

<<<<< HEAD

JLU_Label_Software

📢 目前最新版本Format v1.1.0 (Updated)

本项目在Qt6.6.0下,采用MSVC2019编译器编译!

相关的Opencv的库已经集成在源码库中(Format v1.1.0 方可用到, v1.0.0不需要)

```
libs in ./JLU_Label_Software/libs/libopencv*.dll
```

这下是我的第一个Qt项目了, 经过114514秒的艰苦奋斗, 修了1919810个bug 整出来了这个离谱玩意.

发现任何bug,可以火速联系chengh1922@jlu.edu.cn进行维修

Guidance

Quick Key Sequence Previewed

这里是该软件操作的一系列快捷键——

当前支持的快捷键:

主窗口:

1. 打开 (导入) 图片 **Ctrl + O**
2. 打开 (导入) 文件夹 **Ctrl + Shift + D**
3. 切换到下一张: **->** 或者是 **D**
4. 切换到上一张 **<-** 或者是 **A**
5. 切换图片 **Ctrl + Alt + P**
6. 切换文件夹 **Ctrl + Alt + D**
7. 删除当前图片 **Ctrl + D**
8. 管理标签 **Ctrl + L**

标注模式

1. 切换到下一张: **->** 或者是 **D**
 2. 切换到上一张 **<-** 或者是 **A**
- (注意这里就不允许更换图片了)

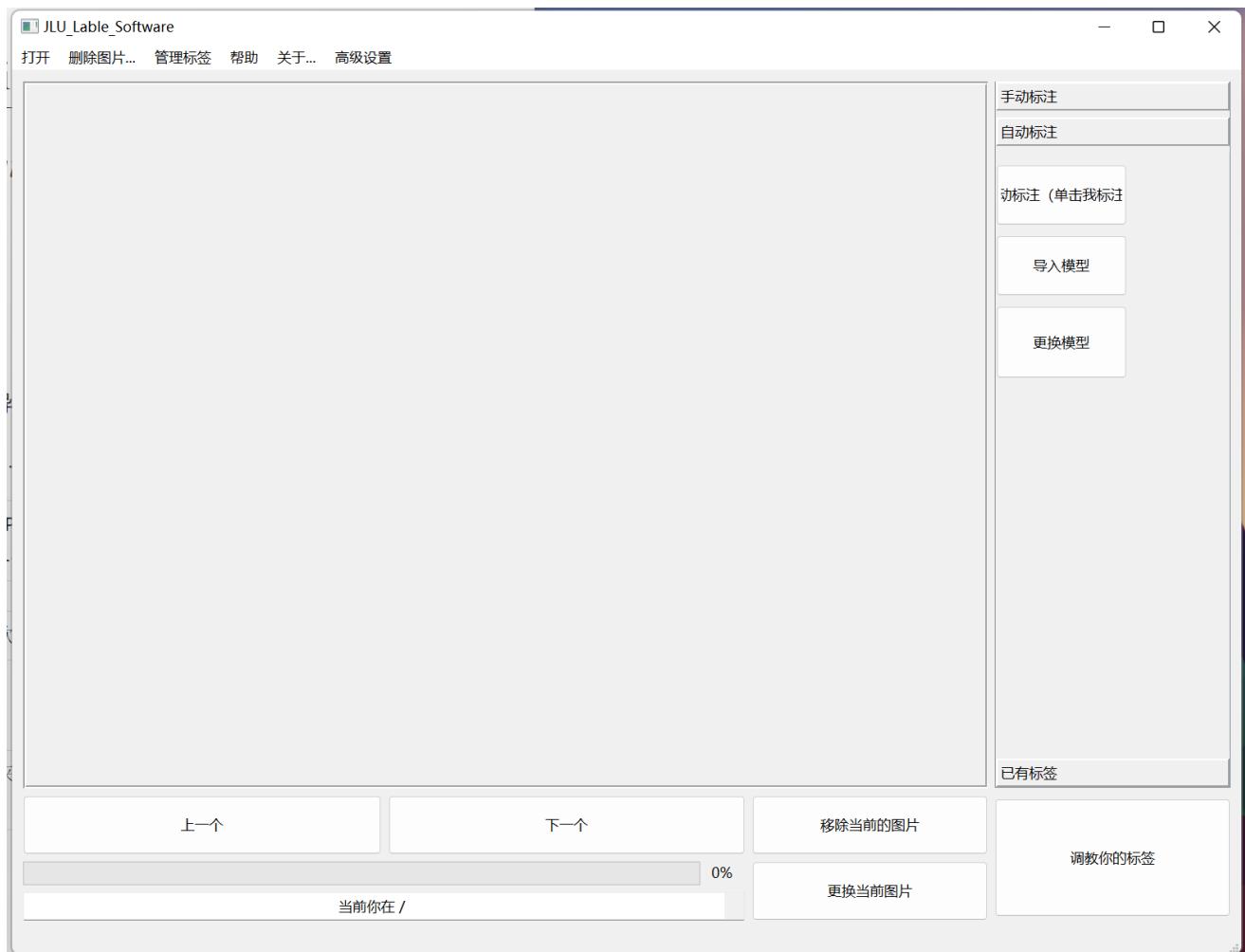
实际标注界面

1. 下一张 (若选择了默认路径就直接按照默认的保存方式保存到自己选择的默认的文件夹)

方向键->

2. 保存图片和标签 **Ctrl + S**
3. 撤销上一次标注 **Ctrl + Z**

start



➡ 导入图片 (文件夹)

此软件正义支持单张图片导入，文件夹导入。自然的，也支持切换图片和文件夹。

导入时，自动过滤无关文件（支持*.jpg;*.png;*.bmp）格式的图片

添加图片的快捷键是：Ctrl + O，如果是文件夹，则是Ctrl + Shift + D

关于图片格式：你当然可以在MainWindow.h找到ALLOWING_PICDIR_TYPE这个宏再添加图片格式，以及不要忘记更新ALLOWING_PICDIR_TYPE，不然QFileDialog不会显示过滤选项

