本次短学期主题为科研和工程实践中的C++编程，对于应用程序框架，工程配置，进程线程等都有理论上的学习，实践方面需要小组合作完成一个小的项目，这就要求我们具备团队协作的能力，以及辅助团队合作的版本控制工具等。

本次项目采用了MVVM模式进行开发，MVVM的优点有很多，比如低耦合，且可以把视图逻辑放在一个ViewModel里 面，让很多view重用这段视图逻辑以达到可重用性，在开发过程中我们也注意到，对于ViewModel可以进行简易的测试，这解决了以往界面难以测试的问题。

MVVM模式所实现的视图逻辑的分离，在开发的过程中明显感觉到对于团队合作效率的大大提高，同时这样的机制也便于后期对原有功能扩展和维护，因为当UI变化时，ViewModel中的逻辑不需要进行变化，反之亦然。

我们小组在经过商量后，定题为“基于MVVMV模式的复杂验证码识别”，在开发过程我遇到的第一个小困难是对于持续集成、持续交付、持续部署概念的理解，之后也花了不少时间去完成持续集成工具Jenkins的配置，逐渐体会到平常课程中的C++与科研和工程实践中的C++编程中的不同。

由于过去我所接触到的C++编程都属于小型的，所以对于Git之类的版本控制系统使用率很低，这次短学期项目则不一样了，是我们小组五个人共同协作的产品，使用Git，我们可以在同一时间对相同的文件进行更改，同时可以创建不同的分支，使得我们可以独立的工作在不同的功能上，然后在各自完成后合并这些功能，大大提升合作的效益，在开发的过程中，我们对Git以及Github的使用变得越来越熟练，受益匪浅。

在参与进行View层设计的时候，感受到MVVM模式对于View层开发的大大简化，不再需要过多的考虑到逻辑层面，这样一来，界面的设计变得更加纯粹了，设计的时候可以对设计本身投入更多的精力，而不用分心去考虑其他部分，而MVVM模式这对于ViewModel的设计则显然带来了更大的挑战。

在为期近两周的时间里，我们小组完成了“基于MVVM模式的验证码识别”，最后看到成品小组成员都挺满意的，这也是小组这些天的不断付出得到的回报，在项目开始阶段，组长对于分工有一个比较合理的安排，对于项目日程也有一个清晰的规划，整个项目开发过程中，掌握了一个比较好的开发节奏，团队的协作也十分融洽，每个人都处于一个合适的开发角色，对于项目都付出了自己的心血。

总的来说，这次的短学期，不仅让我认识到了科研和工程实践的C++编程框架的合理选择的重要性，以及工具的恰当使用的重要性，还认识到了团队协作的确是对于大型项目开发的不可或缺的能力。