实验一 实验/开发环境的搭建

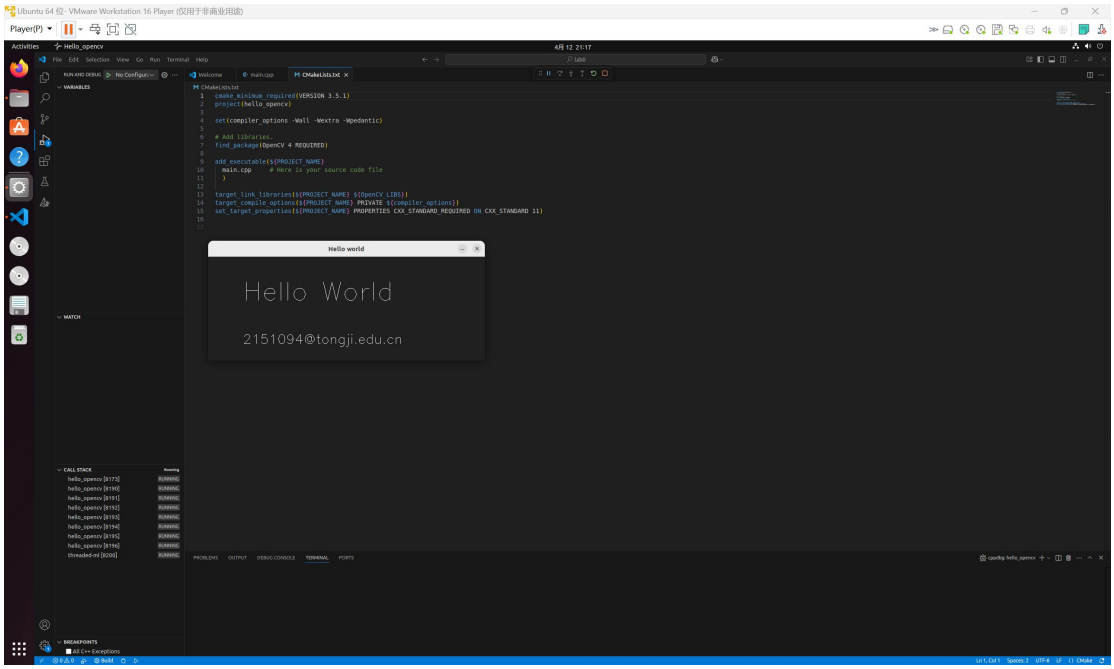
2151094 宋正非

1. 任务要求

根据网站教程安装完实验环境后，写“Hello World”的程序进行环境验证。使用CMake管理项目并驱动编译器生成可执行程序，使用VSCode进行C++编码和调试，显示一个GUI窗口来显示“Hello World”，内容包含“Hello World”以及学号或邮件地址。

2. 实现效果

输出“Hello World”并包含我的邮件地址“2151094@tongji.edu.cn”。



3. 实现代码

#main.cpp

```
#include "opencv2/opencv.hpp"

int main(int argc, char *argv[])
{
    cv::Mat image(240, 640, CV_8UC3, cv::Scalar(32,32,32));

    // Display "Hello World"
    cv::putText(image, "Hello World", cv::Point(80, 100), cv::FONT_HERSHEY_SIMPLEX, 2,
cv::Scalar(255, 255, 255), 1, 8, false);

    // Display email address
    cv::putText(image, "2151094@tongji.edu.cn", cv::Point(80, 200),
cv::FONT_HERSHEY_SIMPLEX, 1, cv::Scalar(255, 255, 255), 1, 8, false);

    cv::imshow("Hello world", image);
    cv::imwrite("Home/work/lab0/lab0.png", image);
    cv::waitKey(0);

    return EXIT_SUCCESS;
}
```

#CMakeLists.txt

```
cmake_minimum_required(VERSION 3.5.1)
project(hello_opencv)

set(compiler_options -Wall -Wextra -Wpedantic)

# Add libraries.
find_package(OpenCV 4 REQUIRED)

add_executable(${PROJECT_NAME}
  main.cpp     # Here is your source code file
  )

target_link_libraries(${PROJECT_NAME} ${OpenCV_LIBS})
target_compile_options(${PROJECT_NAME} PRIVATE ${compiler_options})
set_target_properties(${PROJECT_NAME} PROPERTIES CXX_STANDARD_REQUIRED ON
CXX_STANDARD 11)
```

cv::Mat 是 OpenCV 用来表示图像的基本数据结构，通过 cv::Mat 我们可以对图像进行读取、修改、保存等操作。cv::Mat(int rows, int cols, int type, const Scalar& color = Scalar(0, 0, 0)); 前两个参数分别定义了图像的高度与宽度，**rows** 为图像的行数，即高度；**cols** 为图像的列数，即宽度；单位均为像素。**type** 表示图像的深度和通道数的组合，CV_8UC1 为单通道、8 位无符号整数，显示为灰度图；CV_8UC3 为三通道、8 位无符号整数，显示为彩色图；CV_32F 为单通道，32 位浮动数。**color** 为图像初始颜色值，类型为 Scalar，表示颜色的 BGR 三个分量。cv::Mat image(240, 640, CV_8UC3, cv::Scalar(32,32,32)); 表示我创建了一副宽为 240 像素，长为 640 像素，类型为三通道彩色图，颜色为暗灰色的图像。

cv::putText 用于在图像上绘制文本。cv::putText(cv::Mat& img, const std::string& text, cv::Point org, int fontFace, double fontScale, const cv::Scalar& color, int thickness = 1, int lineType = 8, bool bottomLeftOrigin = false); **img** 为输入的图像，类型为 cv::Mat，文本将被添加到这张图像上。**text** 为将要绘制的文本内容，类型是 std::string。**org** 是文本的起始位置，类型为 cv::Point，它表示文本左下角的位置。**fontFace** 为字体类型，常见的包括 cv::FONT_HERSHEY_SIMPLEX 简单的无衬线字体，cv::FONT_HERSHEY_COMPLEX 复杂的无衬线字体，cv::FONT_HERSHEY_DUPLEX 类似简单字体但线条稍微复杂一些。**fontScale** 为字体大小缩放因子，该值越大则字体越大。**color** 为文本的颜色，类型为 Scalar，表示 BGR 三个分量的颜色。**thickness** 为字体粗细程度，单位为像素，默认为 1。**lineType** 为线条类型，用于指定文本边缘的抗锯齿方式，常用的值是 8 连接类型，为默认值，表示标准抗锯齿；cv::LINE_AA 表示抗锯齿线，通常用于提高文本的显示效果。**bottomLeftOrigin** 是一个布尔值，表示文本是否以左下角为原点绘制；默认为 false，表示文本是从左上角开始绘制的；若设为 true，文本将从左下角开始绘制。cv::putText(image, "Hello World", cv::Point(80, 160), cv::FONT_HERSHEY_SIMPLEX, 2, cv::Scalar(255, 255, 255), 1, 8, false); 表示在 image 图像上绘制“Hello world”字符串，起始位置为（80，160），使用 FONT_HERSHEY_SIMPLEX 字体，字体大小为 2，颜色为白色，粗细程度为 1，线条类型为 8 抗锯齿，文本以左上角为起始点。