导入图片采用的是缓式评估法则，因此，本软件至少可以一键导入至少2000张图片（乐）

⚠ 注意：

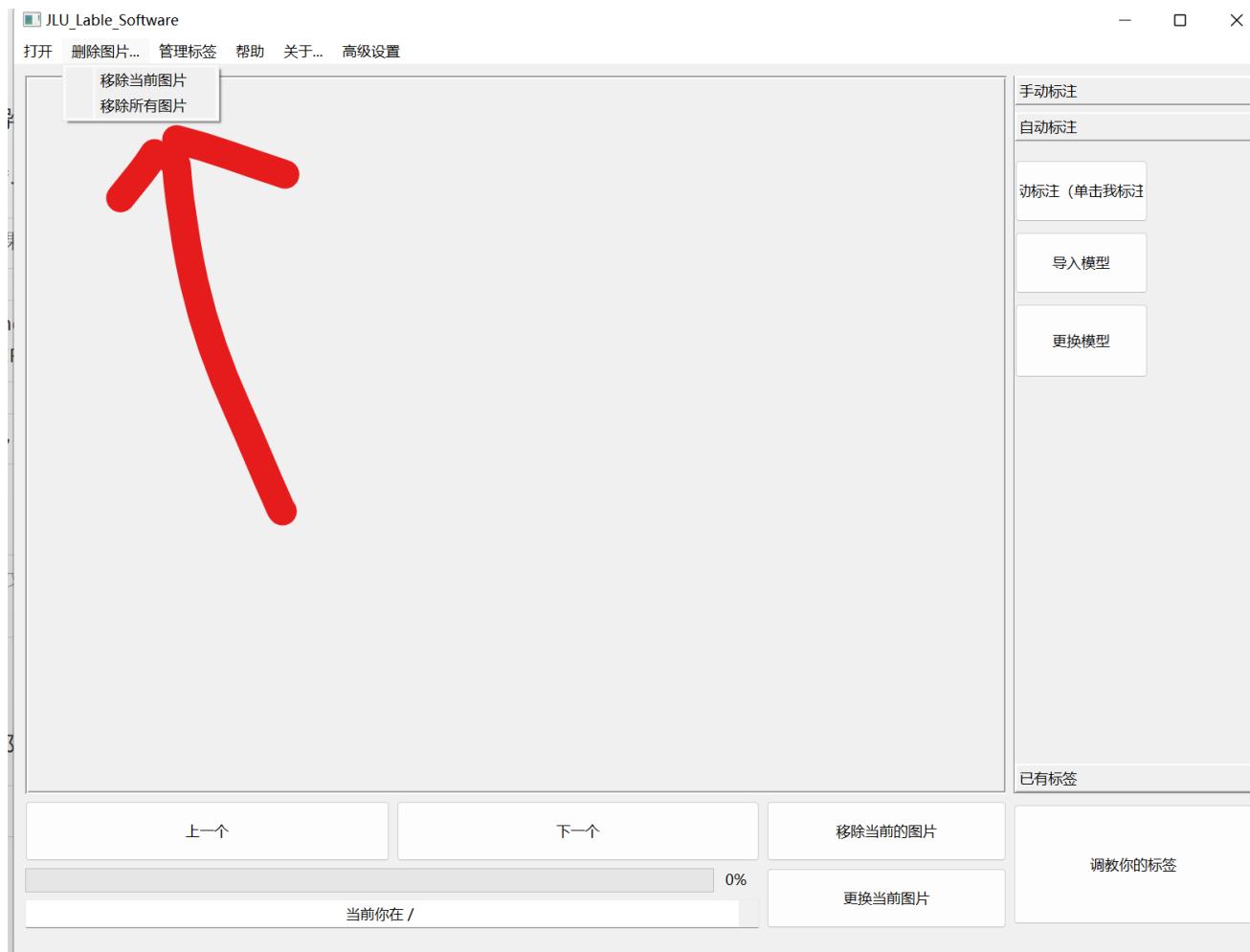
必须强调的是—单击菜单栏的切换图片文件夹后，会先清空当前的图片列表，所以，想清楚了再切换！

➡ 删除图片

如果你觉得这张照片很烂，不想要了，那就可以删掉它：

快捷键：删除当前图片 Ctrl + D

或者，单击菜单栏下的“删除图片”



→ 管理标签

当然，一个图像标注软件没有标签管理，就像西方没有耶路撒冷一样。

你既可以单击菜单栏上的管理标签，也可以单击“调教你的标签”这个按钮。或者：

管理标签 Ctrl + L

你值得拥有。后面阐述具体功能

→ 帮助

你可以查看快捷键，也可以使用之来反馈bug（大喜）

→ giao级设置

可以设置标签分割符，这个软件支持外部写配置文件导入分割符。默认的——我们的分割符是";"(是英文分号！)，你当然可以修改成其他的分割符！

导览图片

你可以勤快的单击上一张和下一张，也可以使用快捷键：

切换到下一张： -> 或者是 D
切换到上一张： <- 或者是 A

当你确认完图片是Acceptable的，可以单击切换到标注模式去做进一步的标注工作了：**所有在列表的图片都会被送到ManuallyLabel，也就是标注模式的Window下，请删选完成图片集后再单击！**

标签管理

标签管理是软件的下一个部分。**在多个窗口中我们都可以实时编辑我们的标签列表！**

以MainWindow下我们打开标签管理为例子：



➡ 添加标签

标签中，我们只支持纯英文输入，结束输入后，请点击确认添加，如果发现自己添加错误，可以单击反悔添加。**原则上，如果你是在非标注时添加的，你点击完确认添加后，可以任意的点击勾选标签后单击选择确认选择这个标签**（这是因为接口共用的原因，考虑到标注时候会设计新标签添加）

