



# 虚拟画板实验报告

## 一. 项目介绍

### 1) 项目背景

传统人机交互技术很大程度依赖于鼠标、键盘等输入设备，无法满足更多的智能交互需求。MediaPipe是一款由Google开发并开源的数据流处理机器学习应用开发框架，结合传统画板，可以实现空中绘画，绘图，手写字符识别，打造出一款拥有更加智能的人机交互，更加精准、灵敏的手势识别和文字识别的多功能空中虚拟画板。

### 2) 项目内容

本项目由五大模块组成，彩笔，形状，粗细，文本和橡皮擦，五个模块分别对应五个菜单，显示在页面的顶部。

1. 彩笔：用户点击彩笔，可以选择彩笔的颜色，目前可供用户选择的颜色有蓝色，红色，绿色和黄色。
2. 形状：用户点击形状，可以进行绘图，目前可供用户选择的图形有圆形，矩形，椭圆形和任意多边形。
3. 粗细：用户点击粗细，可以调整画笔的粗细，目前画笔粗细有四种选择，分别为5px, 15px, 25px, 35px。
4. 文本：用户点击文本，可以开启文本识别的模式，此时用户可以在屏幕任意位置中手写数字、英文、中文，书写完成之后屏幕会显示出识别效果。

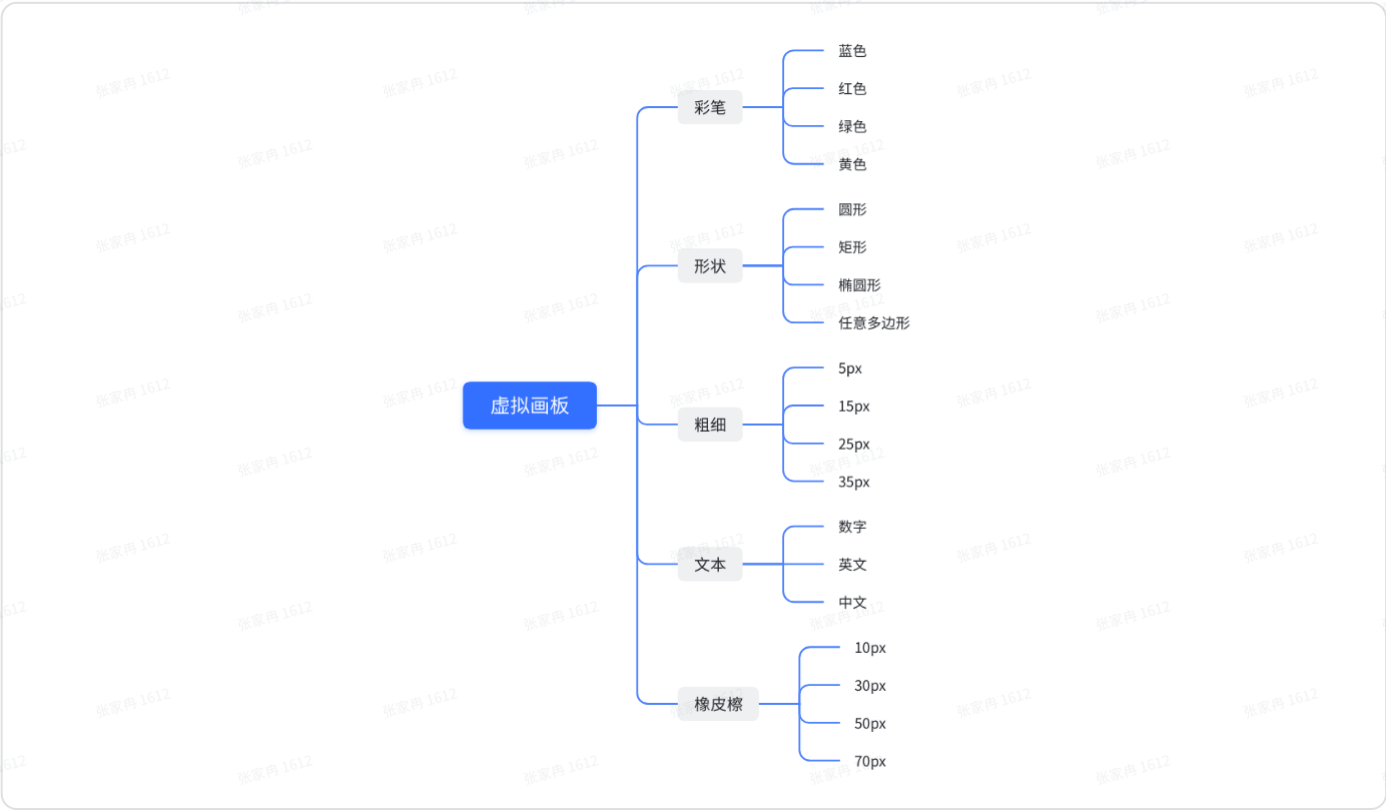
5. 橡皮擦：用户点击橡皮橡皮擦，可以对画板内容进行擦除。

### 3) 项目亮点

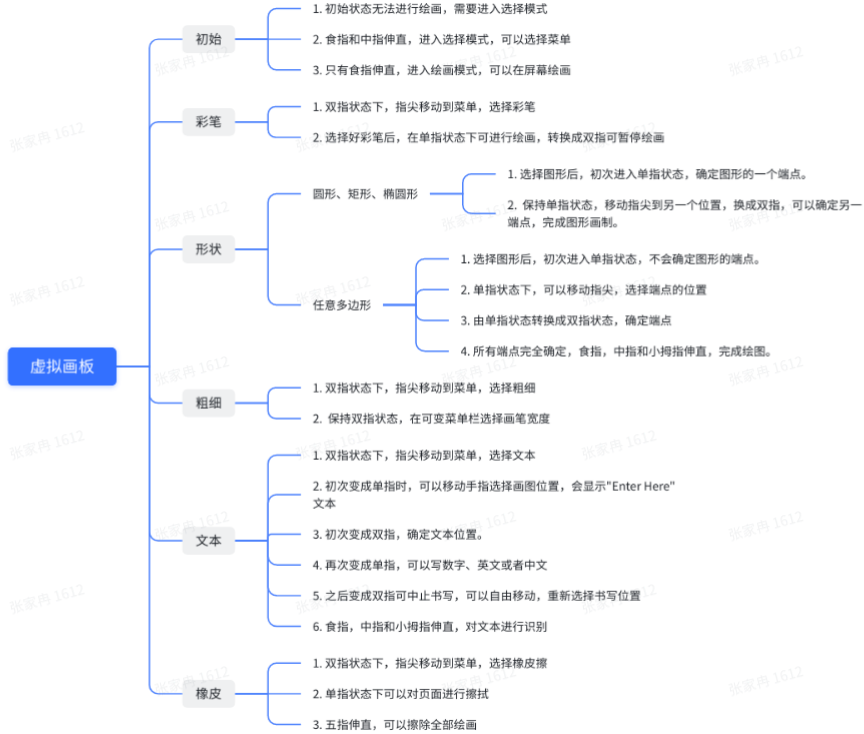
