开发环境搭建：OpenCV 和 Qt 安装
OpenCV and Qt
—《机器视觉技术实训》课程笔记

杨青青

2020 年 2 月 25 日

Email：qqyang@nit.zju.edu.cn

本次课程介绍 Linux 中进行机器视觉开发所需要的两个重要开发库：OpenCV 和 Qt。OpenCV 是开源的计算机视觉库，Qt 是跨平台的图形界面库，两者都提供 C++ 开发接口。掌握 OpenCV 和 Qt 开发库对于机器视觉的开发实践是很有帮助的。

OpenCV

OpenCV 简介

OpenCV 是 Open Source Computer Vision Library 的简写，是一个用于计算机视觉开发的开源、免费的程序库，官网：`https://opencv.org/`。OpenCV 从创建至今，已经包含 2500 多个经过优化的计算机视觉相关的算法，分布在不同的模块 (module) 中。OpenCV 是开源的，意味着所有的源代码都可以参考、使用；OpenCV 的主代码库是完全免费的，可以用于商业产品开发。

我们可以在 Github 上获得 OpenCV 的源代码：`https://github.com/opencv`。其中，opencv(`https://github.com/opencv/opencv`) 是主代码库，完全免费；opencv_contrib(`https://github.com/opencv/opencv_contrib`) 是附加模块，大部分代码都是免费的，但也包含了一些专利保护的算法。在本课程中，我们尽量使用 OpenCV 的主代码库。

安装 OpenCV

在 Ubuntu Linux 上安装 OpenCV，推荐使用源代码编译的方式。

Step 1：安装工具程序

在前一次课程中，我们已经安装了 GCC、CMake 和 Git 工具，因此已经具备了程序开发的基础工具，可以通过源码编译 OpenCV 了。这里将 GCC、CMake 和 Git 工具的安装再进行简要说明，已经完成安装的同学可以跳过第一步。

- 更新软件源：

```
$ sudo apt update
```

- 更新所有软件包：

```
$ sudo apt upgrade
```

- 安装 C/C++ 编译器[1]：

```
$ sudo apt install build-essential
```

- 安装 CMake 和 CMake 命令行 GUI：

```
$ sudo apt install cmake cmake-curses-gui
```

- 安装 Git：

```
$ sudo apt install git
```

[1] 安装这个包会同时安装 C 和 C++ 开发所需要的编译器和标准库。

Step 2：获取 OpenCV 源代码

获取 OpenCV 的源代码一般有两种不同的方式：

1. 在 OpenCV 或者 GitHub 上下载稳定版本的压缩包。稳定版本即 release 版本，是经过测试、错误较少的版本，可以直接用于开发。我们可以通过在 OpenCV 官网的 Releases 页面（`https://opencv.org/releases/`）或者 GitHub opencv 代码的 releases 页面下载所需的稳定版本。截止本文档撰写时间，OpenCV 4 的最新稳定版本号为`4.2.0`，OpenCV 3 是前一个大版本，目前处于维护中，其最新版本为`3.4.9`[2]。将压缩包解压后，即可得源代码文件夹。本次课程中，我们使用 OpenCV 3。
2. 如果需要经常更新 OpenCV 库，则可以选择使用 Git 工具拉取整个代码仓库[3]。我们可以直接使用最新的代码；也可以通过`git tag`指令查看所有加标签的版本（一般为 release 版本号），然后使用`git checkout <`版本号`>`指令即可签出所需的稳定版本。当有新的版本时，我们可以从远程仓库拉取最新的更改。这种方法适用于需要追求并测试使用 OpenCV 最新算法功能的同学。

[2] 一般来说，OpenCV 都会同时推出两个主版本，最新的版本，例如 OpenCV 4，会不断往里面添加新的算法和功能，而前一版本，如 OpenCV 3，新的算法或功能将不再添加，而仅仅是在原来算法功能上进行维护升级。

[3] 注意：第一次拉取代码仓库可能需要比较久的时间。

本文以 OpenCV 主代码库模块为例进行讲解，在大家对编译流程和 CMake 等工具熟悉以后，就可以增加 opencv_contrib 模块的编译了。

Step 3：安装依赖工具和依赖库

做好以上准备之后，在编译源代码前，还需要在系统中安装一些依赖工具和依赖库。

1. 添加编译时所需的必要工具`pkg-config`。该工具在编译程序或库文件时能够在命令行中提供正确的编译选项，例如，它能够在编译时找到已经安装的库。可以通过下面的指令安装：