➡ 删除标签

勾选你想删除的标签，然后点击删除标签即可。注意，一次只会删除一个！（这是因为接口共用的原因）

所有的标签设置—添加行为，都是全软件同步的！

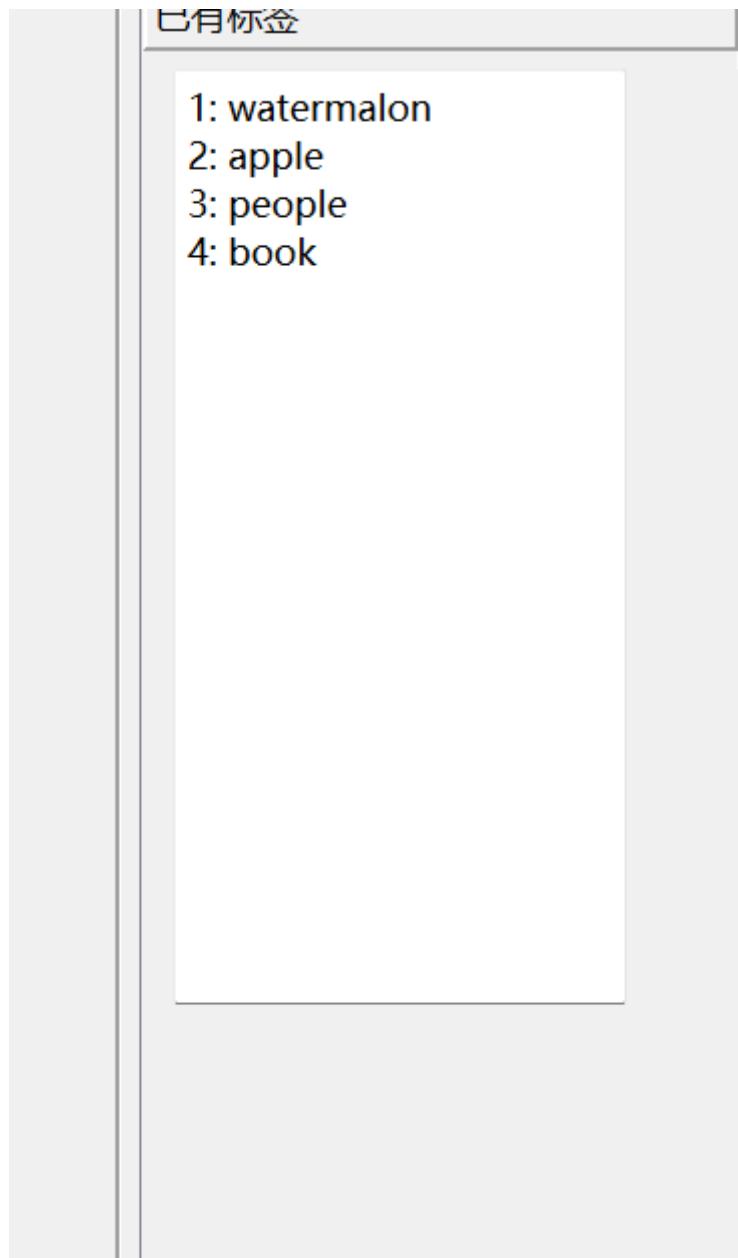
→ 导入标签

你可以单击导入外部标签导入标签，值得注意的是——名称貌似自己输入选择！

注意到可以在主窗口的菜单栏——高级——设置分割符中设置分割符。这样，就可以导入标签了：

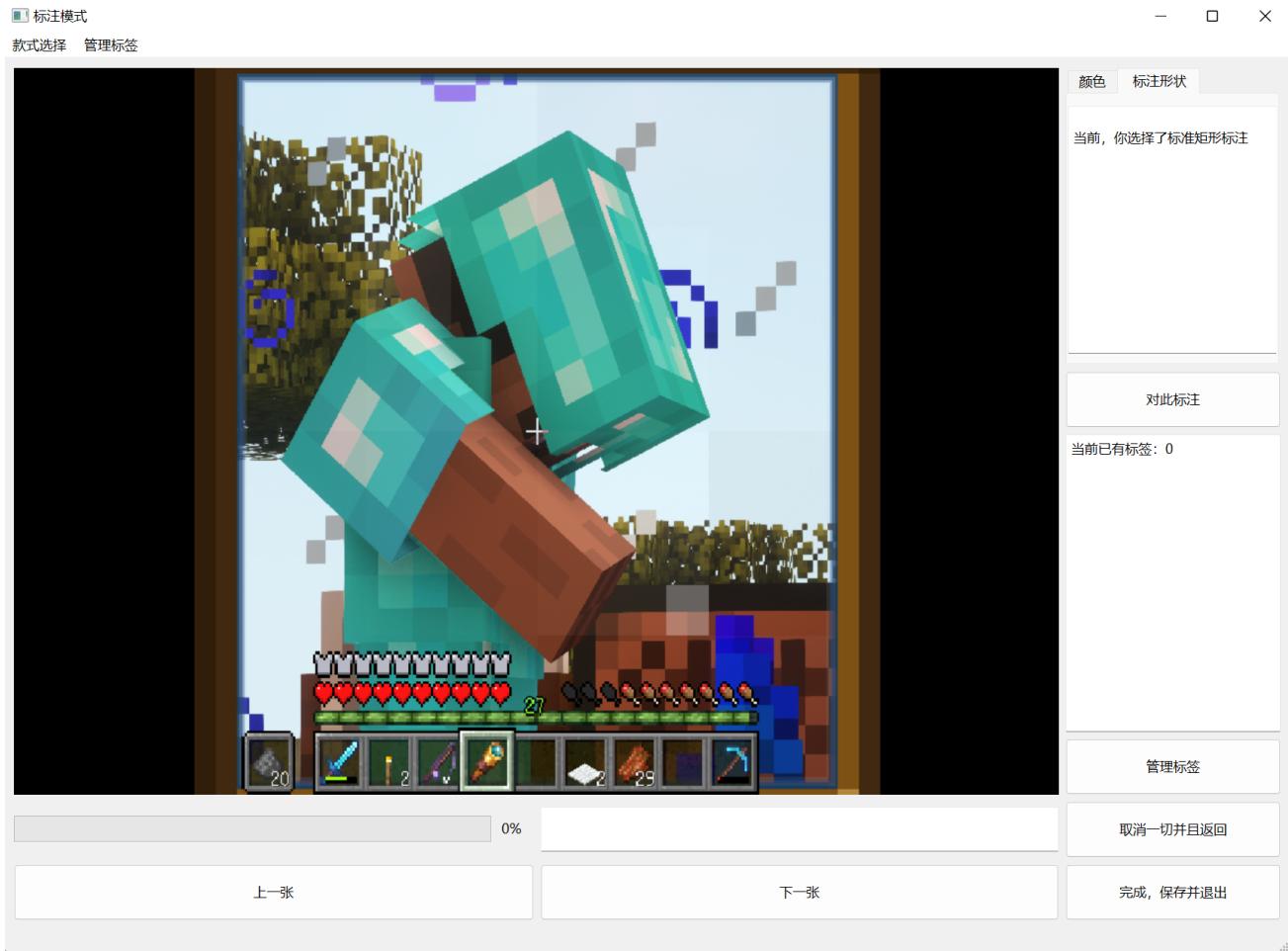
如，默认的分割符是;，于是，这样书写标签

```
watermalon;apple;people;book
```



