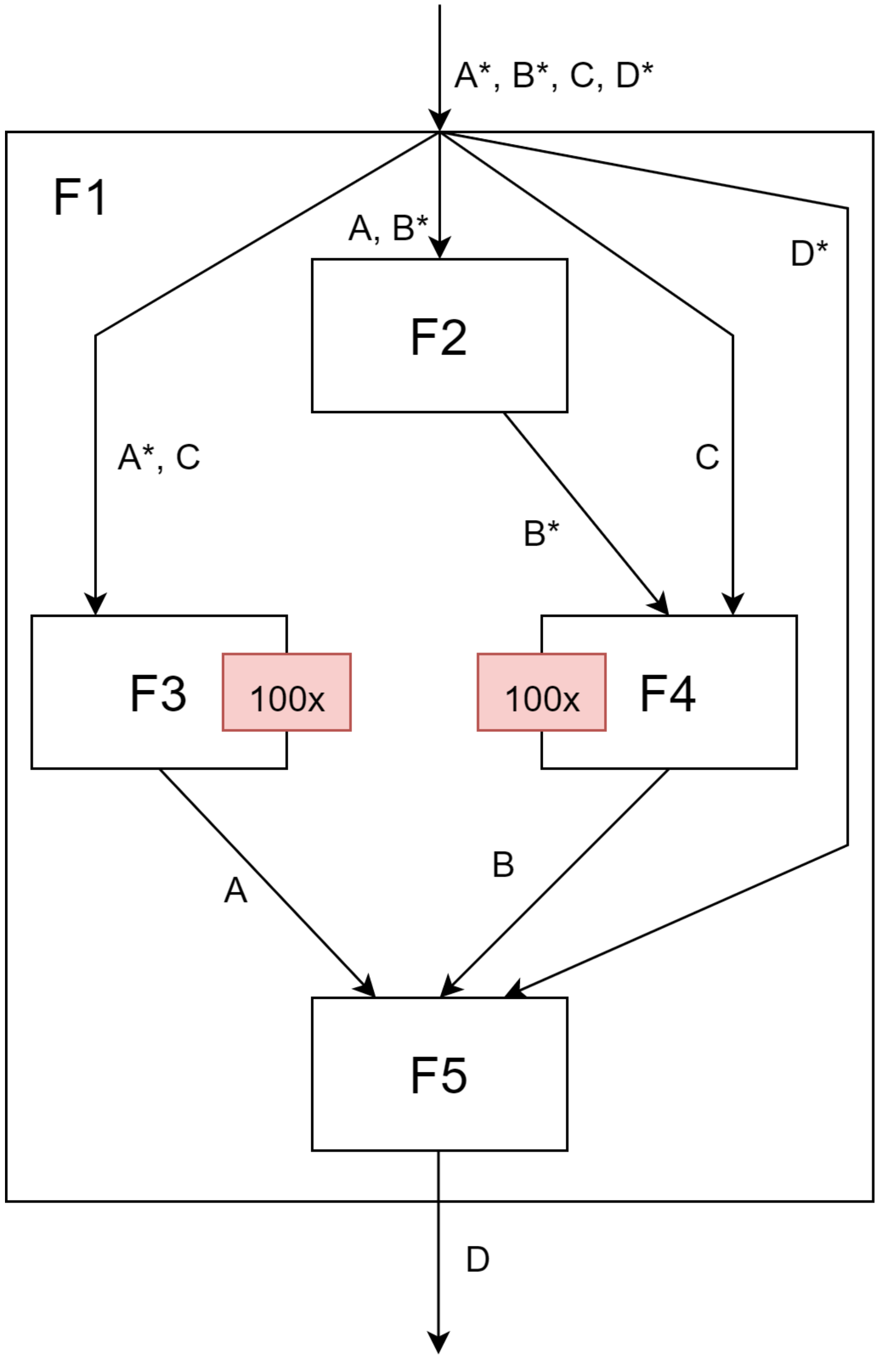
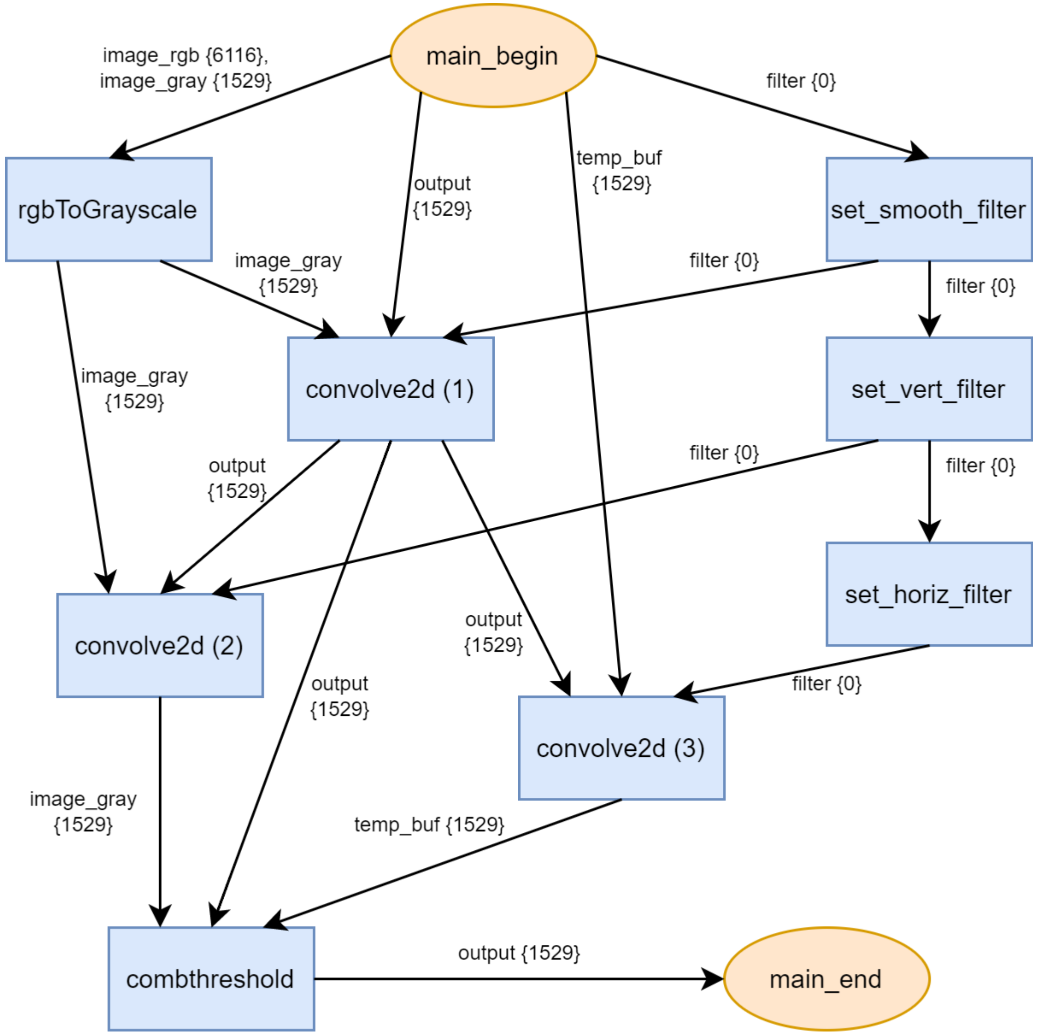
Task Graph

