

# 人工智能导论第一次大作业

张蔚桐 2015011493 自55

2017 年 4 月 2 日

## 1 程序编译与运行

程序在Windows10 + Visual Studio 2015环境下采用C++，利用Qt5.7.0, OpenCV3, Eigen3开源库进行开发。如需要编译此程序，请在以上环境或兼容环境中编译。

请在64位Windows10系统内运行本程序

如图1是程序的运行界面。单击左侧query files一行中右侧”...”按钮打开文件选择对话框，如图2所示。选择需要的处理文件

需要注意的是，源文件必须是txt格式，同时，受到开发时间的限制，程序没有对文件是否合法进行进一步的检查，因此请使用和给定样例相同格式（32\*32）的文件，否则可能导致未知的后果

同时，如图2所示，程序允许选择多个待处理的文件，点击open打开这些文件，回到程序运行主界面

单击左侧template files一行中右侧”...”按钮打开文件选择对话框，如图5所示。选择需要的模板

需要注意的是，源文件必须是txt格式，和前面处理文件不同的是，程序不允许选择多个模板文件，也就是说，程序仅能就一个模板进行多次的匹配操作。同样，点击open打开这些文件

单击左侧output directory一行中右侧”...”按钮打开文件选择对话框，如图10所示。选择需要的输出文件目录

需要注意的是，在这里只能选定一个文件目录，输出的文件按照作业中给出的ground truth的格式，将以”output\_i.j.txt”的名称输出到目标目录下。当然，可以不选择输出文件目录，这时，程序将不进行结果输出