   ```
   $ sudo apt install pkg-config
   ```

2. 添加基本的依赖库。OpenCV 中的很多算法功能需要依赖一些其它的开发库，例如，线性代数运算库、矩阵运算库、图形库等等。下面的命令添加必须（required）的依赖库：

   ```
   $ sudo apt install libgtk2.0-dev libavcodec-dev
   libavformat-dev libswscale-dev
   ```

   对于每个依赖库的具体用处请同学们自己上网查询[4]。

3. 安装可选的依赖库。下面的依赖库都不是必须的，可以选择安装：

   ```
   $ sudo apt install libtbb2 libtbb-dev libjpeg-dev
   libpng-dev libtiff-dev libjasper-dev libdc1394-22-dev
   ```

   一般来说，现在的电脑很少配置 1394 火线接口了，所以`libdc1394-22-dev`一般都不安装。

[4] 这里需要指出的是，在 Linux 系统中，一般结尾是“-dev”的包都是开发库包，会安装该开发库的相关头文件、链接库文件和运行时二进制文件等所有内容。

Step 4：编译源代码

现在，我们可以开始正式编译 OpenCV 源代码了。我们在本次教程中只会编译默认的最小化安装，有些没有添加依赖库的模块我们将不进行编译。可以通过执行下面的指令进行默认编译：

```
$ cd <opencv所在目录>
$ mkdir build && cd build
$ cmake ..
$ make
$ sudo make install
```

这里对每一行指令进行简单介绍。第 1 行指令是 Linux 基本指令，进入到 opencv 源代码所在的目录；第 2 行指令在当前目录下建立一个名为 build 的子文件夹，然后进入到 build 文件夹；第 3 行使用 cmake 进行配置，“..”表示上一层目录，即 opencv 的源代码目录，因为 cmake 依赖一个 CMakeLists.txt 的文件进行配置，该文件在 opencv 的源代码目录中；第 4 行使用`make`指令进行编译，该指令利用 cmake 配置生成的 Makefile 文件对源代码进行编译，然后链接相关的依赖库。

实际上，在第 4 步完成后，OpenCV 的库函数已经编译好了，存放在 build 文件夹中，但是，为了让后续开发过程中能够顺利找

到 OpenCV 的库，我们选择将其安装在系统库目录中，这个操作就由最后一步完成[5]。

如果严格按照本文的介绍执行了上面的所有配置，编译所需的这 5 个步骤应该能够顺利执行。但是对于第一次尝试编译 OpenCV 源代码的同学来说，在整个编译过程会遇到不同的问题。这就需要同学们通过搜索引擎在网上找到相应的方法[6]。

[5] 注意：最后一步我们使用了 sudo，说明我们需要获取管理员权限才能执行

[6] 一个常见的问题是 IPP 模块，因为网络问题容易下载失败，在课程的演示视频中介绍了手动加载的方法，另外，也可以选择将其禁用。可以通过 ccmake 打开 cmake 的命令行 GUI，找到 WITH_IPP 选项，将其禁用。

测试 OpenCV

OpenCV 安装完成后，我们可以写一份简单的代码进行测试。本节通过一个简单的图像读取与显示程序，介绍在 Linux 环境中如何使用 OpenCV 来进行开发。OpenCV 除了 C++ 的编程接口外，还提供了 Python 和 Java 的编程接口。本课程使用 C++ 进行编程。

在 Windows 环境下，大家可能已经熟悉了 Visual Studio 来构建项目，然后在 VS IDE 中进行开发。在 Linux 下，我们使用 CMake 工具来构建项目。所有的程序开发只需要命令行和一个文本编辑器[7]即可完成。

[7] 在 Linux 环境中的两大文本编辑器是 Emacs 和 Vim，但是对于初学者，我推荐 Visual Studio Code。如果不想太折腾编译器，可以选择使用系统自带的 gedit。我使用 Emacs，本课程所有的演示文件、课程笔记和示例代码，都是在 Emacs 中完成的。