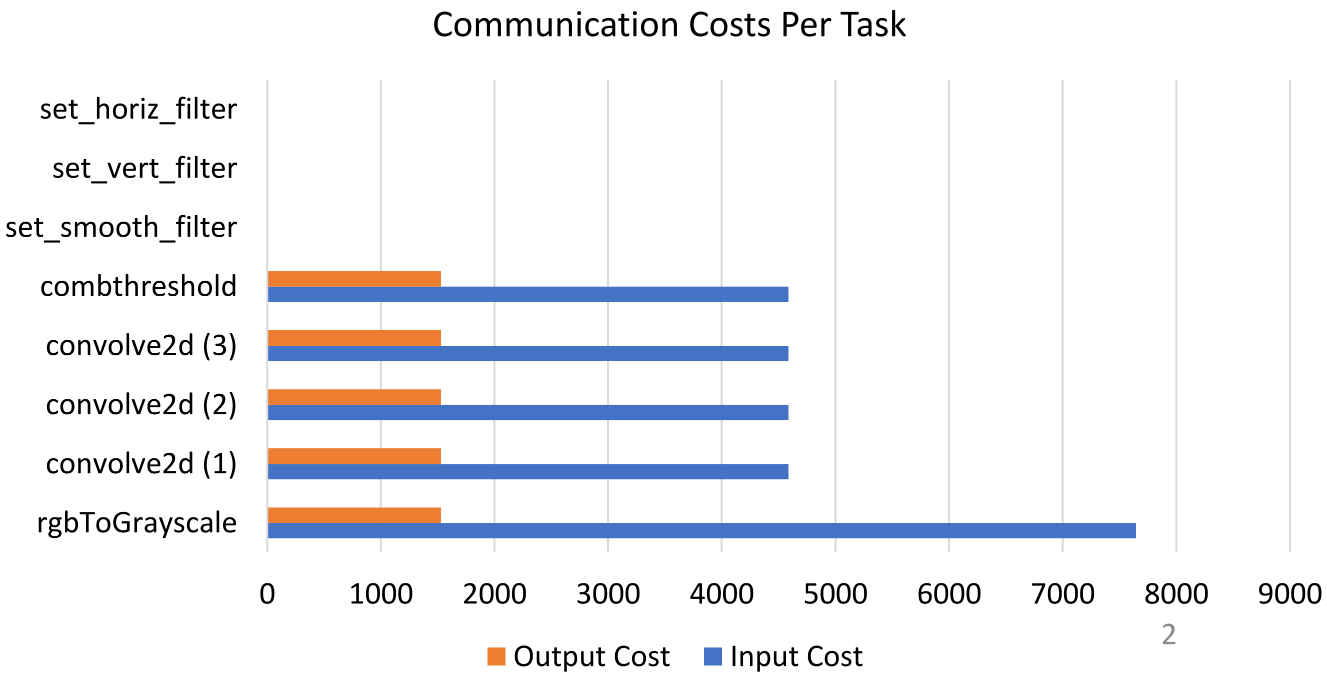
```
void f1(int A[100], int B[100], int C[100], int D[100])
{
    // modifies B
    f2(A, B);

    for (int i = 0; i < 100; i++)
    {
        // modifies A
        f3(A, C[i]);

        // modifies B
        f4(B, C[i]);
    }

    // modifies D
    f5(A, B, D);
}
```