手动标注模式

单击标注模式，你就可以开始管理你具体的标注方式了



只有导入了图片才允许打开标注模式窗口！否则会抛出错误（不妨试试？）

这里，你可以设置标签，也可以设置等会标注时候的笔宽（标注的笔的宽度），笔色。

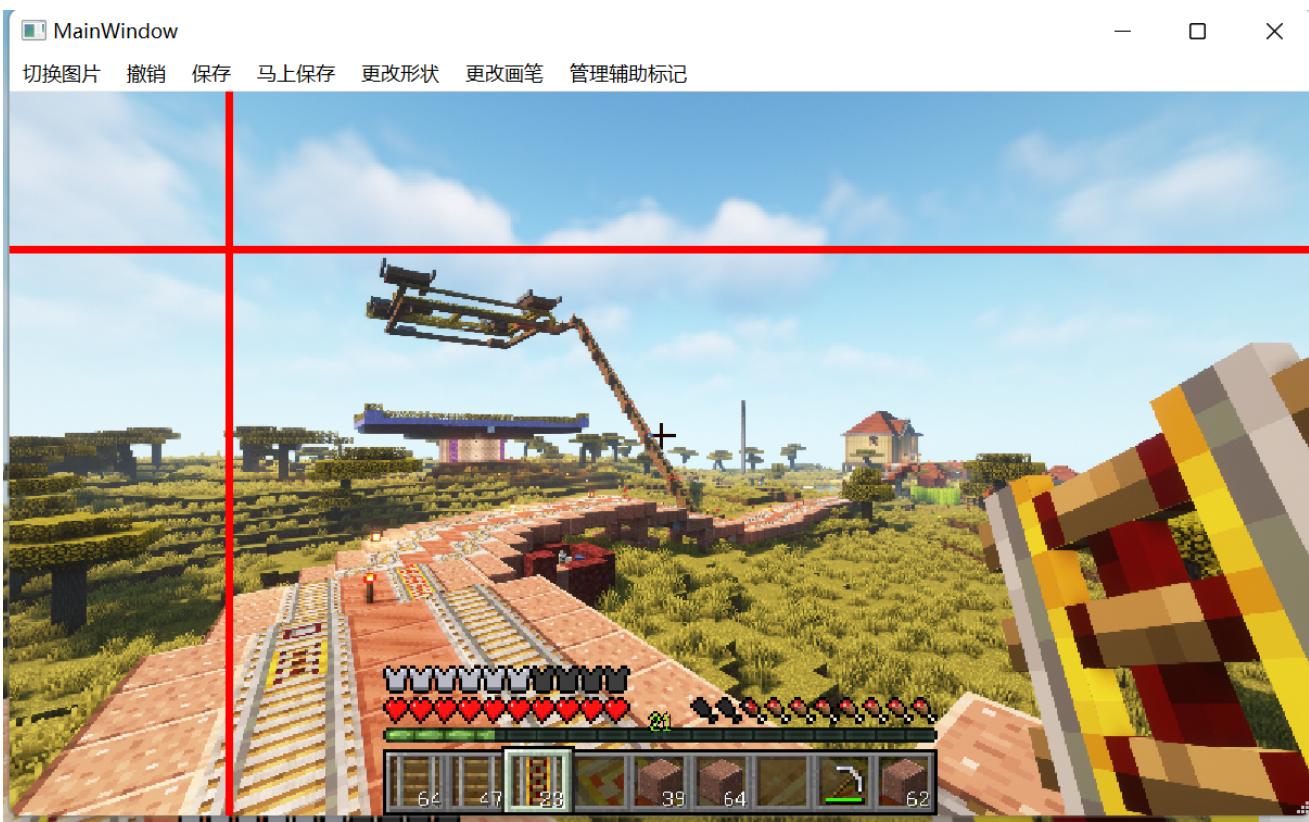
可以单击菜单栏的款式选择专门的选择标注，其余的显而易见，这里不做说明了

正式标注

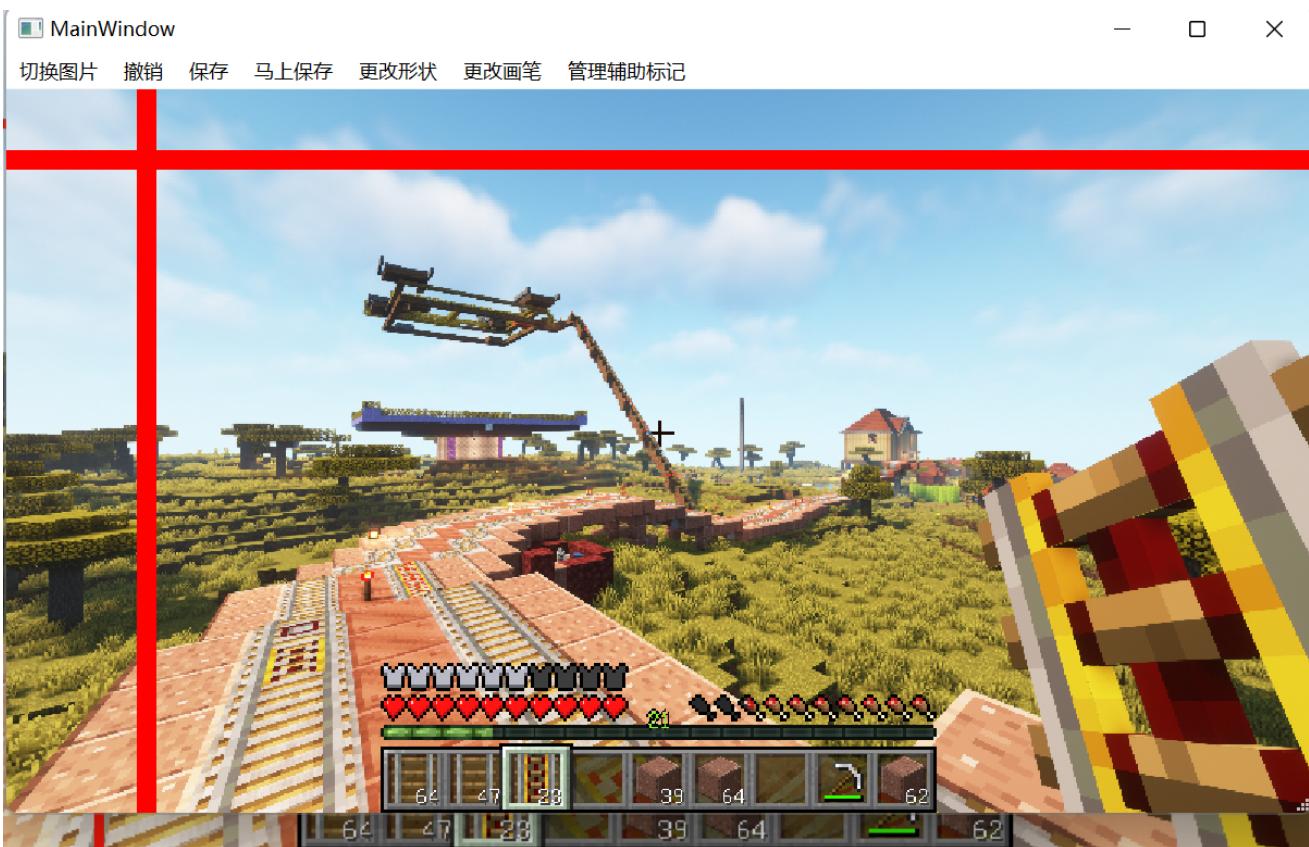
单击标注模式下，对此标注的摁扭开始，我们就会从这个列表的首部开始进行标注。但是在那之前，我们必须获得用户的标注模式。