- 1. 结合MediaPipe技术，在空中仅凭手掌移动和不同手势就能完成多种图形的绘画，突破了传统画板必须依赖键鼠等输入设备的局限。
- 2. 不同于传统的黑底或白底画板，本项目可以打开摄像头，画板背景为当前的摄像机画面，用户的参与感更强。
- 3. 实现空中手写文字的文本识别，对手写板中写入的中英文以及数字进行识别。
- 4. 在虚拟画板中插入的图形不局限于圆形、矩形、椭圆形，还可以根据不同的手势插入任意多边形。

## 二. 功能

### 1) 项目功能模块



### 2) 项目手势模块



## 四. 实现思路

### 1) 前端网页

前端需求：显示项目简介和用户说明书、提供打开程序的接口

鉴于Flask在Python程序中搭建Web框架比较容易且高效，我们决定使用Flask对Python程序与前端设计进行交互。

#### 1. 滑动页面设计

出于对文本篇幅和网页美观的考虑，显示项目简介和用户说明书的功能最好通过不同的交互方式分开。

对于这种需求有两种主流的实现方式有静态网页跳转和使用动态网页。考虑到频繁的网页跳转带给用户的服务体验较差，我们最终决定使用动态网页（以点击按钮对界面进行滑动切换的方式对网页进行动态交互），通过JavaScript让网页在不跳转的前提下实现用户交互和功能需求。