回到程序运行界面发现要求的三个路径已经填写好，如图9所示，此时点击reset按键将清空选定的信息，点击OK开始执行程序

程序执行的过程中，会在右侧显示刚刚执行完成的图像的信息，同时下方进度条将显示整个批量执行的过程。当进度条为100%时程序执行完毕，

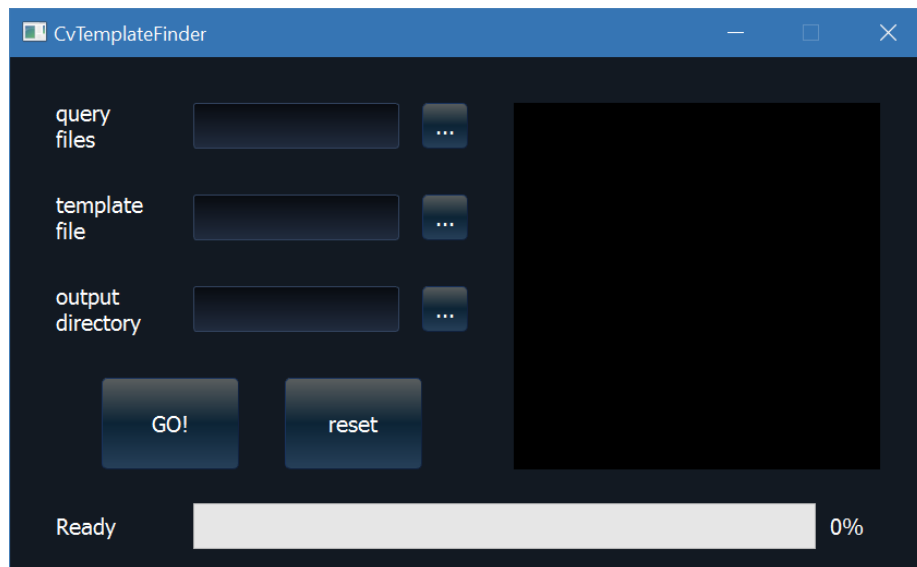


图 1: 开始界面图

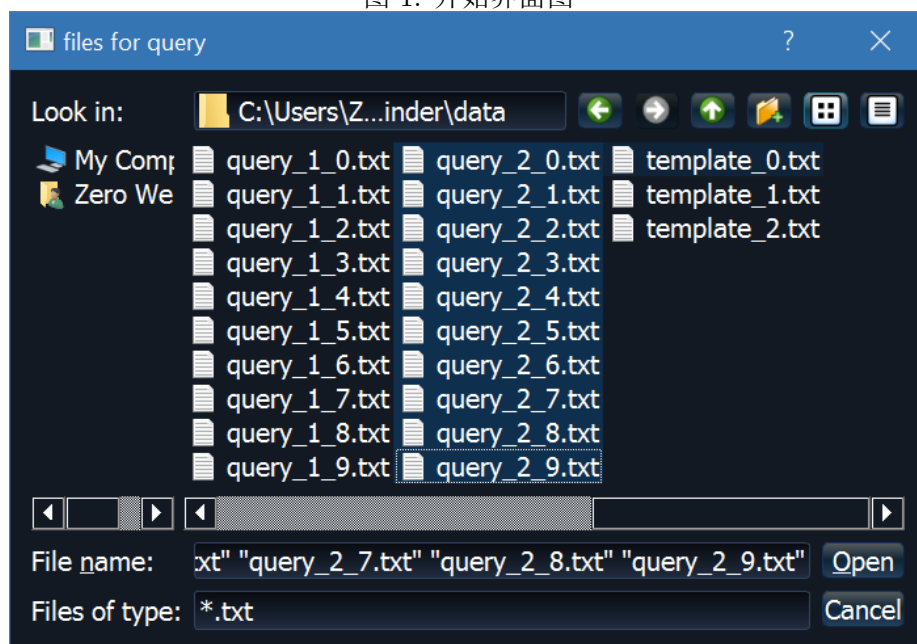


图 2: 选择处理的文件

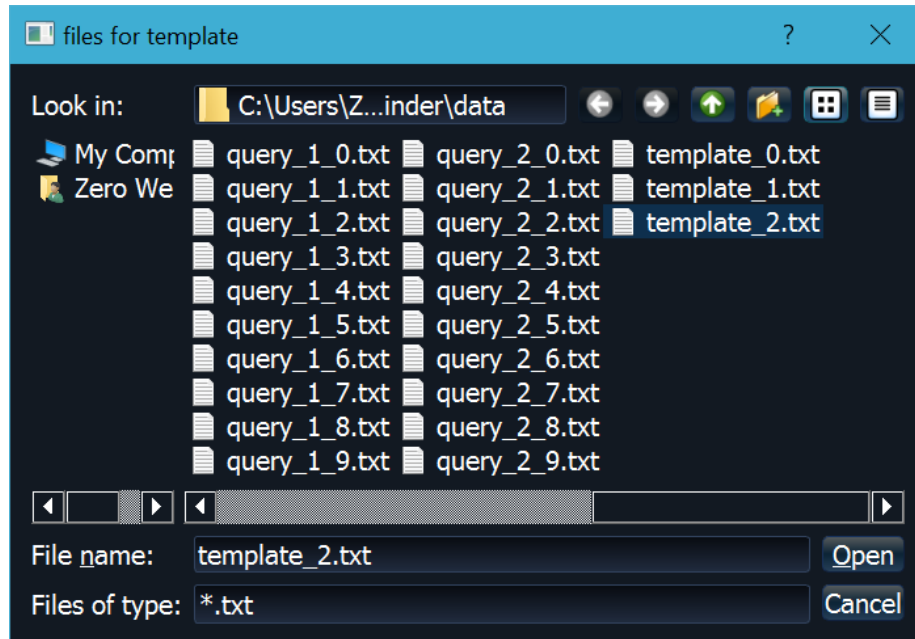


图 3: 选择模板文件

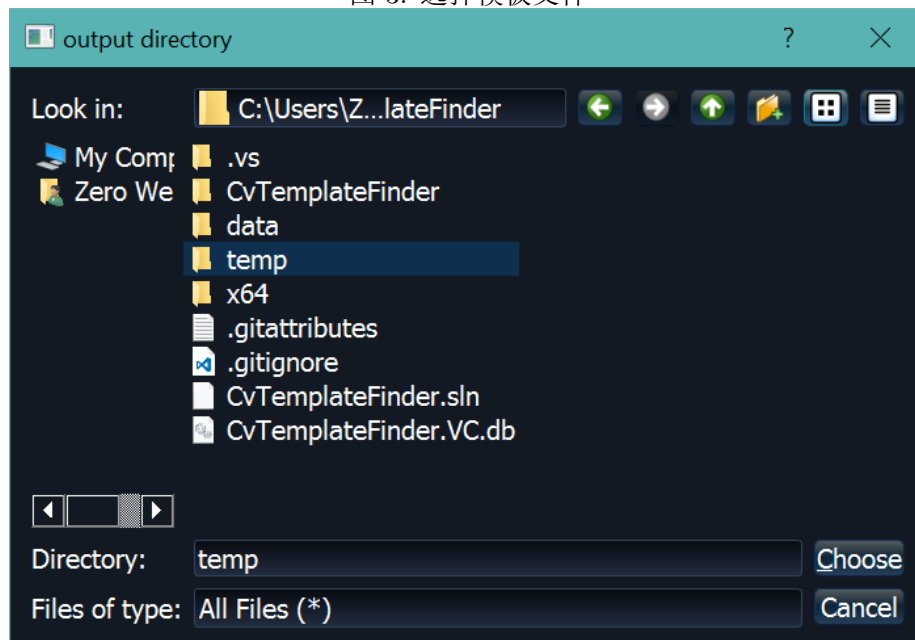


