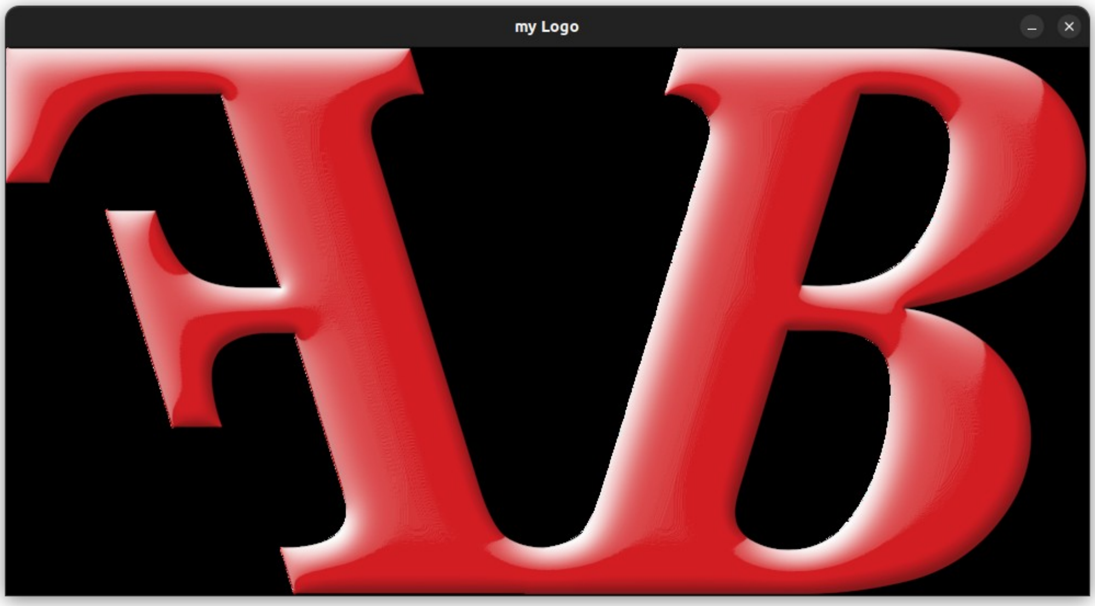
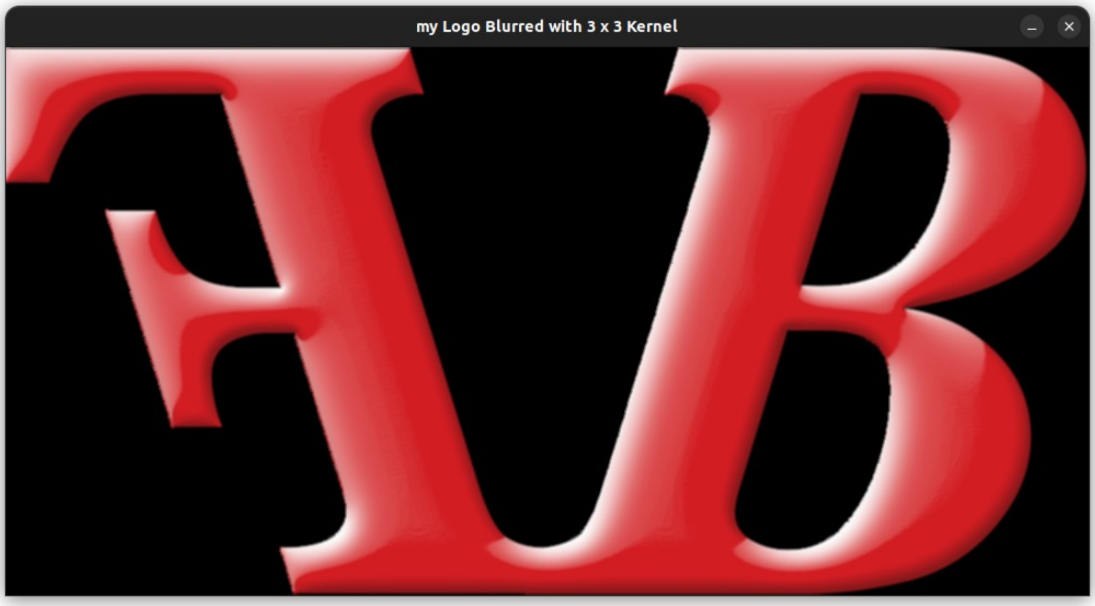
Compiling and run my_logo_image.cpp

