Task 1

**Problem statement**: Implement box filtering and Light Edge Detection in a standard image using CUDA and C++ and measure the speedup that can be obtained by using GPU over CPU for image processing

Solution:

Box filter:

It is an averaging filter which is a matrix that is made fully of 1s. After applying the convolution, the output obtained is divided by the total number of elements in the kernel. I have used stencil operations to apply this kernel.

Light edge detection (Total Variation Filter):

This is a 9X8 matrix with 5th column full of 1s and the main diagonal elements (excluding the element (4, 4)) full of -1s.

The remaining elements are 0. Since the sum of all elements of this kernel is 1, its normalization factor is also 1.

Also since it is a rectangular filter, I have applied it row by row.

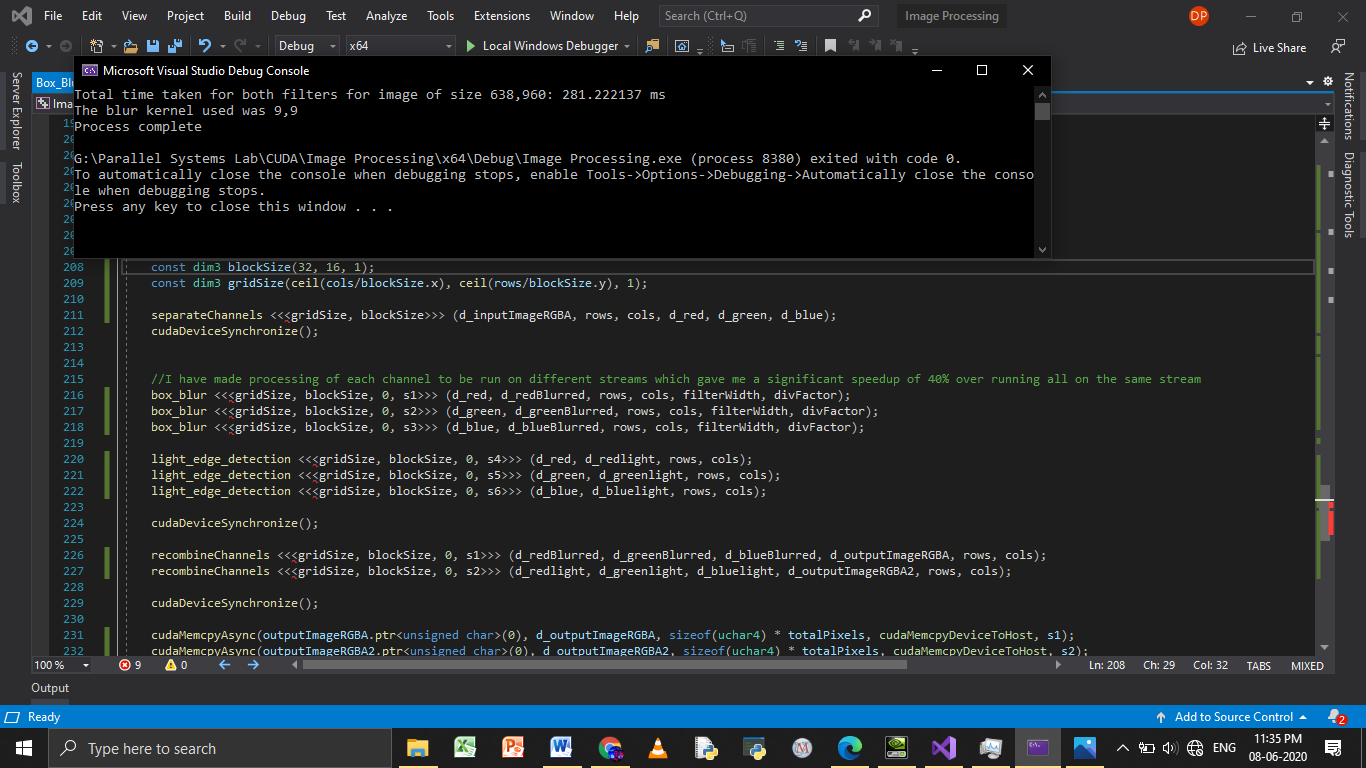
I have also implemented multiple streams so that multiple kernels can be executed simultaneously.