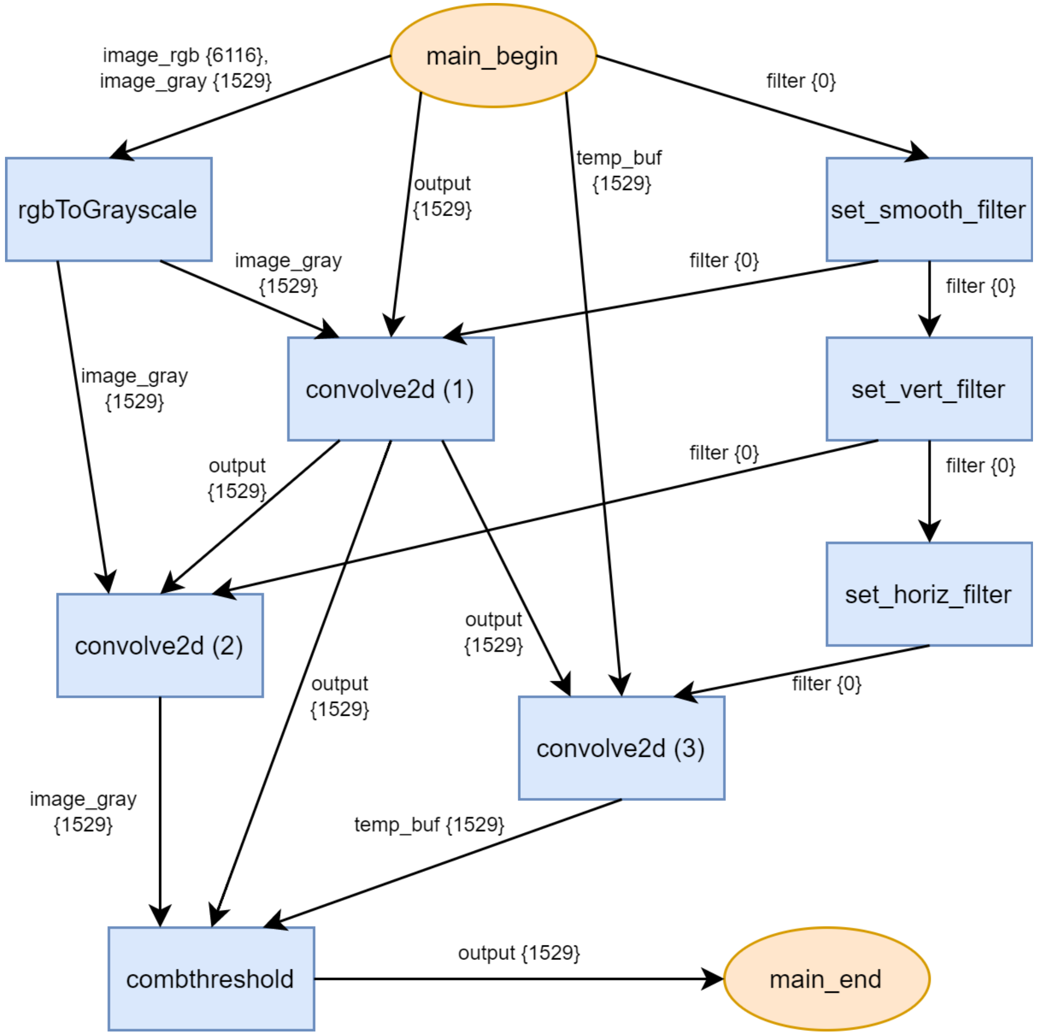
General statistics



- 17 communication tasks
- 8 regular tasks
- 2 special tasks
- Calls to math.h functions: 0
- Calls to external functions (e.g., printf): 0



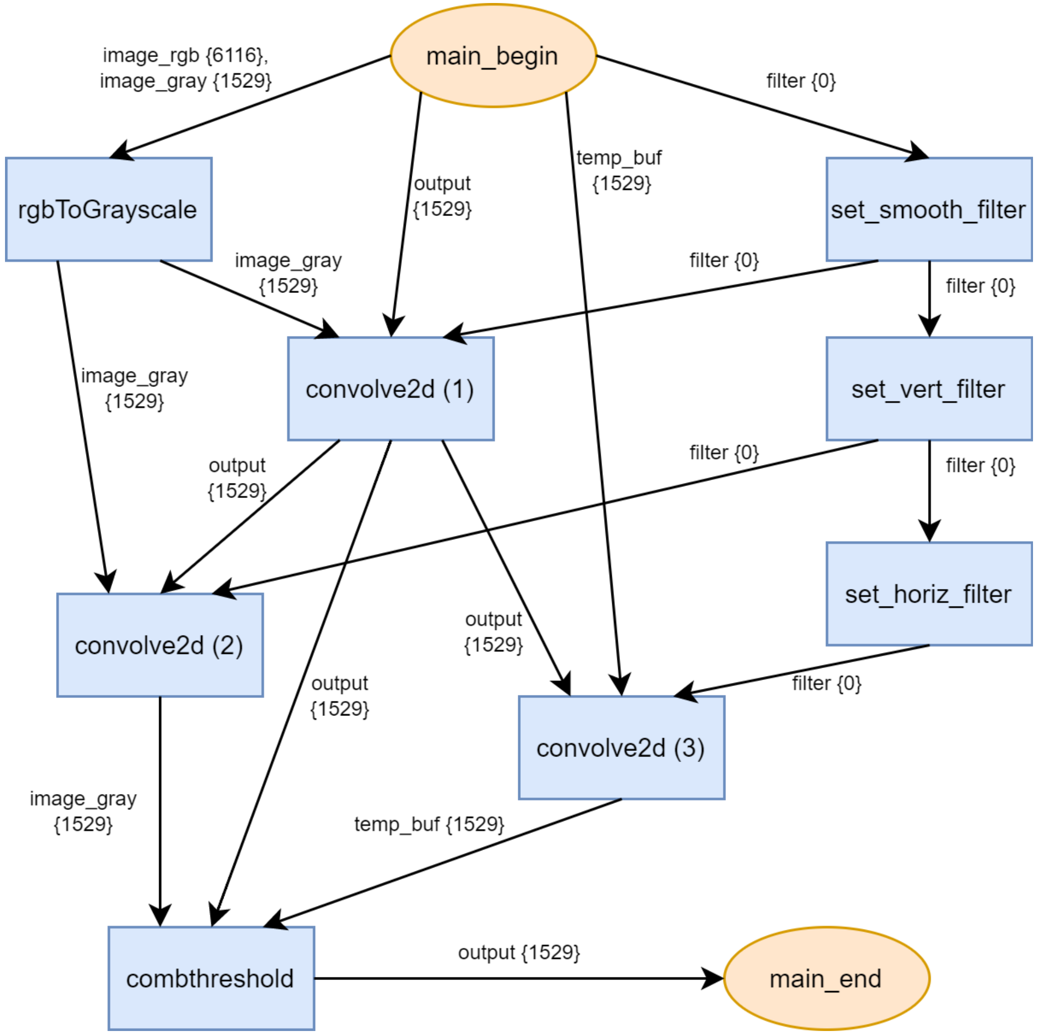
Size of largest clusters (in tasks)