首先，我们建立一个文件夹用于存放源代码，这个步骤就不再详述。

然后，我们需要打开我们的文本编辑，建立一个新的文本文件，作为程序代码文件。我取名为“`display_image.cc`”，这里，后缀名“`.cc`”表示这是一个 C++ 源文件。在源文件中录入下面的代码：

```
1  // 本程序所需要的opencv头文件
2  #include <opencv2/core.hpp>
3  #include <opencv2/imgcodecs.hpp>
4  #include <opencv2/highgui.hpp>
5  // C++标准程序库头文件
6  #include <iostream>
7  #include <string>
8
9  /**
10  *  主程序 : 通过opencv的imread函数读入一张图片，然后在窗口中显示。
11  *
12  *  Usage: display_image <image_filename>
13  */
14 int main(int argc, char* argv[]) {
15   if (argc < 2) {
16     std::cout << "Usage: display_image <image_filename>" << std::endl;
17     return 0;
18   }
19
20   std::string image_name(argv[1]);
21   cv::Mat image = cv::imread(image_name, cv::IMREAD_COLOR); // 以彩色图像方式读入
22   if (image.empty()) {
```

```
23      std::cout << "Cannot open or find image." << std::endl;
24      return -1;
25    }
26    cv::namedWindow("Display Window", cv::WINDOW_AUTOSIZE); // 创建显示图片的窗口
27    cv::imshow("Display Window", image);                    // 在窗口中显示图片
28
29    cv::waitKey(0);                 // 等待任意按键关闭窗口
30    return 0;
31  }
```

在编写好程序代码后,我们需要编写一个名为CMakeLists.txt的文件，来构建 CMake 项目。一个最简的 cmake 文件如下：

```
1  # cmake所需的最低版本要求
2  cmake_minimum_required(VERSION 2.8)
3
4  # 设置项目名称
5  project(display_image)
6
7  # 设置C++11支持
8  set(CMAKE_CXX_STANDARD 11)
9
10 # 寻找 OpenCV 库
11 find_package(OpenCV REQUIRED)
12
13 # 加入OpenCV头文件目录
14 include_directories(${OpenCV_INCLUDE_DIRS})
15
16 # 添加项目的输出程序和所需源代码
17 add_executable(display_image display_image.cc)
18
19 # 将OpenCV库链接到程序中
20 target_link_libraries(display_image ${OpenCV_LIBS})
```

重要的语句都已经在源代码中进行了注释。下一步的编译过程和 OpenCV 的源代码编译相似：

```
$ cd <code_path>
$ mkdir build && cd build
$ cmake ..
$ make
```

相信大家已经对这些命令很熟悉了。执行完这些指令，就会生成一个display_image的可执行文件。这个程序需要数据一个命令行参数，即需要显示的图片的名称。OpenCV 已经支持了一些主流的图片格式。指定任意一张图片，该程序就会弹出一个窗口对图片进行显示。

Qt

Qt 简介

Qt 是一个跨平台的软件工具包，主要用于图形用户界面设计。Qt 具有商业版和开源版两个版本[8]，商业版提供更加强大的工具包和 Qt 的官方技术支持，但是需要付费（价格不菲）；开源版包含了 Qt 的所有基本功能和大部分的附加功能和工具，基于 (L)GPL v3 协议[9]使用。

英文网址：`https://www.qt.io`
中文网址：`https://www.qt.io/cn`。

[8] 两个版本的详细比较请见官方网站：`https://www.qt.io/cn/download`。

[9] 如果仅使用免费模块的链接库，可以进行商业用途，不需要开放源代码；但如果改动其原始代码，则需要开源所有修改的内容。当然，我们课程的所有示例代码都是开源的，不存在版权问题。

Qt 的功能是以模块的方式提供的，这一点和 OpenCV 非常相似，其中，基本功能包含（括号中是对应功能模块的名称）：Qt Core（Qt Core）、UI 技术（Qt GUI, Qt Widgets、Qt Quick 等）、网络和连接（Qt Network）、多媒体（Qt Multimedia, Qt Multimedia Widgets）、数据存储（Qt SQL, Qt Core - JSON classes）、国际化（Internationalization）、测试框架（Qt Test）。

除了基本功能外，Qt 的开源版还提供了多个附加功能，包括 3D/2D 图形和图像处理等。另外，Qt Creator IDE 和工具是很实用的 Qt 开发工具，大家可以将其当作 C++ 的一个 IDE 使用。