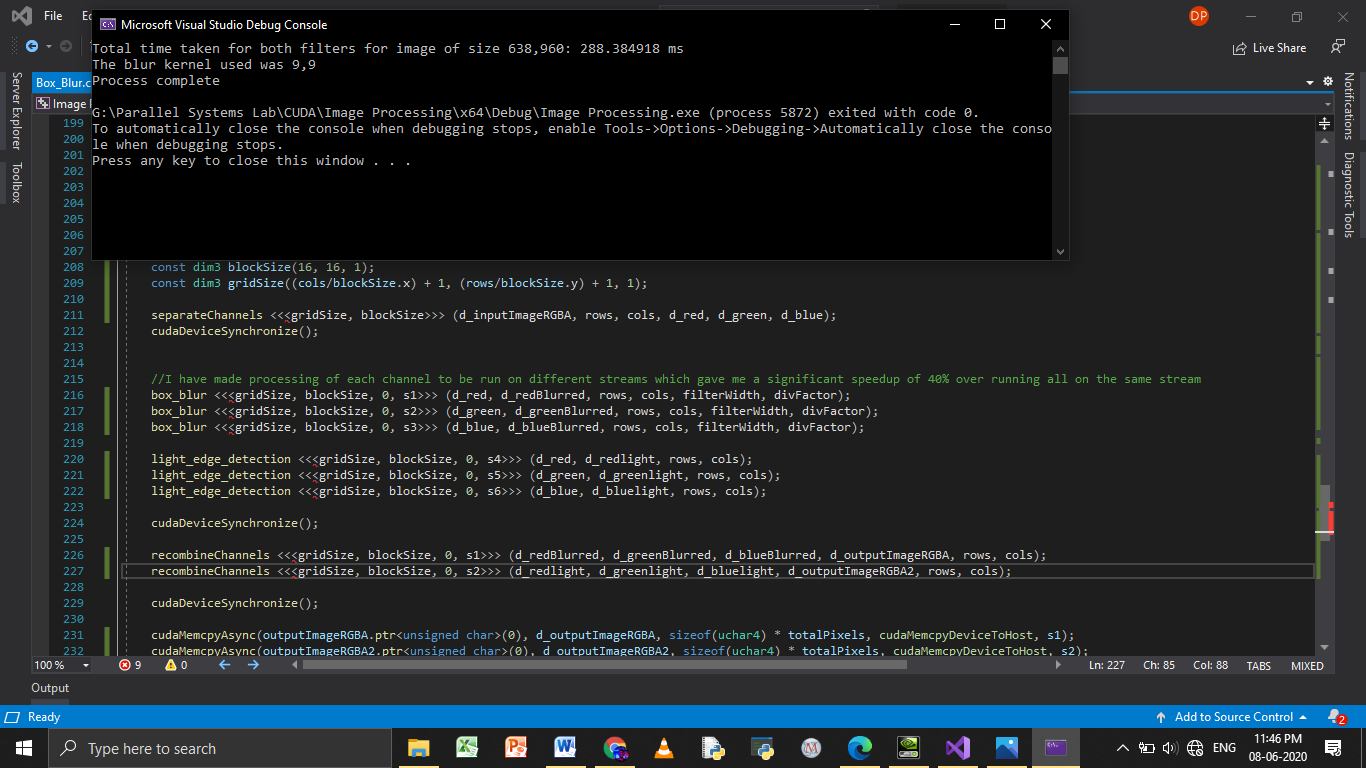
For the GPU part, I have used 4 kernels in total for the following tasks:

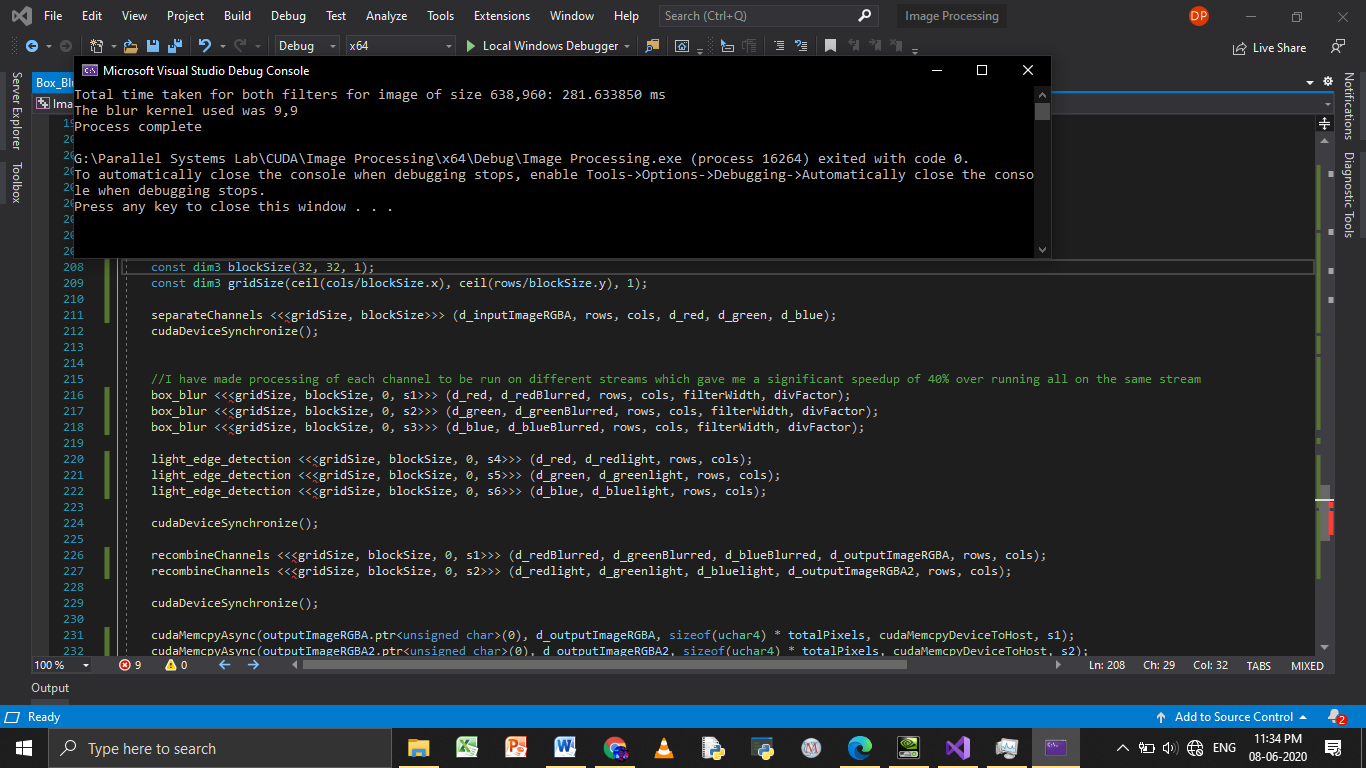
1. Splitting the image into its different channels. (Map operation)
2. To apply box blur kernel on a single channel. (Stencil operation)
3. To apply a total variation filter on a single channel. (Stencil operation)
4. To combine the output channels into a single image. (Map operation)