A cluster consists of tasks possibly to implement in hardware

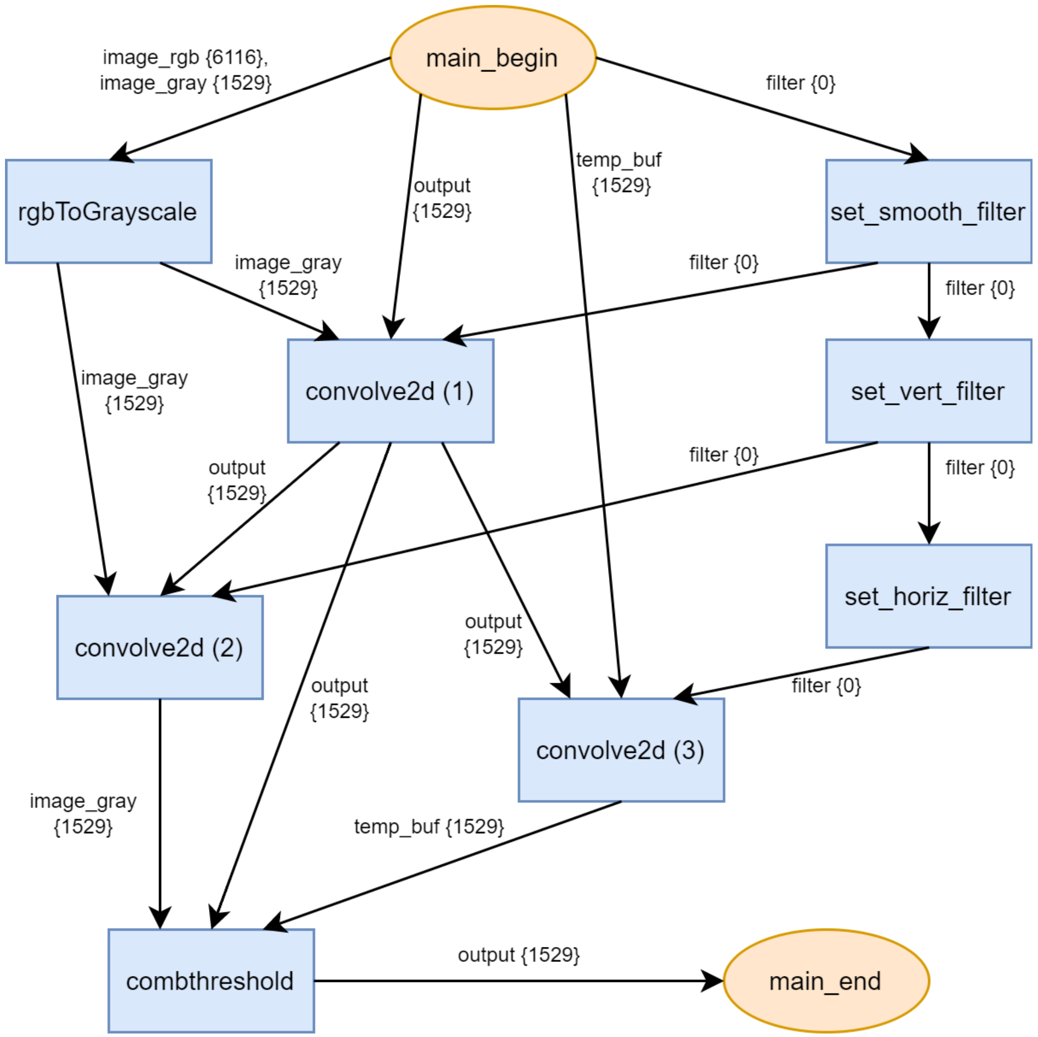
Maximum size here is 8 (all tasks can form a cluster, except for main_begin and main_end)

Array usage (read or write)



image_rgb: 2
image_gray: 3
filter: 7
output: 6
temp_buf: 3

Distance to array production



rgbToGrayscale:
- image_rgb: 1
- image_gray: 1

convolve2d (1):
- image_gray: 2
- output: 1
- filter: 2

convolve2d (2):

- image_gray: 2
- output: 2
- filter: 3

convolve2d (3):
- output: 2
- temp_buf: 1
- filter: 4

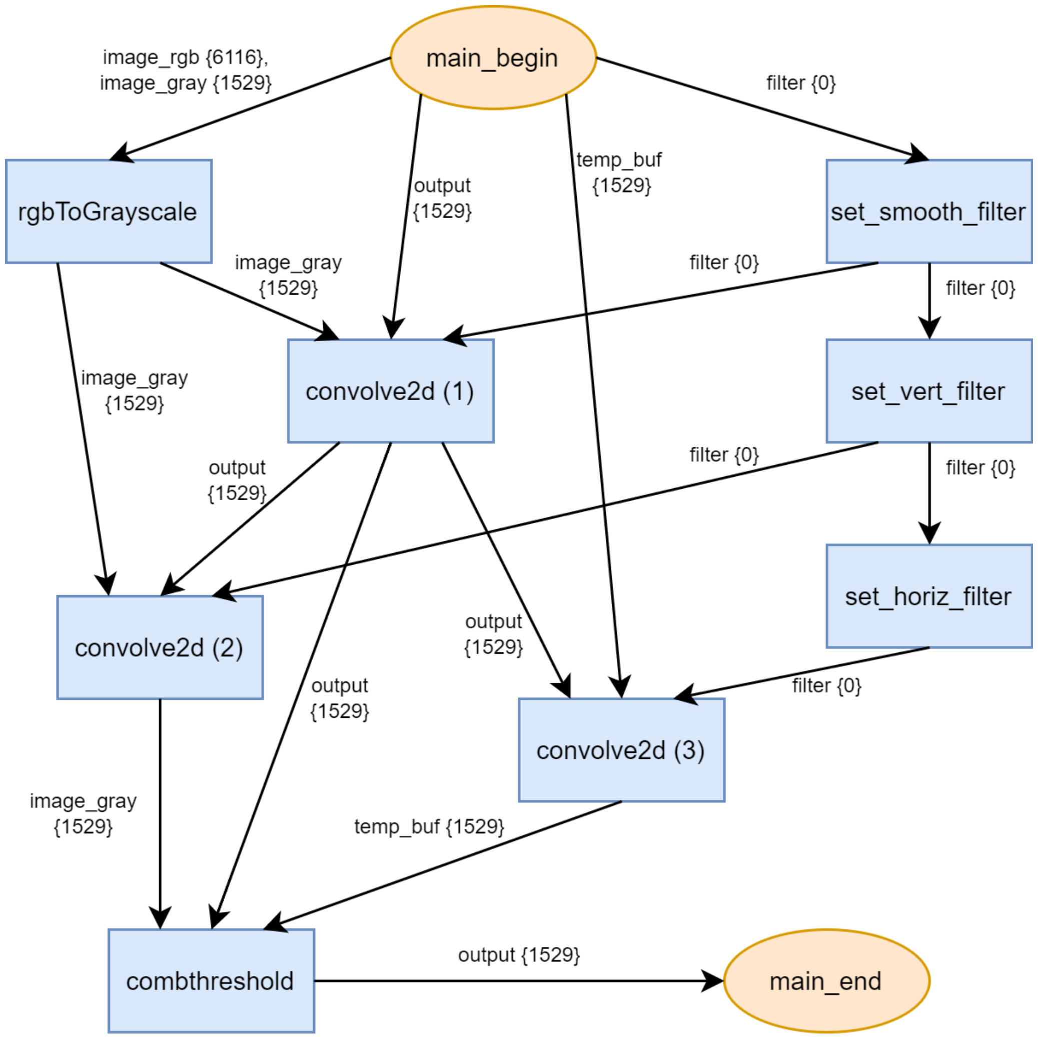
combthreshold:
- image_gray: 3
- output: 2
- temp_buf: 2