安装 Qt

在 Ubuntu 中安装 Qt 有两种方法：

- 方法一，在 Qt 官方网站下载安装文件进行安装[10]。在注册下载后会在本地获得一个 `.run` 文件，该文件类似于 Windows 下的 `.exe` 文件。例如对应的最新版本，下载后的文件名是 `qt-opensource-linux-x64-5.14.1.run`。在 shell 中进行安装，参考以下步骤进行即可：

  ```
  $ cd <Qt下载目录>
  $ chmod +x ./qt-opensource-linux-x64-5.14.1.run
  $ ./qt-opensource-linux-x64-5.14.1.run
  ```

  这里，第一个步骤将当前工作目录转换到 Qt 下载的目录；第二个步骤将下载的安装文件赋予可执行权限[11]；第三步直接运行该文件进行安装。

[10] 在`https://www.qt.io/cn/download`页面选择开源版，点击最下方“Go open source”按钮。注意：下载开源版需要进行注册。安装文件分为在线安装包和离线安装包，从默认的 download 界面下载的是在线安装包，离线安装包可从这个页面下载：`https://www.qt.io/offline-installers`。

[11] 在 Linux 中，每一个文件都有三组权限值，分别是用户、组和其它用户权限；每一组权限值都包含 rwx 三个标记，其中 r 代表可读，w 代表可写，x 则代表可执行。如果没有某一个权限，该标记会用“-”替代，例如“rw-”就表示只可读写，不能执行。只有具有可执行权限的文件，才可以

- 方法二，使用 APT 进行安装：

  ```
  $ sudo apt install qt5-default qtcreator -y
  $ sudo apt install qt5-doc qt5-doc-html qtbase5-doc-html qtbase5-examples -y
  ```

  命令最后的 `-y` 选项表示确认安装，不会再显示确认安装提示。

对于一般的 Ubuntu 桌面系统进行 Qt 开发，推荐使用第一种方式进行安装。本课程后期会使用 Jetson TX2 平台进行实际操作，因此选择方法二进行安装，保证所使用的包的兼容性最优[12]。

[12] 注意，这两种安装方式的安装目录不一样。一般第一种安装方式会安装在用户家目录下，而使用 apt 的安装方式则会安装在默认的系统安装目录下。

运行一个示例

在安装完成后，可以打开 Qt Creator 工具来浏览并运行示例。下面，我们使用一个 Qt Widgets 示例 Image Viewer 来尝试一下简单的 Qt 程序。

Qt Creator 是一个集成开发环境，可以在该环境种完成所有的 Qt 程序开发、调试和测试。大家也可以将其作为一个代码编辑器来使用。在系统菜单栏里搜索 Qt，就可以找到 Qt Creator 应用，点击打开会得到如图 1 启动界面：