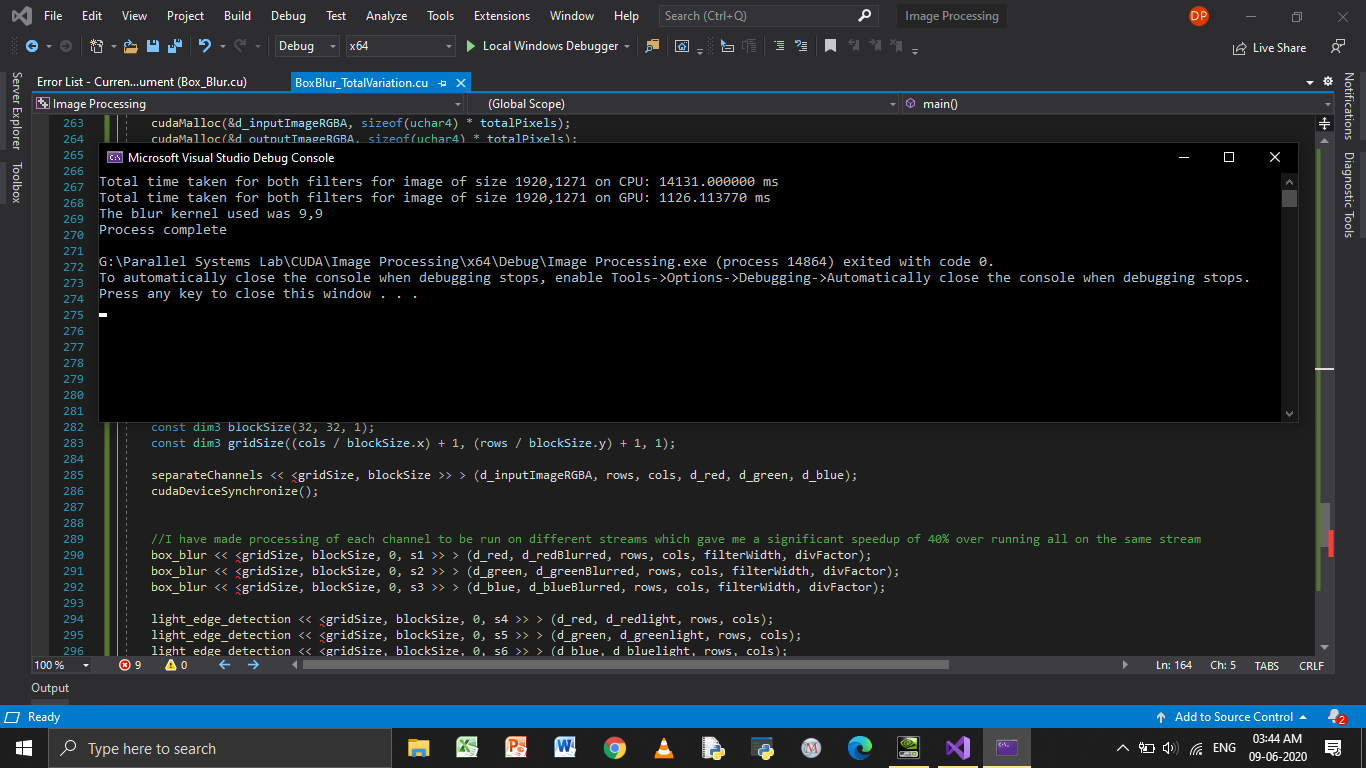
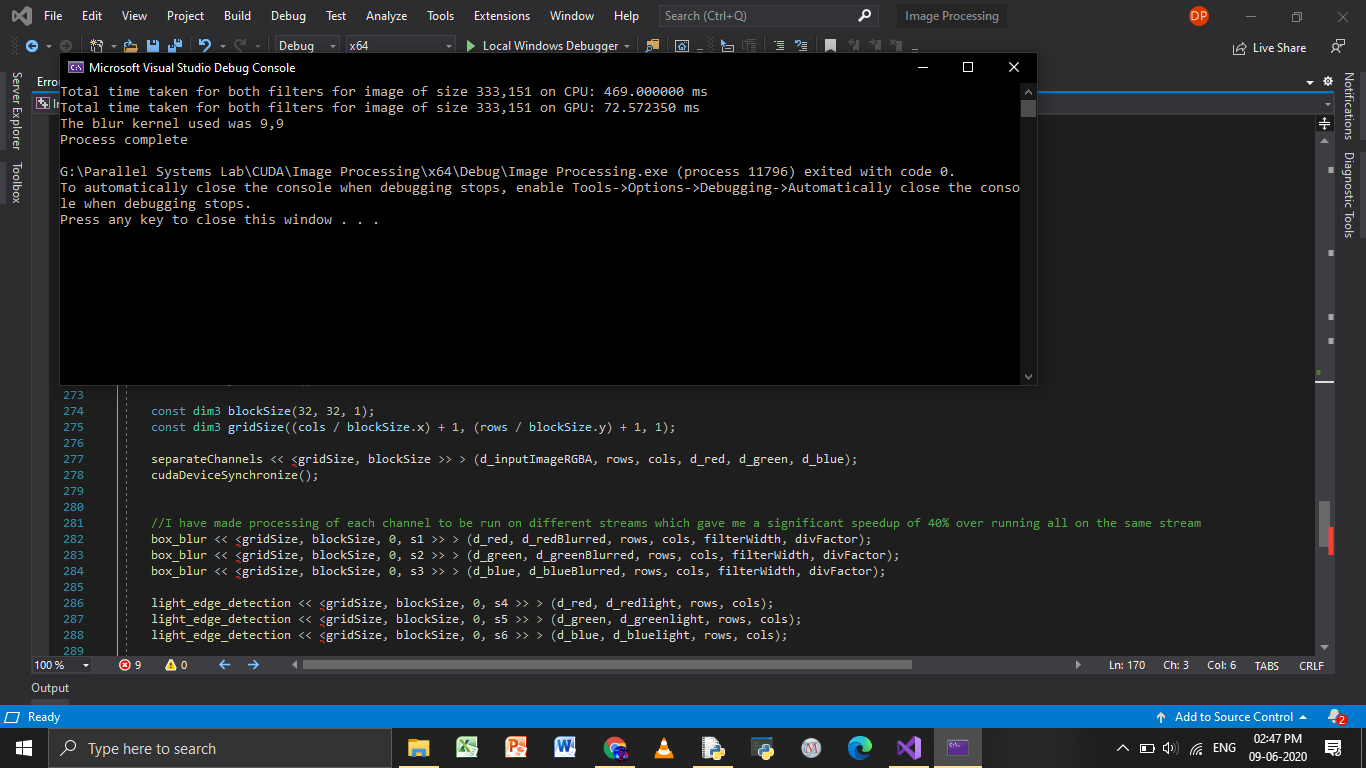