具体做法：在页面中添加两个按钮，一个是当网页处于显示项目简介部分时切换至使用说明书，另一个则恰好相反。随后为两个按钮添加触发器，实现滑动功能。

#### 2. 使用说明书的分块展开

由于使用说明书太长，如果要保证全部展开，字体势必过小，不利于网页的美观和用户的阅读。我们决定将使用说明书分块展示（分为“初始”、“彩笔”、“形状”、“粗细”、“文本”、“橡皮擦”六个模块）。使用说明书界面只展示这六个模块名，需要点击相应的模块名才能阅读相应说明书的介绍。

一开始我们的思路是在现有的粉色界面内加入使用说明书的所有内容，再根据发生响应的按钮令相应的部分可视化，显示对应的内容。但是考虑到系统的自动排版等因素，如果采用这种方法进行实现。为了保证网页排版正常，逻辑上会非常混乱，极易出错。所以我们选择了另一种方法。

在保留现有粉色界面的前提下，创建几个和当前粉色界面同等大小、位置重合的粉色界面，在每个粉色界面中写入相应的展示内容，并设置其为不可视。当点击相应的按钮时，将原有的粉色界面修改为不可视，显示界面修改为可视；从显示界面返回原有界面时则恰好相反。

### 2) 后端画板

虚拟画板的核心就是识别用户的手势，并将绘画的结果展示在摄像机捕捉到的画面。Mediapipe自带的方法已能识别手掌并获取手掌上的关键点信息，有了关键点信息就可以基于其实现许多功能，如使用关键点序号为8的食指指尖作为绘画指针，而“向上指”这一手势的判断就是比较一根手指的指尖和下面的所有关节的y坐标。除了一些普遍的基本用法外，结合要实现的功能，主要问题有两个：绘画的持久显示与擦除的实现，以及单项工具下的手势切换。

## 1. 多画布叠加

和普通的 Mediapipe 应用方式相同，通过 OpenCV 的 VideoCapture 和 read 方法来获取摄像机画面，并通过 imshow 和 while 循环实现摄像头实时画面的输出。而绘画的实现较为简单，声明两对坐标变量存储当前画面帧和上一个画面帧中指针（在本项目中即为食指指尖）的坐标，使用函数画线即可。而由于摄像头画面在不断更新，为了效率，程序不可能存储之前所有需画线的点，因此需考虑如何将用户所画的持久地显示在画面中。

这也是本项目实现过程中较为巧妙的地方，采用多画布叠加的方式来呈现最终效果。该方法的核心出发点就是每一张图片，本质上都是一个三维 numpy 矩阵，而 OpenCV 存在方法 bitwise\_and 和 bitwise\_or 可进行两矩阵的与、或操作，这样就可以进行图像的像素级操作，类似于 PS 的多图层叠加效果。

摄像机获取的画面存为 img，为最终要展示的画面，另建一个与 img 相同尺寸的三维 numpy 矩阵 imgCanvas 作为绘画画布，考虑到图像叠加方法，将该画布设计为全黑背景，因为黑色的 RGB 均为 0，在后续的图像或操作中不会产生印象。而用户的绘画实际是绘画在该画布上，完成当前帧的操作后，将该画布叠加到 img 即可。

若直接使用图像或操作叠加画面后，会产生同一颜色的画笔在不同区域会有不同颜色的效果的问题。这是因为摄像机画面 img 本身每个点的 RGB 值都不一样，直接或运算后的结果也不一样。而黑色的 RGB 均为 0，因此，考虑将 imgCanvas 反转黑白二值化，将其变成白底黑线的画布，再将其与 img 做画面与操作，则画笔所画的地方均变为黑色，而其他地方由于白色的 RGB 均为 255，因此不产生影响。叠加后的结果再与 imgCanvas 做或操作即可。

擦除的实现较为简单，由于 imgCanvas 背景为黑色，那么擦除时直接使用黑色覆盖原像素点即可。

手写文字识别输入功能也采用类似方法，创建 imgWriting 作为手写的画布，再创建 imgResult 作为识别结果的画布，而每次识别都会清空 imgWriting，将识别结果加入 imgResult，最后将 imgResult 与 img 叠加。

