

1. Implementation Result.

| | img_128 | img_256 | img_512 | img_1024 |
|------------------------------|---------|---------|---------|----------|
| Average (About 5 times) | 2.23 | 2.20 | 2.18 | 2.38 |
| Best Case (About 5 times) | 2.33 | 2.26 | 2.26 | 2.54 |

My implementation showed the best result in img_1024, but I there was no tendency.

2. Optimization approaches

1) Succeeded approach

At first, I tried to reduce the multiplication and unnecessary function call. So I combined two given function, `convolution()` and `filter_optimized()`, in one function. During the process of merging the functions, the return parameter was found to be unnecessary, thus eliminating the need for initialization using the `memset` function. Additionally, it became unnecessary to allocate dynamic memory unnecessarily.

Next, efforts were made to reduce multiplication operations. To achieve this, all indices were calculated outside the forloop, and only additional addition operations were performed within the loop. Following this, loop unrolling was implemented. This unrolling allowed the indices for the filter variable to be extracted as integers, thereby reducing the number of operations further.

2) Failed approach

After loop unrolling, the number of repetitive ifstatements increased significantly. I attempted to handle this by converting them into a switchstatement to move the internal multiplication operations outside, but this approach failed since the process was sequential rather than parallel. Consequently, I tried to reduce the operations by modifying the conditions in the ifstatements, but unfortunately, I was unable to achieve that.