档》
- [4] 《Autopy 中文官方文档》
- [5] 《PyAutoGUI 中文官方文档》

6 成员分工与自评

分工：

- 郑欣雨：主要负责手势识别部分代码编写，对界面设计一部分进行调整，PPT 制作；
- 刘欣：主要负责手势控制功能部分代码编写，并对项目各部分进行整合、调试，演示视频制作；
- 徐奕：主要负责网页界面整体设计以及实现，对部分算法进行优化并撰写报告。

自评：

• 郑欣雨：本次实验我主要负责手势识别部分代码编写，通过自学查找资料利用 MediaPipe 完成手势识别功能，为后面实现手势控制功能奠定基础。通过动手对算法的实现，对机器学习、机器视觉有了更深的理解。在程序设计和实现过程中也让我对如何与其他组员沟通、对实验进行更合理的模块划分和任务分配有了新的认识。同时参与界面的一部分调整也让我学习到了有关网页的相关设计知识。

• 刘欣：本次实验我主要负责实现手势控制功能，在手势识别实现的基础上，思考实用性应用场景，实现不同的功能，在过程中也增进了对算法的理解总体上能积极进行学习认真完成了自己的工作，设计功能完整的模块，并在实现过程中对于出现的问题及时进行解决。同时对项目各部分进行整合、调试，让我在算法的原理知识掌握基础上关注到了许多在实现过程中的细节，提高了动手实践的能力，积极与其他两位同学沟通，根据反馈及时调整实现方法，锻炼了多人合作模式下如何设计封装性好，功能齐全的程序模块的能力。

• 徐奕：本次实验我主要负责网页界面设计，通过自学查找资料能熟练掌握利用 html+css 搭建网页前端界面的方法，努力设计出美观有一定交互的用户友好的程序界面。同时对部分算法进行了一定的优化，在过程中也增进了对算法的理解。在整个实验进程中，能认真设计了自己负责的部分，同时也对实验的各个模块进行了合理的整合，调试过程中也始终保持耐心，关注每一个细节，根据运行情况对各个部分进行适应性的调整，并及时给其他两位组员反馈，共同修改，解决问题。