set_smooth_filter:
- filter: 1

set_vert_filter:
- filter: 2

set_horiz_filter:
- filter: 3

Distance to array consumption



rgbToGrayscale:
- image_rgb: 0
- image_gray: 2

convolve2d (1):
- image_gray: 0
- output: 2
- filter: 0

convolve2d (2):
- image_gray: 1
- output: 0
- filter: 0

convolve2d (3):
- output: 0
- temp_buf: 1
- filter: 0

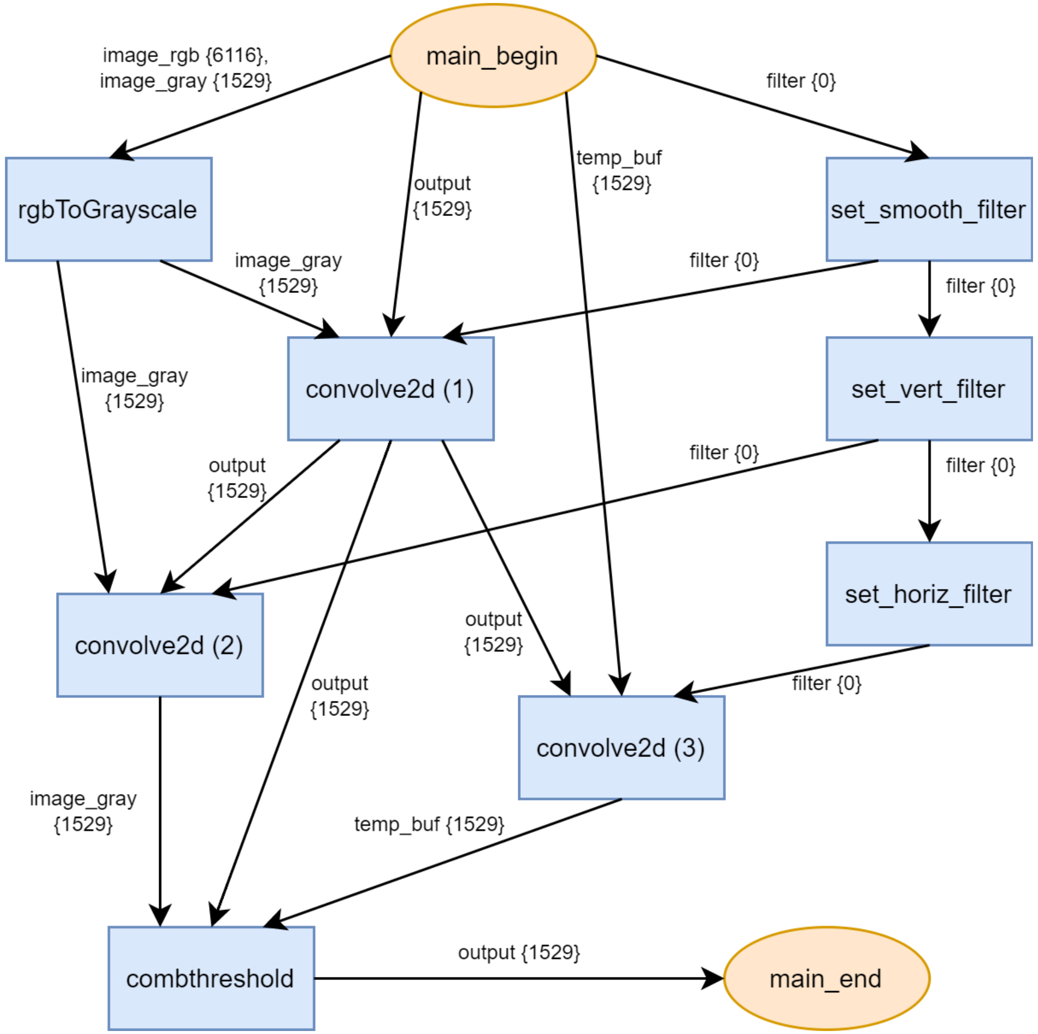
combthreshold:
- image_gray: 0
- output: 1
- temp_buf: 0