图 4: 选择输出文件目录

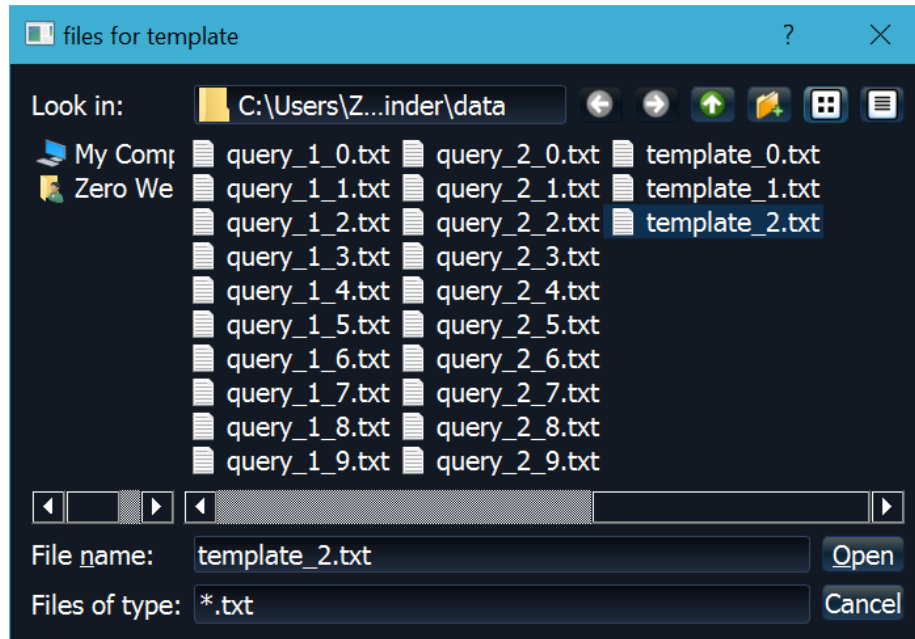


图 5: 选择模板文件

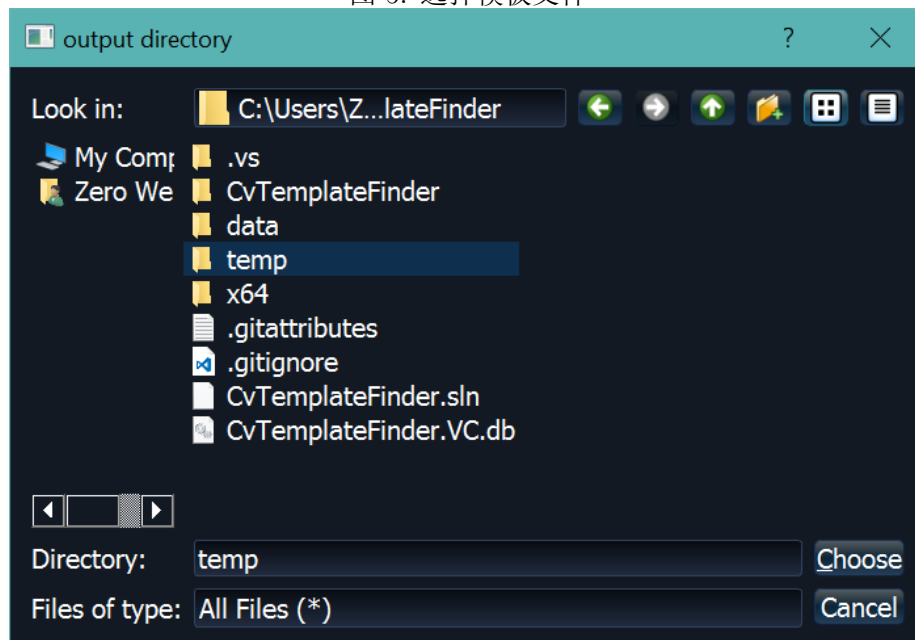


图 6: 选择输出文件目录

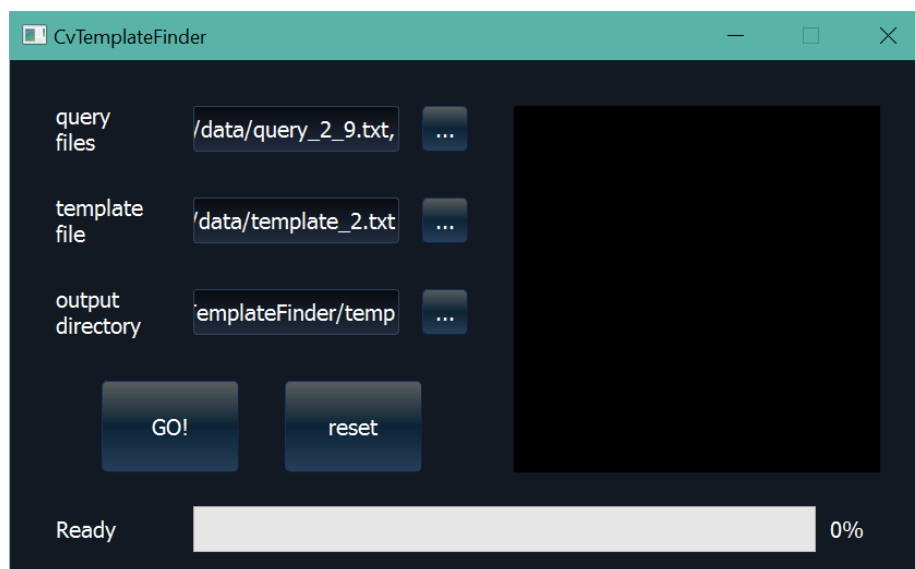


图 7: 文件选定后界面

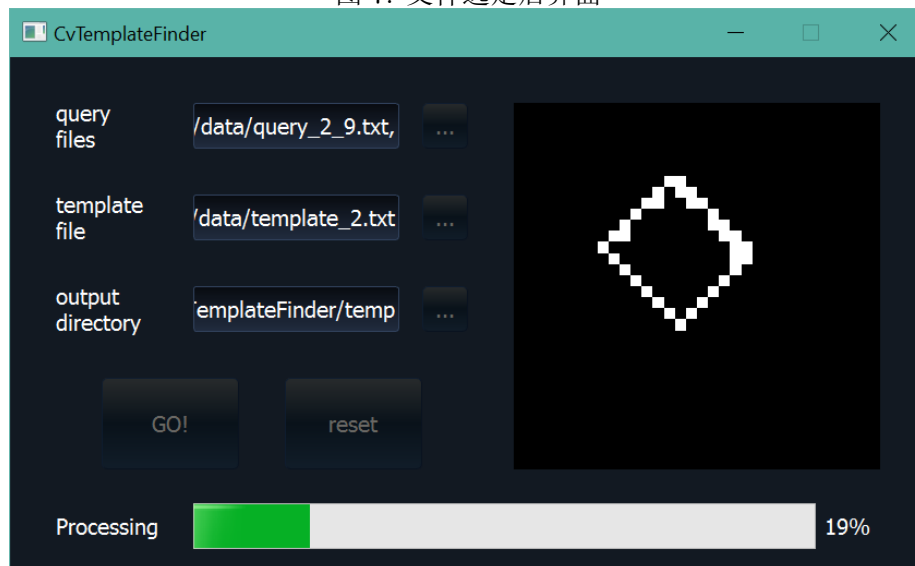


图 8: 程序运行界面

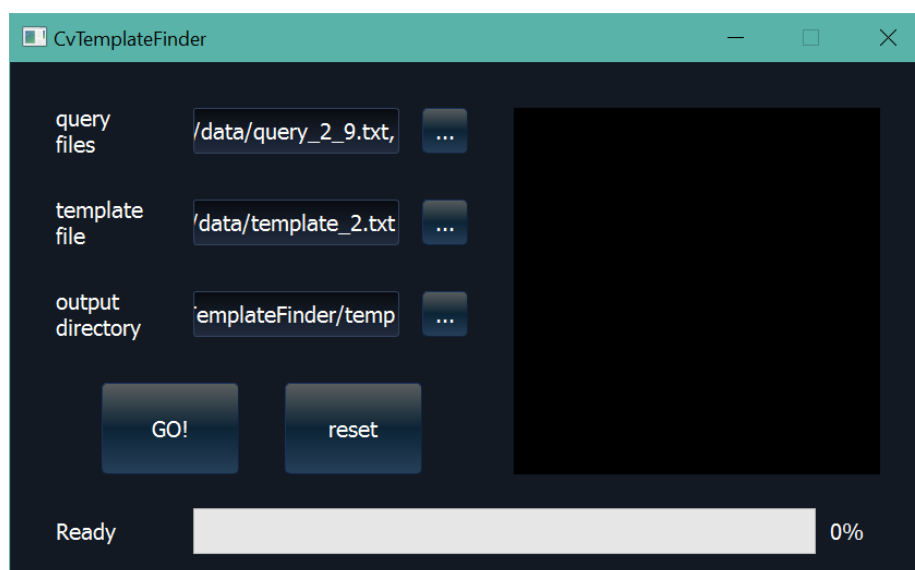


图 9: 文件选定后界面

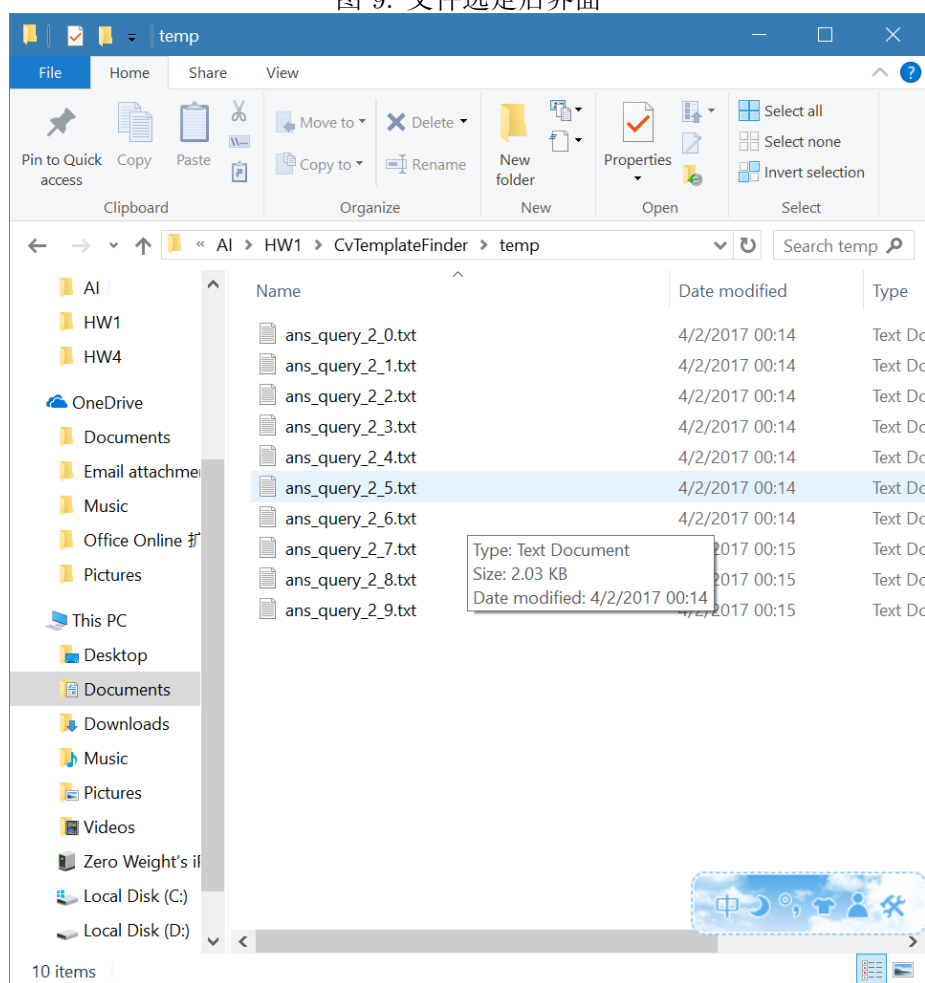