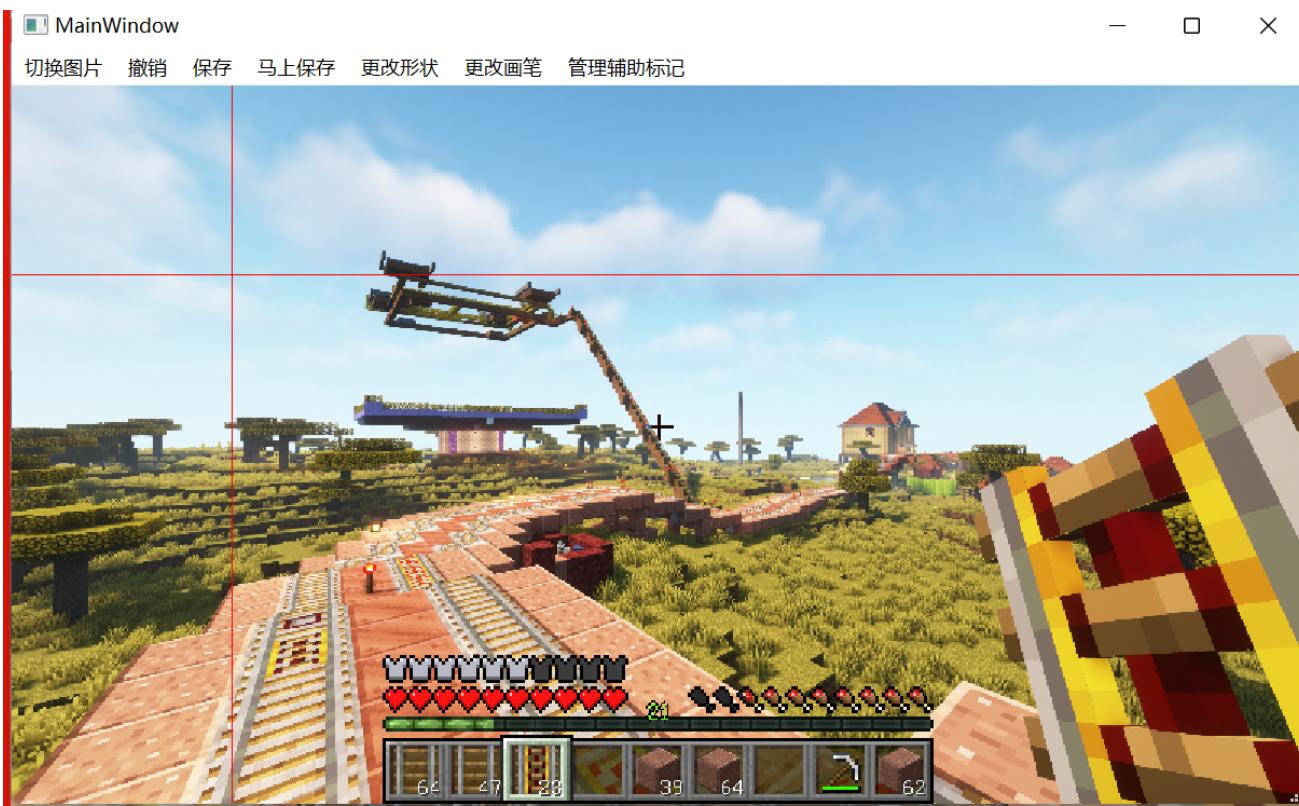
智能起见，此软件会在每一次打开全列表标注的时候询问一次是否保存固定路径的问题。包括图片固定路径和标签固定路径，和本次标注的方式。请确认好后选择！