set_smooth_filter:
- filter: 3

set_vert_filter:
- filter: 2

set_horiz_filter:
- filter: 1

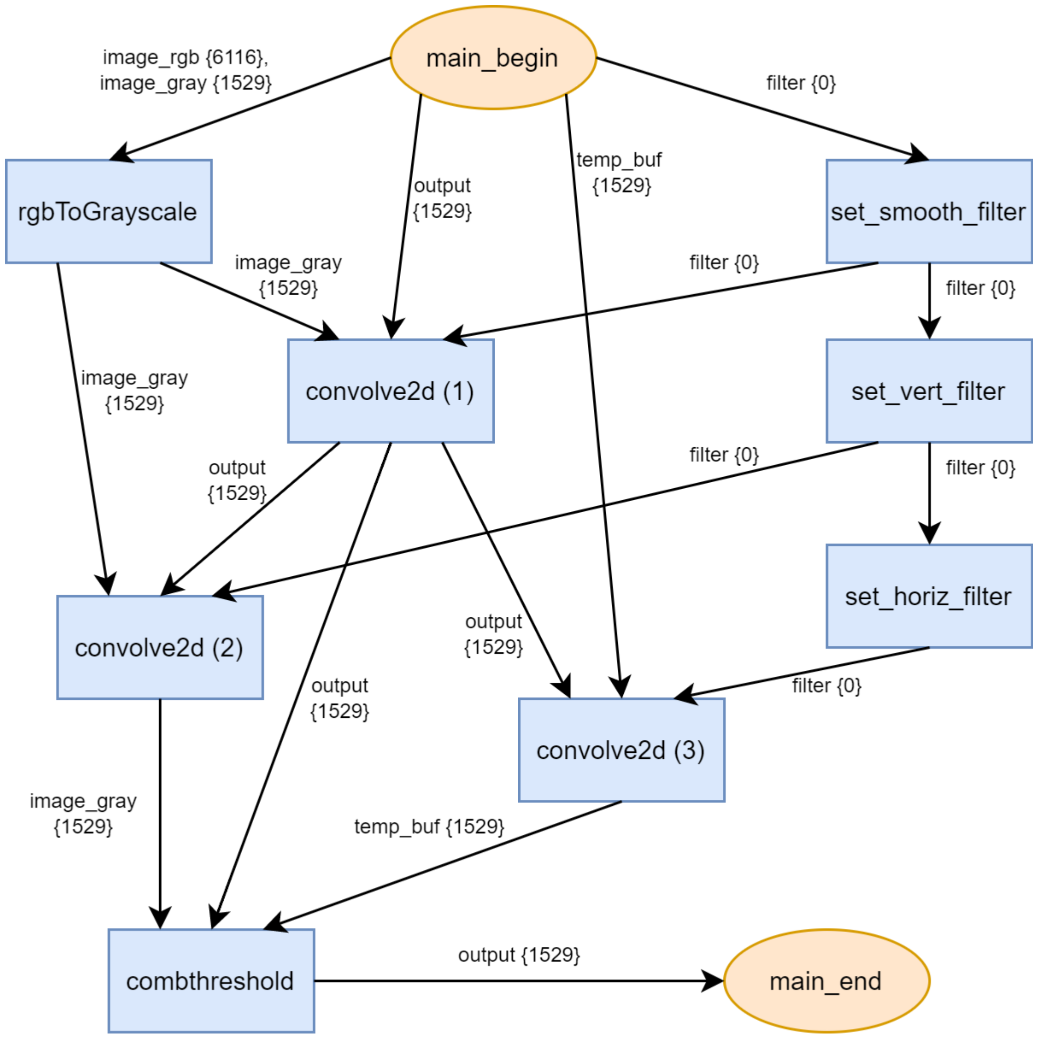
Producer-consumer relations



Number of consumer-producer relations between tasks (i.e., one task produces results in the same order as the following task consumes them)

There are none

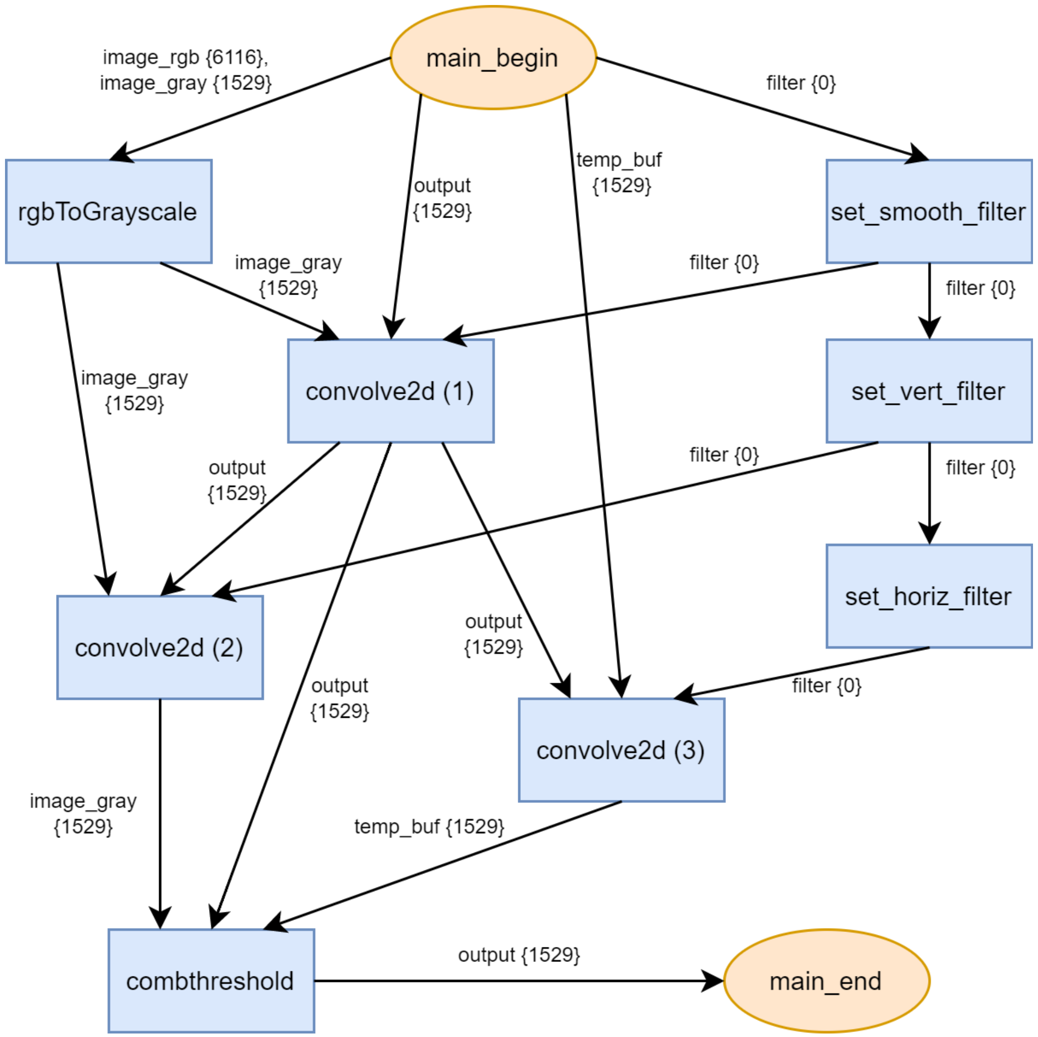
Tasks using/calling resources not supported by hardware