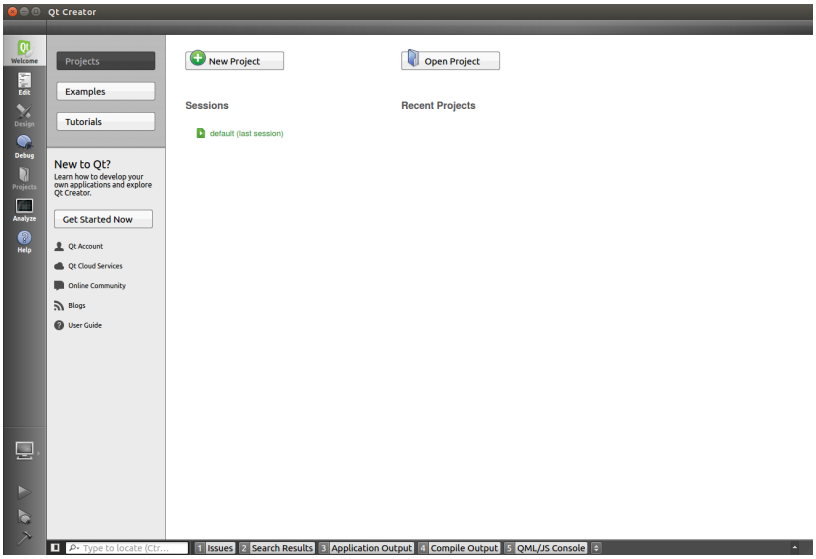

图 1: Qt Creator 启动界面。默认是项目选项卡。

启动界面默认的是 Projects 标签，选择 Examples 标签浏览示例。该页面上可以看到所有已经安装的 Qt 示例程序：

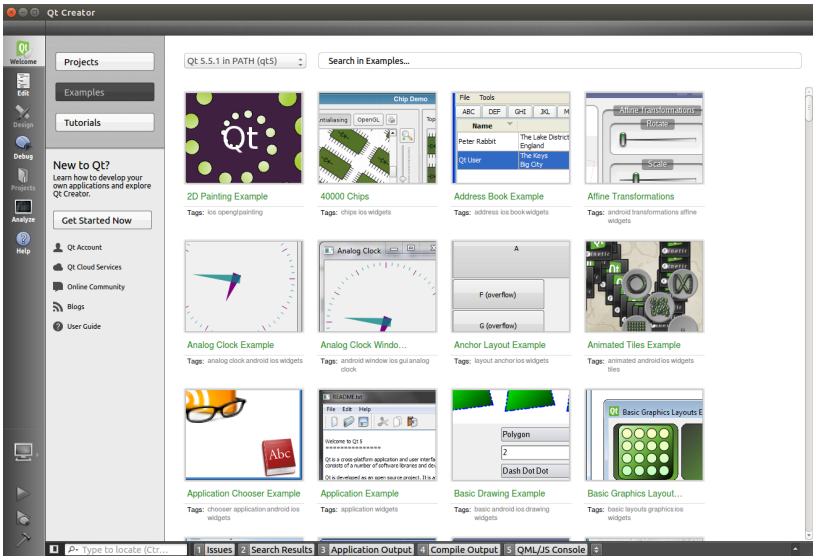

图 2: Qt Creator 示例程序浏览界面。

可以选择其中一个示例，会弹出一个窗口提示进行代码复制（如

图 3所示）。

在复制路径中，选择一个合适的路径将该示例工程进行复制。在这个过程中，也可以点开“Browse..”按键（图 3 中高亮的按钮），选择已存在的目录或者新建目录来拷贝这个示例工程。选择“Copy Project and Open”按钮进行工程拷贝，并打开该 Qt 工程。这里，我们选择打开了一个名为“Image Viewer”的示例工程。在打开后，Qt Creator 会跳转到项目配置页面，同时打开一个文档窗口，显示该示例项目的说明（如图 4 所示）。