图 10: 输出文件

输出文件在选定的目录下，如图10所示

整个程序执行过程中需要注意的一点是，程序并不会对路径信息和文件的合法性进行检查，因此建议采用上述方式选择对应的文件夹和文件，不要修改左侧显示文件路径信息的文本框。除非保证输入的路径完全正确，否则会导致未定义的行为

## 2 搜索算法简述

程序首先对图片中的白色部分进行连通域检测，划分出图片中的每一块白色连通域，这一部分由openCV中的函数和BFS衍生的函数完成

其次，程序对每个检测出的白色连通域单独检测。将这块连通域单独放入一个32\*32的黑色图像中，对黑色部分进行连通域检测，如果发现全部的黑色部分均是一个连通域，说明白色连通域不构成闭合曲线，不符合题意，舍去，如果黑色连通域没有包含全图中的黑色区域，则说明白色部分是一个闭合曲线，保留之。这部分算法由BFS衍生而来，同时，为了处理有些图像中白色曲线出现“裂缝”情况，对白色图像在辨识之前进行模糊化处理。（保留的白色图像是未经模糊化的图像）

之后对检测剩余的图像进行剪枝。剪枝算法由简单的迭代完成，检查图片中的所有白色像素点，如果出现了周围白色像素点小于3个的白色像素点，我们认为这个点是一个分支的端点，将其染为黑色。逐次迭代直到没有这样的点

经过这些搜索，图像的F1-score已经能达到80%。下面进行对线性变换的搜索提高准确度

利用搜索得到的图像中白色点的像素，解线性方程得到变换矩阵，验证变换矩阵的作用效果，将作用效果较好的矩阵得到的变换点留下，经过这样，F1-score已经能达到90%

至此，输出并保存文件，核心算法结束。核心算法利用了程序中给定的“闭合多边形”和“线性变换”的条件

## 3 其他工作量

除了完成核心算法外，同时设计了用户界面，提供比较友好的交互形式。同时用户界面支持批量处理，并提供进度指示。利用OpenGL完成了图形的绘制工作，使得程序对用户比较友好

同时，在思路，除了应用基本的图像搜索算法外，利用了题目中给出的“仿射...”等条件，在数据规模被减小到一定程度之后对模板和目标

图像之间的线性变换进行搜索，进一步（约10%）提升了准确度，但是，这种查找计算量相当大，因此只有利用搜索算法将查找集缩小到不超过40个的情况下才可以使用这种算法。本程序中，搜索算法和这种算法同时作用，提升了算法性能同时减少了运行时间