## 2. 状态变量

由于在单项工具的使用过程中会出现单种或两种手势无法实现所有功能的情况，如任意多边形的绘制和文本输入，加入状态变量能较好地实现多手势对应的功能并优化代码。对于绘画工具，创建变量代表当前工具；对于可变工具栏，创建变量代表该部分应显示的内容，当指针移动到特定选项区域时，修改变量的值来使得可变工具栏显示对应的内容。

状态变量的判定和初始化也是设计中较为重要的一点，对状态变量的判断和对应的值的修改是多手势功能的核心，如任意多边形的绘制，绘画的过程中需要单指来移动指针，双指来确定多边形中一个点的位置，三指来闭合多边形并结束绘画。也就需要一个变量、三种值来表示未选择初始状态、当前点在移动、当前点已确定三种状态，来完成多边形的绘制（具体绘制流程见项目网页的“使用说明书”部分）。且所有的判断语句均使用 if...elif... 格式，来减少每一帧的工作量，使画面较为流畅。

## 3) 文字识别

在虚拟画板的文本功能中，我们实现了将手写文字（中英文）和手写数字的识别。在此，我们调用了PaddleOCR开源的API，将图片识别部分写入了util.py文件的getText函数中，把上文创建的imgWriting手写画布作为参数传入getText中，函数将返回识别出的文本结果。

## 五. 环境

### 1) 开发&运行环境

IDE：PyCharm 2022.1.3

### 2) 包

- python 3.8.10
- mediapipe 0.9.0.1
- opencv-contrib-python 4.6.0.66
- numpy 1.23.1
- requests 2.28.1
- flask 2.2.2

## 六. 交互

### 1) 网页部分

前置条件	用户操作	响应事件
滑动界面处于“项目简介”状态	点击按钮“查看使用说明书”	粉色界面滑动切换至“使用说明书”状态
滑动界面处于“项目说明书”状态	点击按钮“查看项目简介”	粉色界面滑动切换回“项目简介”状态
滑动界面处于“项目说明书”状态	点击按钮“初始”	粉色界面显示“初始”功能的说明
滑动界面处于“项目说明书”状态	点击按钮“彩笔”	粉色界面显示“彩笔”功能的说明
滑动界面处于“项目说明书”状态	点击按钮“形状”	粉色界面显示“形状”功能的说明
滑动界面处于“项目说明书”状态	点击按钮“粗细”	粉色界面显示“粗细”功能的说明
滑动界面处于“项目说明书”状态	点击按钮“文本”	粉色界面显示“文本”功能的说明
滑动界面处于“项目说明书”状态	点击按钮“橡皮擦”	粉色界面显示“橡皮擦”功能的说明
滑动界面处于功能说明界面	点击按钮“返回”	显示界面返回“项目说明书”界面



无条件要求	点击按钮 "START"	执行程序，进入虚拟画板
进入虚拟化版	按esc	程序退出，跳转至结束界面

2) 画板界面

前置条件	用户操作	响应事件
无	双指状态，且食指指尖进入“彩笔”工具区域	可进行彩笔绘画，可变工具栏显示四种彩笔、形状、文本的颜色
已选中“彩笔”工具	双指状态，且食指指尖进入可变工具栏的“蓝色” / “红色” / “绿色” / “黄色” 区域	彩笔、形状、文本的颜色变为蓝色/红色/绿色/黄色
无	双指状态，且食指指尖进入“形状”工具区域	可进行形状绘制，可变工具栏显示四种形状
已选中“形状”工具	双指状态，且食指指尖进入可变工具栏的“圆形” / “矩形” / “椭圆形” / “任意多边形” 区域	绘制的形状变为圆形/矩形/椭圆形/任意多边形
无	双指状态，且食指指尖进入“粗细”工具区域	可变工具栏显示四种线条粗细选项，可选择线条粗细
已选中“粗细”工具	双指状态，且食指指尖进入可变工具栏的四种粗细中的一种所在的区域	彩笔、形状、文本的线条粗细变为对应粗细
无	双指状态，且食指指尖进入“文本”工具区域	可进行手写文本输入，可变工具栏显示四种字体
已选中“文本”工具	双指状态，且食指指尖进入可变工具栏的四种字体大小中的一种所在的区域	字体大小变为1/2/3/4
无	双指状态，且食指指尖进入“橡皮擦”工具区域	可变工具栏显示四种线条粗细选项，可选择橡皮擦粗细
已选中“橡皮擦”工具	双指状态，且食指指尖进入可变工具栏的四种粗细中的一种所在的区域	橡皮擦的粗细变为对应的10/30/50/70像素
已选中某种工具	单指状态	食指指尖处有圆形跟随
已选中某种工具	双指状态	食指指尖与中指指尖之间有矩形跟随
无	手掌完全在画面内且五指向上指	清除整个画面