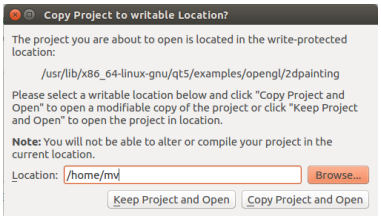

图 3: 复制示例程序选项框。可以选择示例程序的拷贝路径，并打开拷贝完成的示例程序项目。

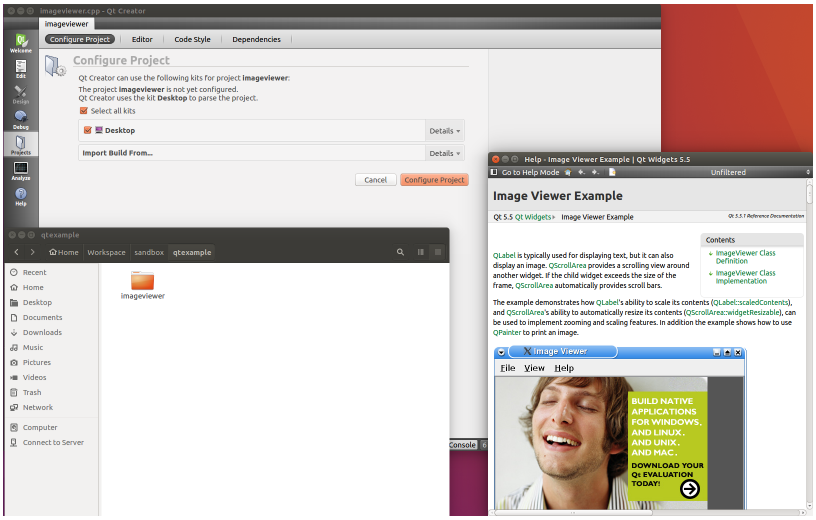

图 4: 在 Qt Creator 开发示例代码工程，同时会打开一个窗口，显示该示例代码的说明页面。

可以看到，“Image Viewer Example”是在“Qt Widgets”下的一个示例。在文件夹浏览器中，可以发现该示例工程已经复制到所选的路径下了。

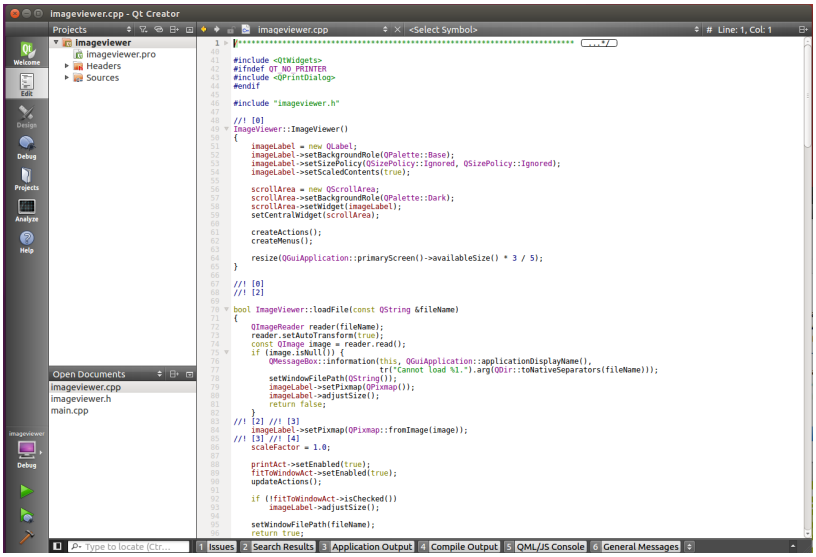

图 5: Qt Creator 中的项目开发页面。

在 Qt Creator 中接受默认的配置选项，点击高亮的按钮：“Configure Project”。这时 Qt Creator 会打开工程，并在代码编辑窗

口显示源代码（如图 5 所示）。点击左下角的绿色运行图标，Qt Creator 就会编译程序，如果编译正确就会打开编译完的程序。

这时，说明 Qt 的开发环境已经安装完成了。

小结

本节介绍了 Linux 环境下开发机器视觉应用所需要的软件库的安装，包括视觉库 OpenCV 和图形界面库 Qt。这两个库都提供了 C++ 的接口函数，需要大家具有一定的面向对象的基础。除了 OpenCV 是通过源码进行编译安装外，其它的开发库和依赖库都是通过 APT 进行安装。希望大家学习完本课程内容后，对 Linux 下的开发开发有更深入的了解。在后续课程中，我们将进一步介绍 OpenCV 图像处理相关的算法和 Qt 界面开发，并介绍将两个库结合使用的方法。

阅读和参考资料

[1] "OpenCV: Installation in Linux."[Online]. `https://docs.opencv.org/master/d7/d9f/tutorial_linux_install.html`.
该教程相对比较老，但是最为基础，可以借鉴使用。
[2] "Comprehensive guide to installing OpenCV 4.1.0 on Ubuntu 18.04 from source,"[Online]. CV-Tricks.com, 06-Jun-2019. [Online]. `https://cv-tricks.com/installation/opencv-4-1-ubuntu18-04/`.
该教程比较详尽，但是有很多配置是目前不需要的，推荐需要详细了解 OpenCV 编译的同学阅读。
[3] "Install Qt Creator on Jetson TX1,"[Online]. `https://www.jetsonhacks.com/2017/01/31/install-qt-creator-nvidia-jetson-tx1/`.
该教程介绍了在 Jetson TX1 平台安装 Qt 的方法，同样的方法适用于 TX2 和普通的 Ubuntu 系统。
[4] "Qt Widgets Documentation,"[Online]. `https://doc.qt.io/qt-5/qtwidgets-index.html`. Qt Widgets 是 Qt 桌面程序界面开发的主要库，多参考文档和示例程序有助于快速掌握 Qt 开发。

实践

- 参考演示视频通过源码编译安装 OpenCV。
- 在使用 CMake 编译并运行本笔记中 `display-image` 程序，并通过 Git 提交到 GitHub 的 `mvia-course-project` 中[13]
- 安装 Qt 并运行一个示例程序。

[13] 注意，请单独作为文件夹进行提交，仅提交源代码。
探索：查阅资料研究如何在 Git 版本管理中，忽略二进制代码的改动。