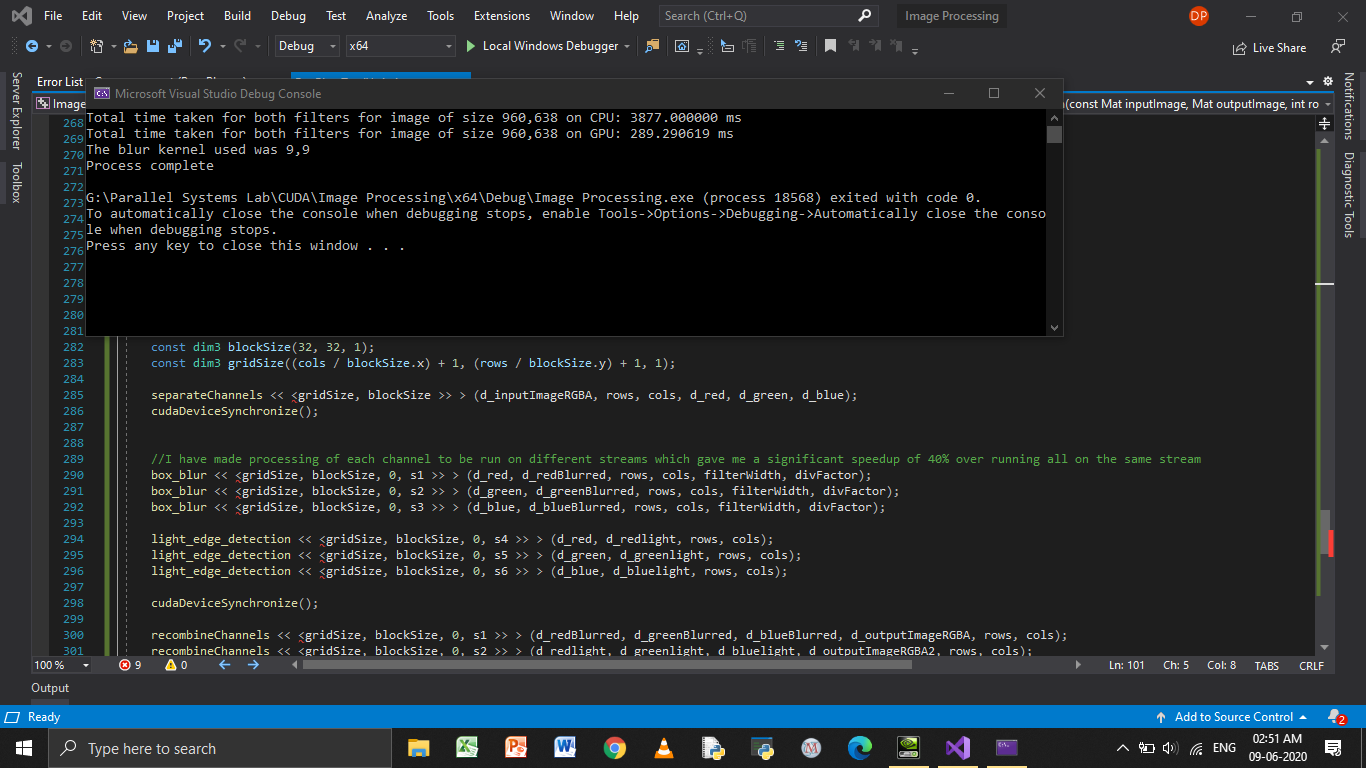
For the serial code, I reused the same logic as that of the above kernels.