七. 测试

1) 网页部分

用例编号	用例名称	用例说明	基本事件流	异常事件流
001	粉色界面的滑动检测	检测动态网页的滑动功能是否能正常实现	1.在界面处于“项目简介”状态下滑动至“使用说明书”  2.在界面处于“使用说明书”状态下滑动至“项目简介”	1.界面不能正常滑动 2.界面正常滑动，但是
002	使用说明书中“初始”部分的显示	在“使用说明书”状态点击粉色界面中的“初始”按钮，粉色界面会显示相应的说明界面	1.点击按钮“初始”，粉色界面显示“初始”功能的说明  2.点击按钮“返回”	1.粉色界面不能正确显示 2.显示界面不能正常返回
003	使用说明书中“彩笔”部分的显示	在“使用说明书”状态点击粉色界面中的“彩笔”按钮，粉色界面会显示相应的说明界面	1.点击按钮“彩笔”，粉色界面显示“彩笔”功能的说明  2.点击按钮“返回”	1.粉色界面不能正确显示 2.显示界面不能正常返回
004	使用说明书中“形状”部分的显示	在“使用说明书”状态点击粉色界面中的“形状”按钮，粉色界面会显示相应的说明界面	1.点击按钮“形状”，粉色界面显示“形状”功能的说明  2.点击按钮“返回”	1.粉色界面不能正确显示 2.显示界面不能正常返回
005	使用说明书中“粗细”部分的显示	在“使用说明书”状态点击粉色界面中的“粗细”按钮，粉色界面会显示相应的说明界面	1.点击按钮“粗细”，粉色界面显示“粗细”功能的说明  2.点击按钮“返回”	1.粉色界面不能正确显示 2.显示界面不能正常返回
006	使用说明书中“文本”部分的显示	在“使用说明书”状态点击粉色界面中的“文本”按钮，粉色界面会显示相应的说明界面	1.点击按钮“文本”，粉色界面显示“文本”功能的说明  2.点击按钮“返回”	1.粉色界面不能正确显示 2.显示界面不能正常返回
007	使用说明书中“橡皮擦”部分的显示	在“使用说明书”状态点击粉色界面中的“橡皮擦”按钮，	1.点击按钮“橡皮擦”，粉色界面显示“橡皮擦”功能的说明	1.粉色界面不能正确显示 2.显示界面不能正常返回



		粉色界面会显示相应的说明界面	2.点击按钮“返回”	
008	运行虚拟画板	点击按钮 "START" 运行虚拟画板	1.点击按钮 "START" 2.运行虚拟画板程序	1.按钮无法点击 2.按钮点击无效
009	结束界面	退出虚拟画板，跳转至结束界面	1.运行虚拟画板 2.按esc退出 3.跳转至结束界面	1.无法退出虚拟画板程序 2.未跳转至结束界面

2) 画板界面

用例编号	用例名称	用例说明	基本事件流	异常事件流
010	初始状态下的绘画	检测初始状态下是否能进行绘画	1.无法进行绘画	1.能进行彩笔的绘画, 的输入
011	“彩笔”工具选择	选择后，进行彩笔绘画，可变工具栏部分发生相应变化	1.可变工具栏部分变为四种颜色 2.单指状态下，进行绘画 3.双指状态下，暂停绘画	1.可变工具栏无变化 2.无法绘画 3.无法暂停绘画 4.恢复绘画后，与前面
012	颜色切换	选择颜色后，修改彩笔、形状、文本的颜色	1.彩笔、形状、文本的颜色对应改变 2.从其他工具模式切回后，将上次选择的结果作为颜色	1.彩笔/形状/文本的颜色不对应 2.从其他工具模式切换后未认颜色
013	“形状”工具选择	选择后，进行形状绘制，可变工具栏部分发生相应变化	1.可变工具栏部分变为四种颜色 2.单指状态时，可进行点的移动 3.单指状态变为双指状态时，完成点的确定或形状的绘画	1.可变工具栏无变化 2.单双指状态下，功能
014	形状切换	选择形状后，修改将要绘制的形状	1.完成对应形状的绘画 2.形状绘画完成后，变为初始状态	1.形状不对应 2.未变为初始状态
015	“粗细”工具选择	选择后，进行彩笔、形状、文本的线条粗细选择，可变工具栏部分发生相应变化	1.可变工具栏部分变为四种线条粗细	1.可变工具栏无变化
016	粗细切换			