Number of tasks that have no available implementation (e.g., printf, system calls)

There are none (main_begin and main_end use them, but they are, by definition, SW tasks)

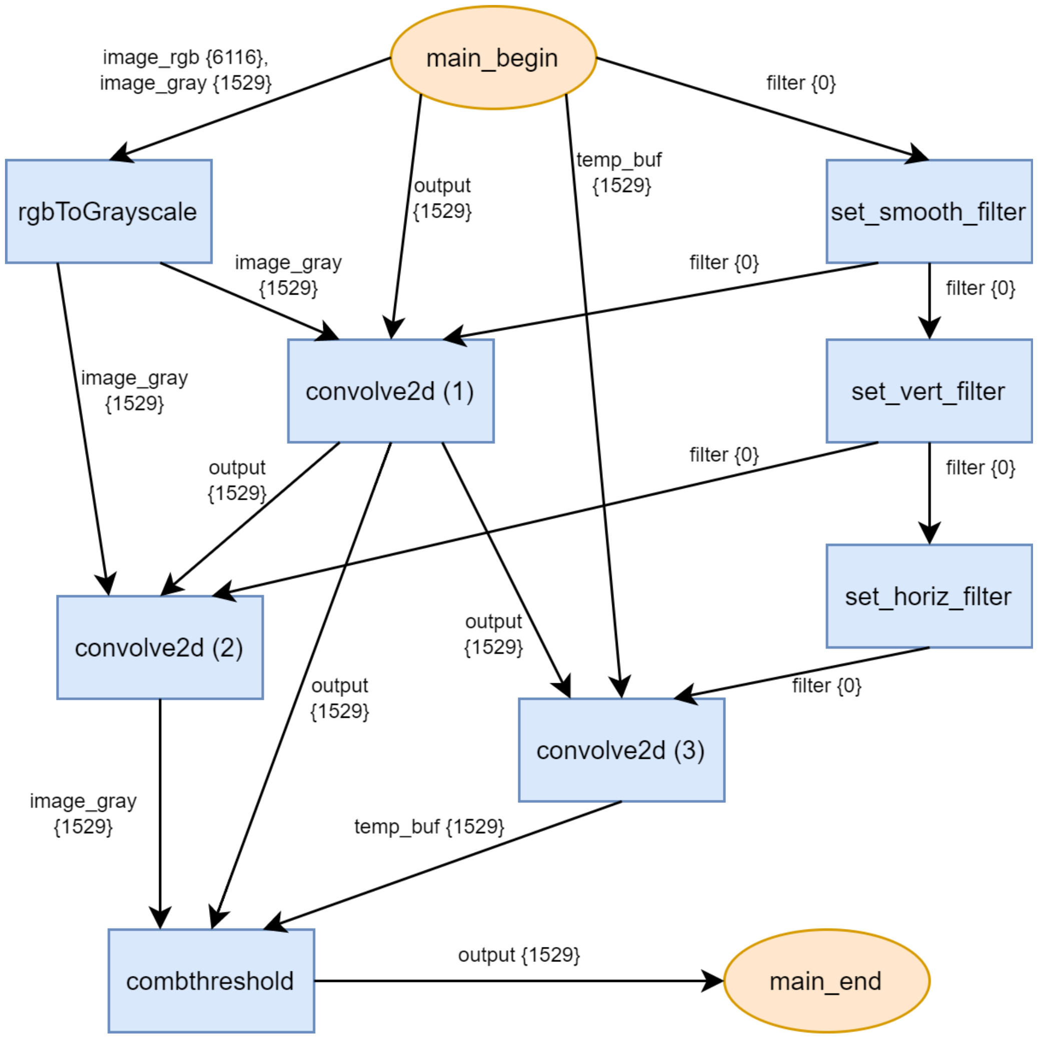
Task level parallelism



Pairs of tasks that can be executed in parallel (i.e., have no data dependencies to each other)

- convolve2d (2) and convolve2d(3)
- rgbToGrayscale and set_smooth_filter
- convolve2d (1) and set_vert_filter
- convolve2d (2) and set_horiz_filter