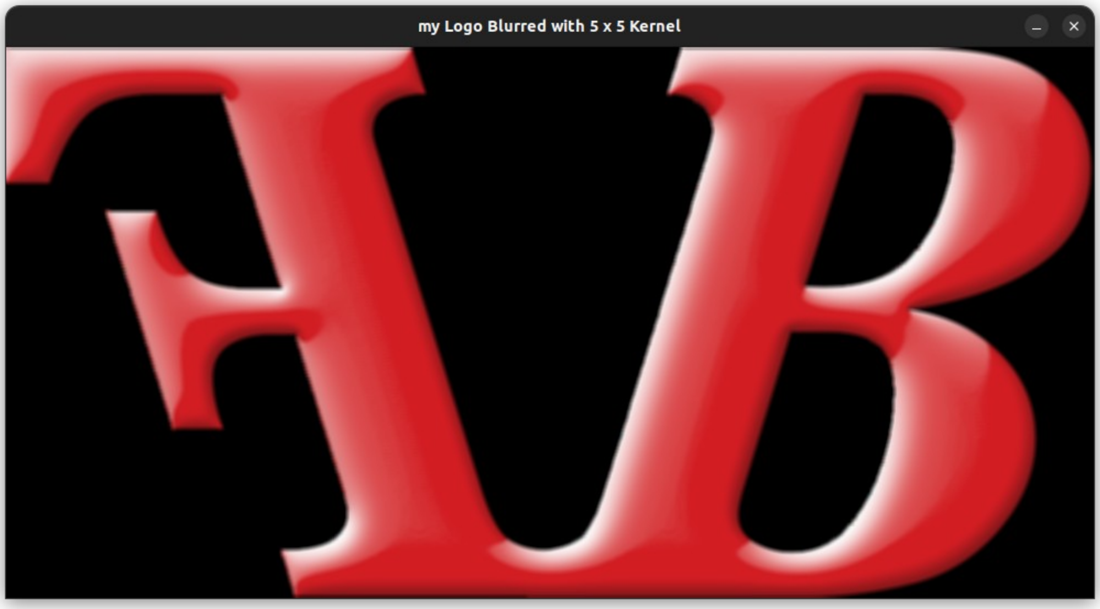
```cpp
//my_logo_image.cpp
#include <opencv2/opencv.hpp>
#include <iostream>

int main(int argc, char** argv)
{
    // Read the image file
    cv::Mat image = cv::imread("logo.png");

    // Check for failure
    if (image.empty())
    {
        std::cout << "Could not open or find the image" << std::endl;
        std::cin.get(); //wait for any key press
        return -1;
    }

    //Blur the image with 3x3 kernel
    cv::Mat image_blurred_with_3x3_kernel;
    cv::blur(image, image_blurred_with_3x3_kernel, cv::Size(3, 3));

    //Blur the image with 5x5 kernel
    cv::Mat image_blurred_with_5x5_kernel;
    cv::blur(image, image_blurred_with_5x5_kernel, cv::Size(5, 5));

    //Define names of the windows
    cv::String window_name = "my Logo";
    cv::String window_name_blurred_with_3x3_kernel = "my Logo Blurred with 3 x 3 Kernel";
    cv::String window_name_blurred_with_5x5_kernel = "my Logo Blurred with 5 x 5 Kernel";

    // Create windows with above names
    cv::namedWindow(window_name);
    cv::namedWindow(window_name_blurred_with_3x3_kernel);
    cv::namedWindow(window_name_blurred_with_5x5_kernel);

    // Show our images inside the created windows.
    cv::imshow(window_name, image);
    cv::imshow(window_name_blurred_with_3x3_kernel, image_blurred_with_3x3_kernel);
    cv::imshow(window_name_blurred_with_5x5_kernel, image_blurred_with_5x5_kernel);

    cv::waitKey(0); // Wait for any keystroke in the window
    cv::destroyAllWindows(); //destroy all opened windows

    return 0;
}
```

Ubuntu 22.04 LTS intel5 4-cores processor

```
$ cd opencv/my_logo_image/
$ ls
CMakeLists.txt  CMakeLists.txt.user  logo.png  my_logo_image.cpp
$ g++ -ggdb my_logo_image.cpp -o my_logo_image `pkg-config --cflags --libs opencv4`
$ ls
CMakeLists.txt  CMakeLists.txt.user  logo.png  my_logo_image  my_logo_image.cpp
$ ./my_logo_image
```







Francesco Bonfissuto Munich, August 02.2022