标注



如你所见，两个交叉的线就是辅助线，帮助你定位你鼠标的位置的。你可以——通过滚轮来调整辅助线的宽度，向上滚是加粗，向下滚是变细





如你所见，只需要单击窗口，就可以标注了！对于上一个点都不满意者，需点击撤销上一步

单击即可标注：



对于单个点的重置，你可以采取点击菜单栏上的撤销上一个点，或者是

快捷键：Ctrl + <-

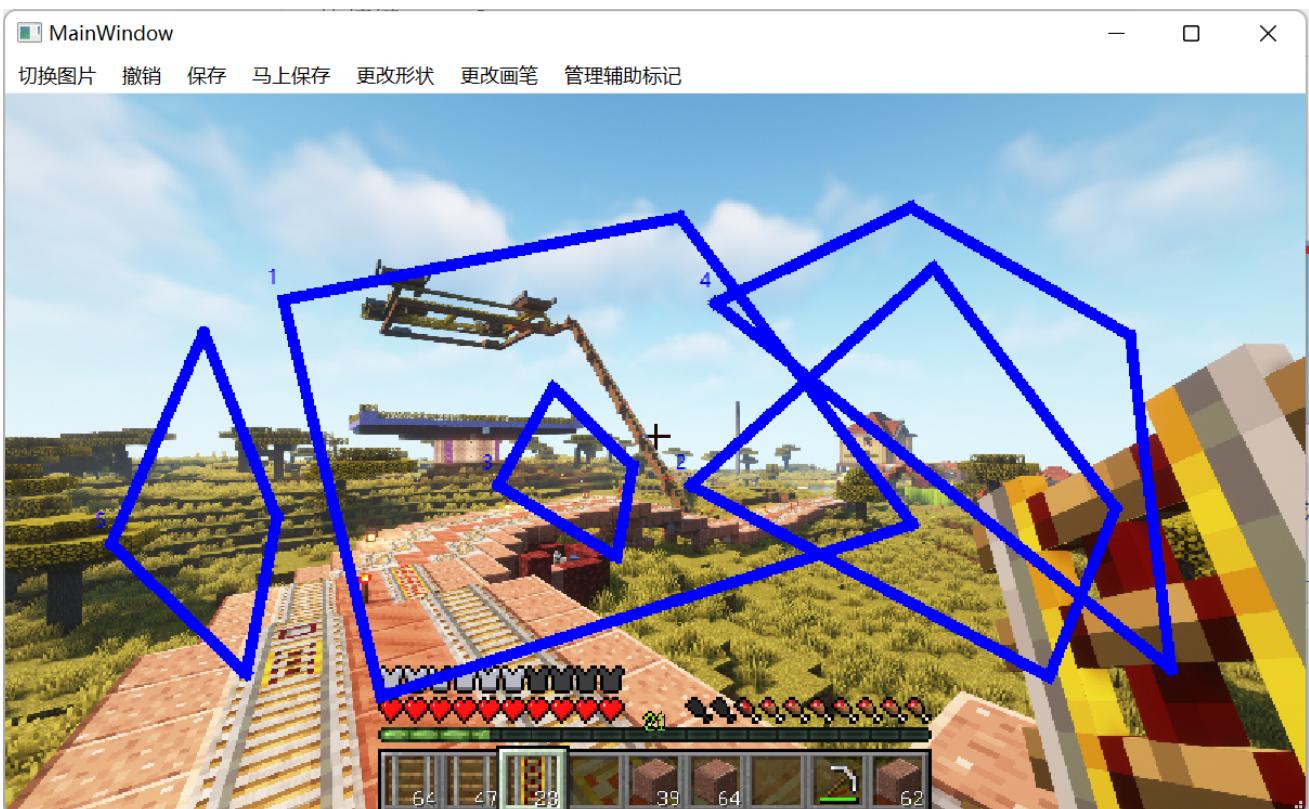
当你选择完标注后，会自动弹出标签窗口——



对于单个点的重置，你可以采取点击菜单栏上的撤

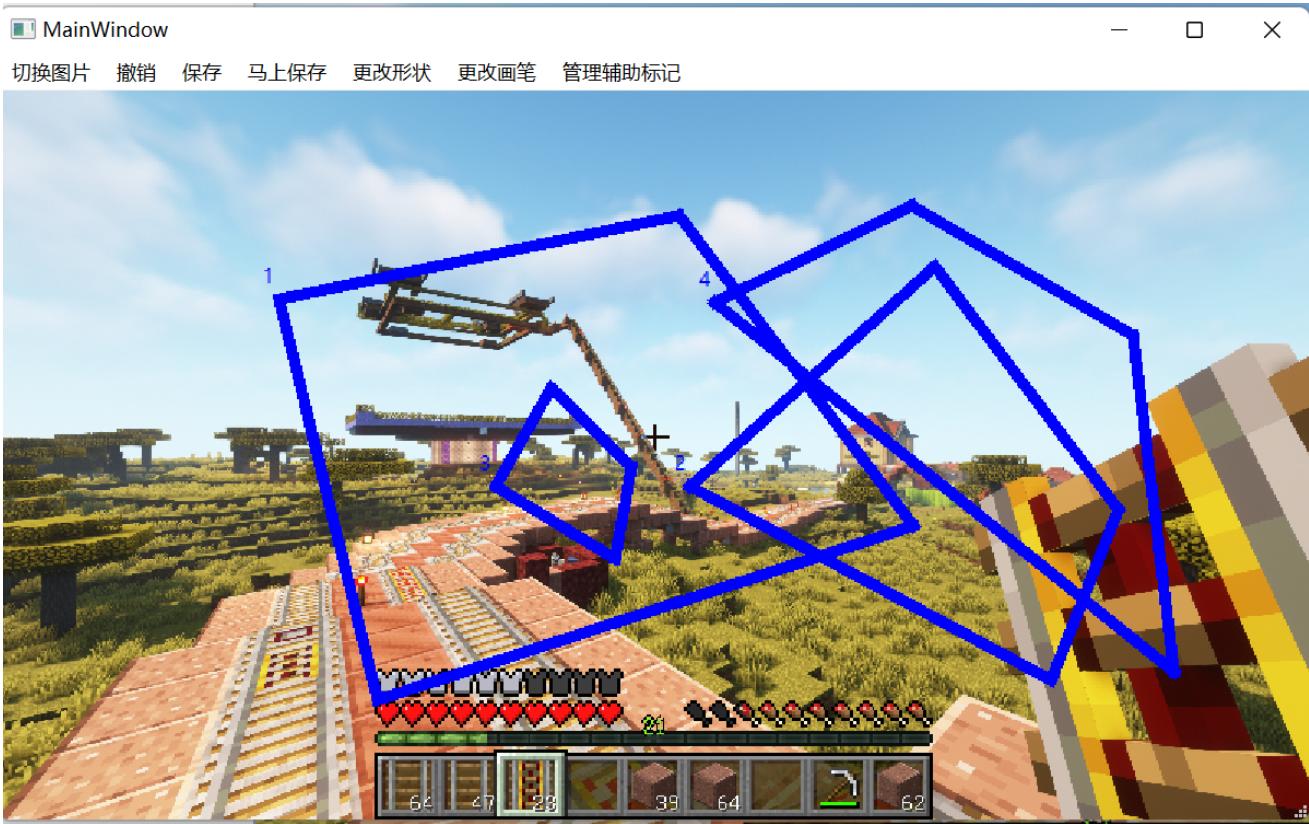
快捷键：Ctrl + <-

然后愉快的编辑确定后，就可以继续标注了：



哦，第五个没有标好（看到那个小小的数据了嘛，提示你这是第几个轮廓的！），你可以单击菜单栏上的——撤销——撤销上个轮廓，也可以使用