		选择粗细后，修改彩笔、形状、文本的线条粗细	1.彩笔、形状、文本的线条粗细发生对应变化 2.从其他工具模式切回后，将上次选择的结果作为线条粗细	1.彩笔、形状、文本或改变不对应 2.从其他工具模式切回为默认粗细
017	“文本”工具选择	选择后，进行文本输入，可变工具栏部分发生相应变化	1.可变工具栏部分变为四种字体大小 2.初次变为单指能进行文本效果预览 3.双指时进入文本输入状态，与“彩笔”绘画模式相同 4.三指状态时进行文本识别或退出文本输入状态	1.可变工具栏无变化 2.文本预览效果未消失 3.上一个文本的绘画模式未消失 4.文本识别结果未显示 5.无法进行多次输入
018	字体大小切换	选择字体大小后，修改文本的字体大小	1.文本的字体大小发生对应变化 2.从其他工具模式切回后，将上次选择的结果作为字体大小	1.文本的字体大小未消失 2.从其他工具模式切回为默认大小
019	“橡皮擦”工具选择	选择后，可进行擦除绘画，可变工具栏部分发生相应变化	1.可变工具栏部分变为四种橡皮擦粗细 2.单指状态下，可进行绘画擦除	1.可变工具栏无变化 2.无法进行擦除 3.双指状态下无法暂时隐藏
020	橡皮擦粗细切换	选择橡皮擦粗细后，修改橡皮擦的粗细	1.橡皮擦的粗细发生对应变化 2.从其他工具模式切回后，将上次选择的结果作为橡皮擦粗细	1.橡皮擦的粗细未消失 2.从其他工具模式切回细变为默认粗细
021	全屏清除	清除所有绘画	1.五指向上时，进行全屏清除 2.手掌不完全在画面内时，禁用该功能	1.无法进行清除 2.手掌不完全在画面内时，禁用该功能

## 八. 部署

- 解压缩文件后运行 main.py 文件，点击运行弹出的网址即可进入“虚拟画板 Vitural Painter”网页端。
- 体验方式
  - 在网页中可以查看项目简介以及使用说明书。点击start按钮即可开始体验虚拟画板，按esc按钮关闭画板
  - 关闭后需重新启动程序再次体验虚拟画板。

## 九. 项目现状

## 1) 项目完成情况:

1. 实现了web端。前端实现了项目介绍,项目说明等页面,后端实现了彩笔,形状,粗细,文本和橡皮擦五大功能。
2. 彩笔功能可以选择画笔的颜色;形状功能可以选择绘制圆形,椭圆形,矩形和任意多边形;粗细功能可以调整画笔的大小;文本功能可以对手写字符识别;橡皮擦可以对绘画的内容进行擦除。

## 2) 项目欠缺情况:

1. 手写字符识别功能不够完善,书写不够规范将导致无法正确识别。
2. 手写字符识别只能显示在页面上,没有办法输出到用户电脑中。
3. 项目没有部署到服务器上,只能本地运行和访问。
4. 每个功能可以选择的内容还比较少,比如只能选择四种颜色,四种画笔大小和四种图形。

## 十. 未来工作

1. 将项目部署在服务器上,加入用户注册和登录功能,保存个人信息,对用户识别的数据进行保存。
2. 优化手写字符识别,从算法层面提高识别的精确度,灵敏度;支持多行文本识别,支持传入其它手写图片识别;支持识别内容输出到用户电脑。
3. 支持更多的手势识别的功能,比如加入手语识别。
4. 扩展彩笔的选择,加上其它颜色或者实现一个颜色圆盘,自由选择颜色。
5. 扩展图形的绘制,比如加上三角形,扇形等等。
6. 扩展画笔粗细的选择,可以换成滑动选择,可以适应不同用户的需求。