Trying multiple Grid and Block sizes:



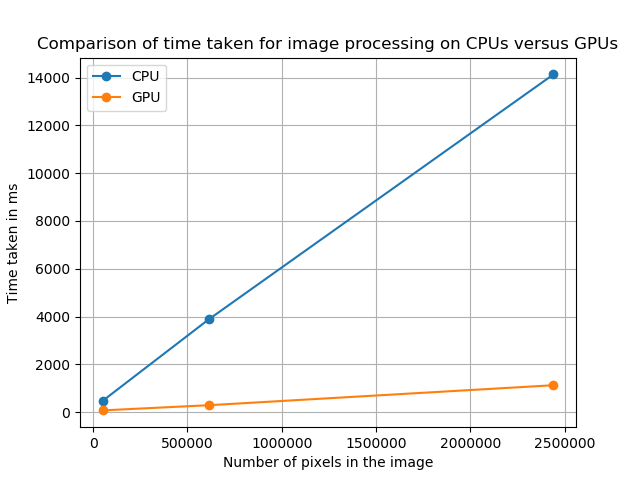




**Comparison between CPU and GPU for applying these filters:**

As seen from the above image, applying these filters on the GPU takes about 13.5 times less time compared to serial execution.

**Graph:**



**Software requirements:**

* Nvidia Toolkit
* OpenCV C++

**Hardware requirements:**

The GPU I have used here has a limit of 1024 threads per block and hence a compatible GPU needs to be used to run this code without any alterations. However for older GPUs which have a limit of 512 threads per block, the dimensions of the blocksize needs to be altered before running the code.

**What I learnt from this:**

I got hands-on experience of applying what I learnt in the Udacity course to solve a real world problem. I also explored a little of OpenCV and C++ and was introduced to vectors.

-Dhruval PB