快捷键：Ctrl + Z

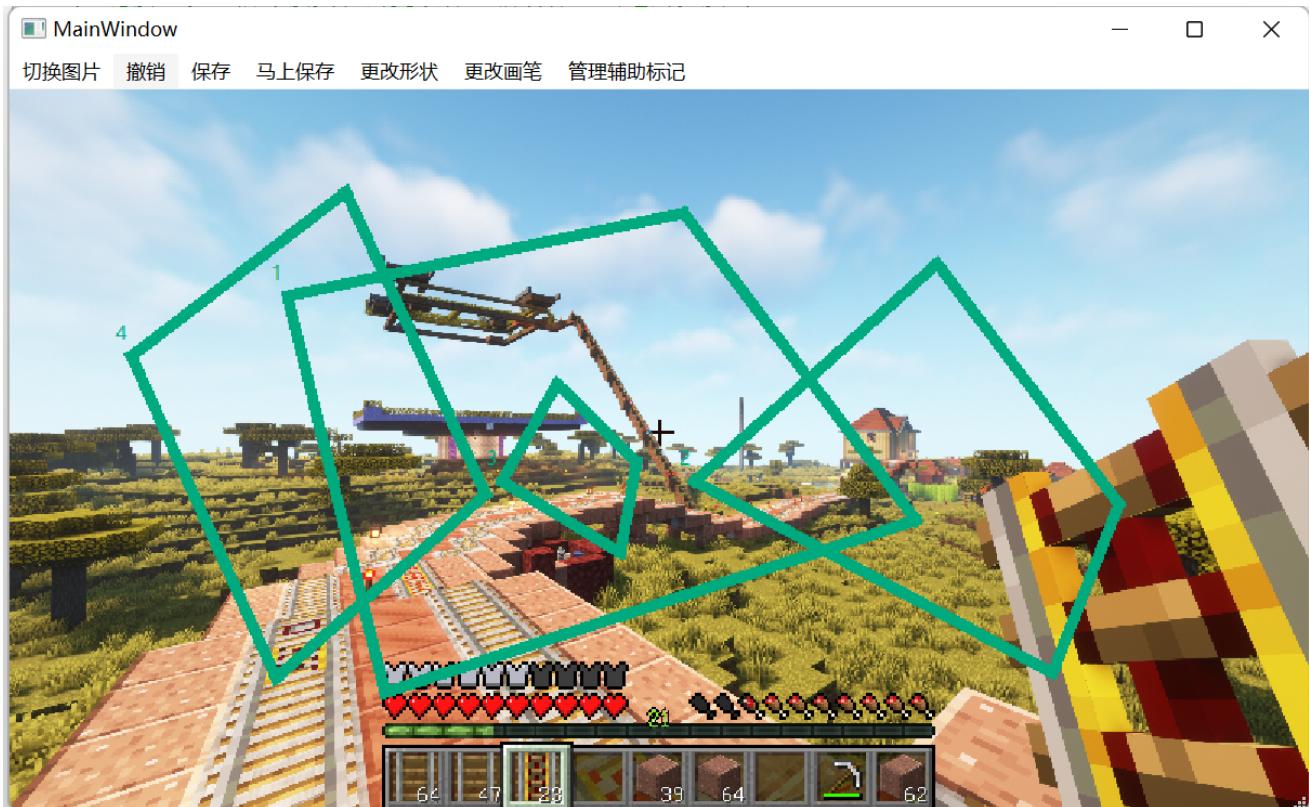


款式选择

如你所见，你可以更改画笔



画笔的颜色和粗细是全局的，一改尽改！

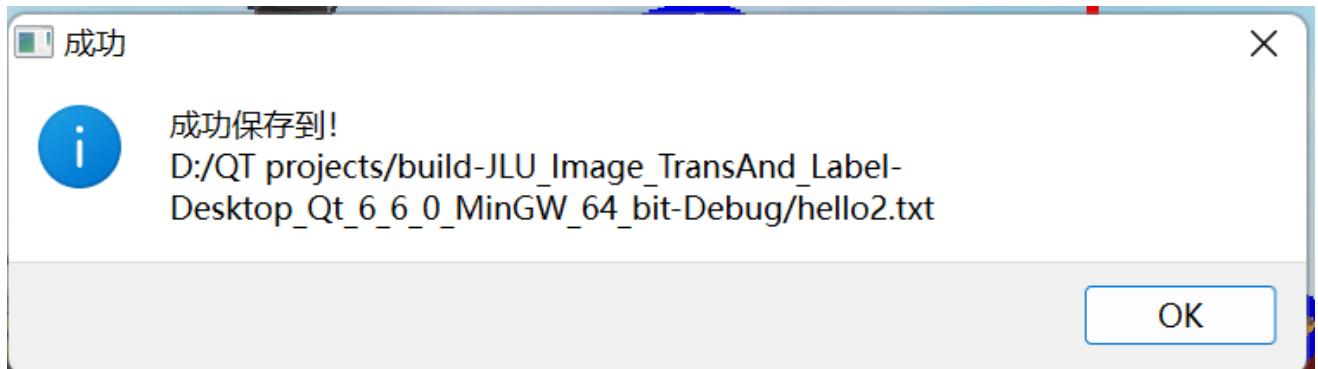


中途改变标注方式

你当然可以中途改变标注方式，代价是可能导致本次图片的标注数据清空。私人很不建议这样做，但是我依旧留了这个接口！

保存 (非自动版)

如果，你没有在开始标注的时候指定路径，那么，就默认你没有进行自动保存。那么，每一张图片标记结束后，都会询问你保存的路径和位置，以及文件名（这是为了兼容格式所作的，你可以自己在命名的时候做格式指定）



如你所见，此软件会提示你你保存的地址！

保存 (自动版)

如果你指定了，那么，好消息是，当你结束标记，你可以直接摁->使之前往下一个。

数据保存

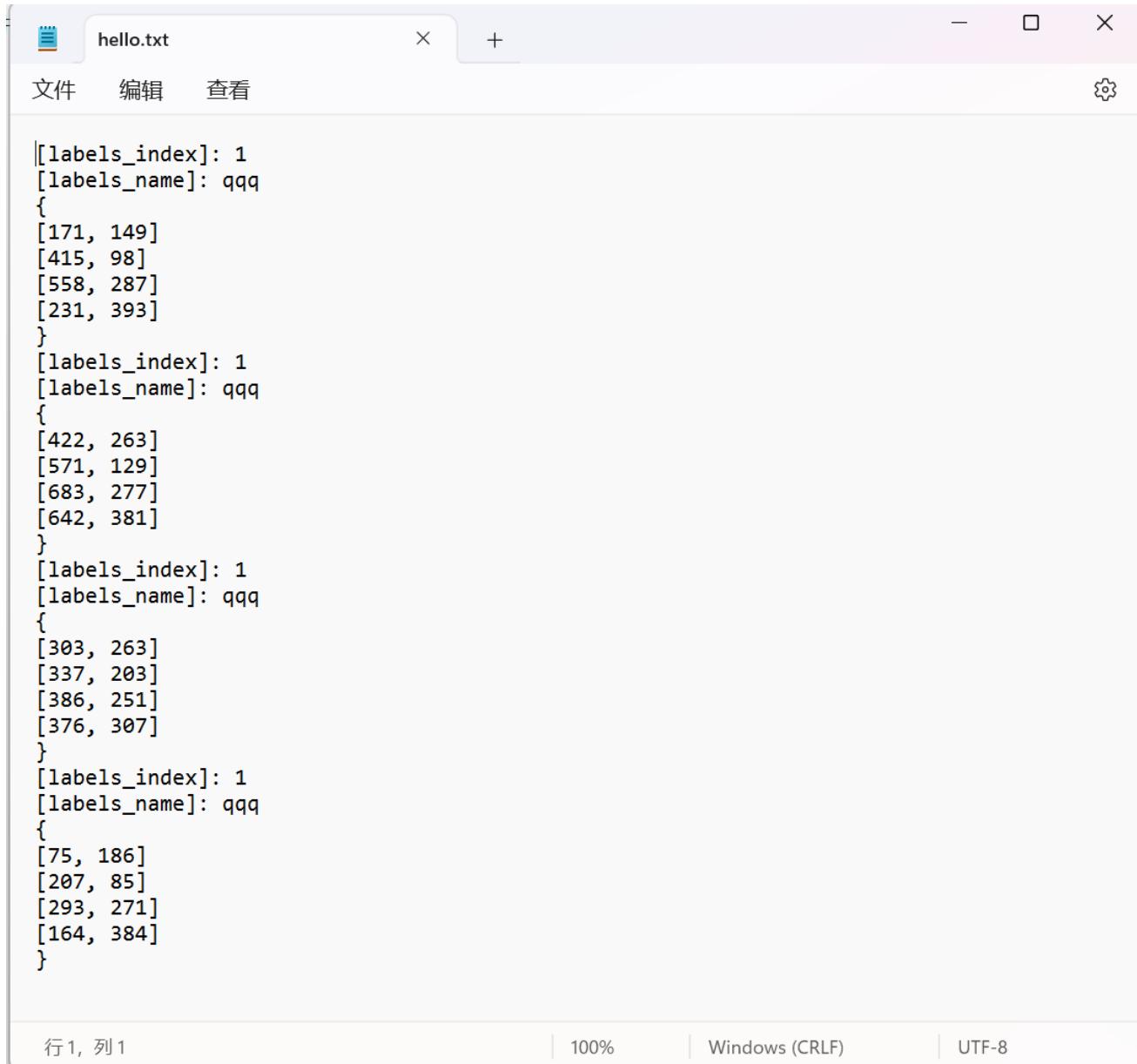
可以参见这个项目的curPicForLabeling_MainWindow类下的writingMethod函数，

目前软件提供YOLO标注和标准标注两种格式，可以自行参看，希望自定义的话，可以自己重写writeMethod接口

```
// 打印标签编号的前缀
#define LABEL_TEXT_INDEX_PREFIX          QString("[labels_index]: ")
// 打印标签名的前缀
#define LABEL_TEXT_LABLENAME_PREFIX      QString("[labels_name]: ")
// 打印点集的前缀
#define LABEL_TEXT_POINTSET_SHOW_PREFIX   QString("{")
// 打印单个点的坐标前缀
#define LABEL_TEXT_POINT_SHOW_PREFIX     QString("[")
// 换行
#define LABEL_TEXT_INSERT_SLASH          QString("\n")
// 点坐标分割符
#define LABEL_TEXT_POINT_SPLIT           QString(", ")
// 打印单个点的坐标后缀
#define LABEL_TEXT_POINT_SHOW_SUFFIX     QString("]")
```

```
// 打印X坐标的前缀  
#define LABEL_TEXT_POS_X_PREFIX           QString("x: ")  
// 打印Y坐标的前缀  
#define LABEL_TEXT_POS_Y_PREFIX           QString("y: ")  
// 打印点集的后缀  
#define LABEL_TEXT_POINTSET_END_SHOW      QString("}")
```

这是函数内使用到的一些宏，作为下一次GUI接口的更改，可以使用之。



The screenshot shows a text editor window titled "hello.txt". The content of the file is a JSON-like structure representing a point set. It consists of multiple objects, each with an "index" field (containing a value of 1) and a "name" field (containing the string "qqq"). Each object is enclosed in curly braces {}, and there are square brackets [] around the coordinate pairs. The coordinates listed are: [171, 149], [415, 98], [558, 287], [231, 393], [422, 263], [571, 129], [683, 277], [642, 381], [303, 263], [337, 203], [386, 251], [376, 307], [75, 186], [207, 85], [293, 271], and [164, 384].

```
[{"labels_index": 1, "labels_name": "qqq",  
 {"[171, 149], [415, 98], [558, 287], [231, 393]},  
 {"[422, 263], [571, 129], [683, 277], [642, 381]},  
 {"[303, 263], [337, 203], [386, 251], [376, 307]},  
 {"[75, 186], [207, 85], [293, 271], [164, 384]}]
```

行1, 列1

100%

Windows (CRLF)

UTF-8

默认下的标签——点集产出。

自动标注



请!在!不含有中文的路径下使用这个功能!

连续图片集标注

写在前面: 本次自动标注更新是实验性的, 尚未由更加良好的集成! 首先, 自动标注只支持一个标签对一个物体标记, 换而言之, **它不支持两个以上的物体标记. 它不支持两个以上的物体标记. 它不支持两个以上的物体标记. 它不支持两个以上的物体标记.** 更好的扩展功能将会在下一次更新中再次添加.

这个功能于8月17日左右开发完毕, 采用的是opencv中Tracker追踪器. 值得注意的是我们采用的是传统视觉的算法, 难以保证其所自动标注的就是准确的. 由此, 他只是起一种辅助标记的作用! 仅良好的适用于表征连续运动的物体图片集.

! : 在开始标注之前, 请务必选好保存的位置(精确到文件夹即可!), 否则可能会出现保存失败的情况.

**! : 在开始标注之前, 请务必选好标签! 并且请确保你的标签列表中只有一个标签! (有点弱鸡 (悲)) **

随后, 可以单击自动标注图片集进行标记

确定ROI区域

我们的标注必须在您手动确定好ROI后才会进行追踪标记! 确定的方式很简单, 只需要在窗口中单击, 第一次单击确定左上角点, 第二次单击确定右下角点. 当你确定好ROI后, 你才可以进行下一次的标注.

进行标注

当你确认自动追踪的标记显示满足您的需求精度, 你就可以简单的单击键盘上的 **回车键**, 一般的, 我们的格式是:

```
 ${curPixelMapIndex}_AfterAutoLabel.txt
```

如果路径没有错误, 他将会出现在你指定的文件夹下面

ROI矫正

当你发现自动标注的ROI丢失严重, 可以自己重新设置ROI(跟ROI设定章节讲述的方法一样). 他将会开启一段新的追踪!

⚠ 缺陷

作为实验内容, 值得注意的是: 自动标注不会回退到上一张, 如果发现自己的标注不满意, 在v1.1.0版本中只能重新标注而无法进行编辑.