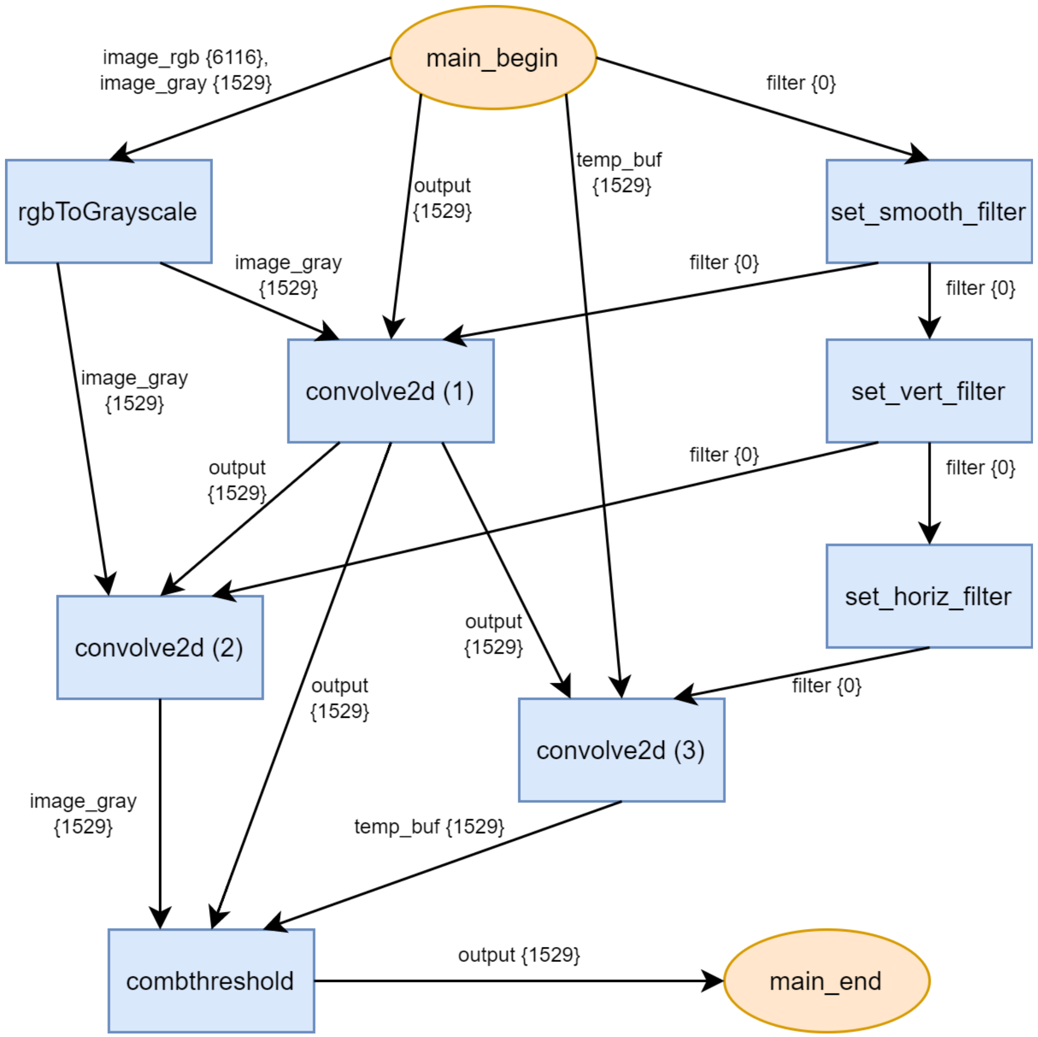
Task exit points



Number of possible exit points in a task (e.g., conditional premature return statements)

- rgbToGrayscale: 1
- convolve2d (1): 1
- convolve2d (2): 1
- convolve2d (3): 1
- combthreshold: 1
- set_smooth_filter: 1
- set_vert_filter: 1
- set_horiz_filter: 1

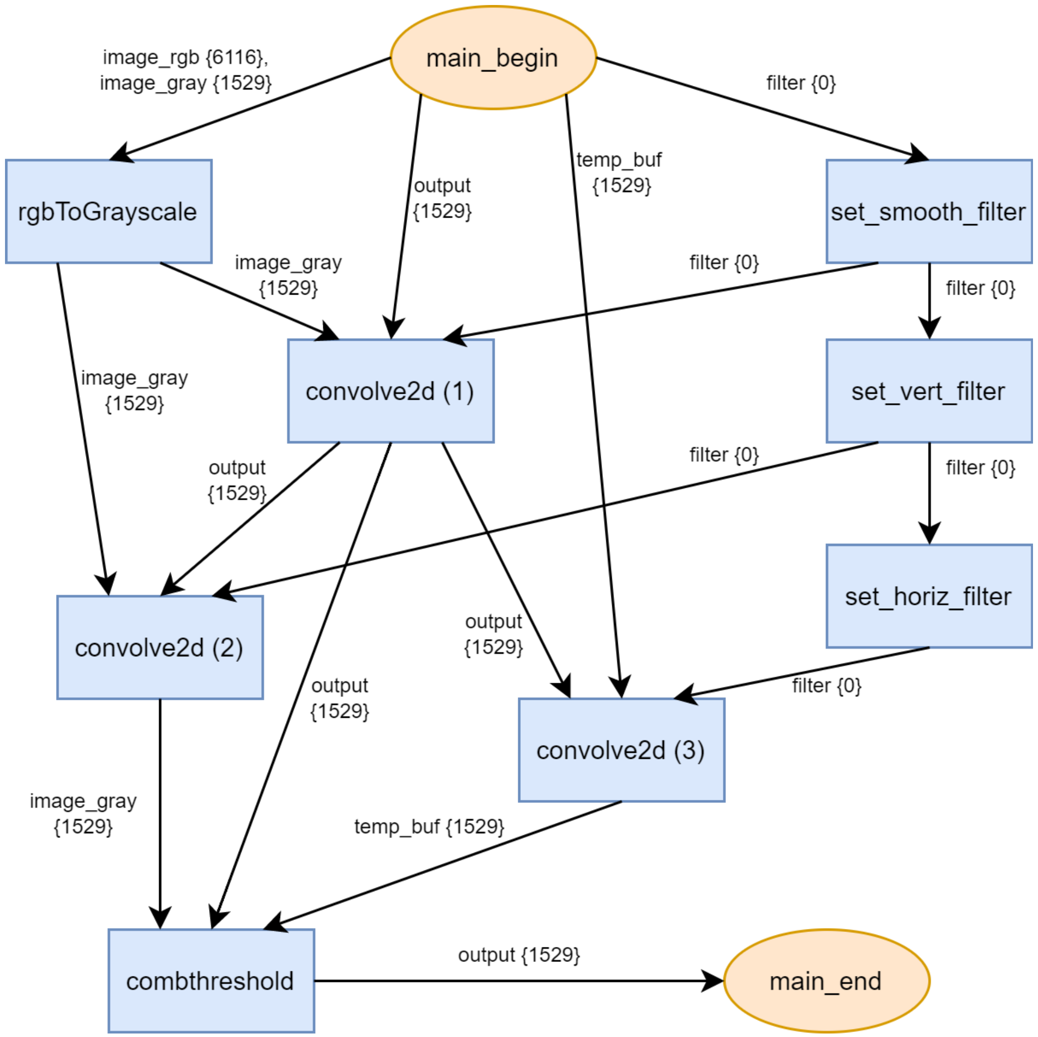
Conditional tasks



Tasks that may only execute depending on a condition

There are none.

Task Read/write ratio

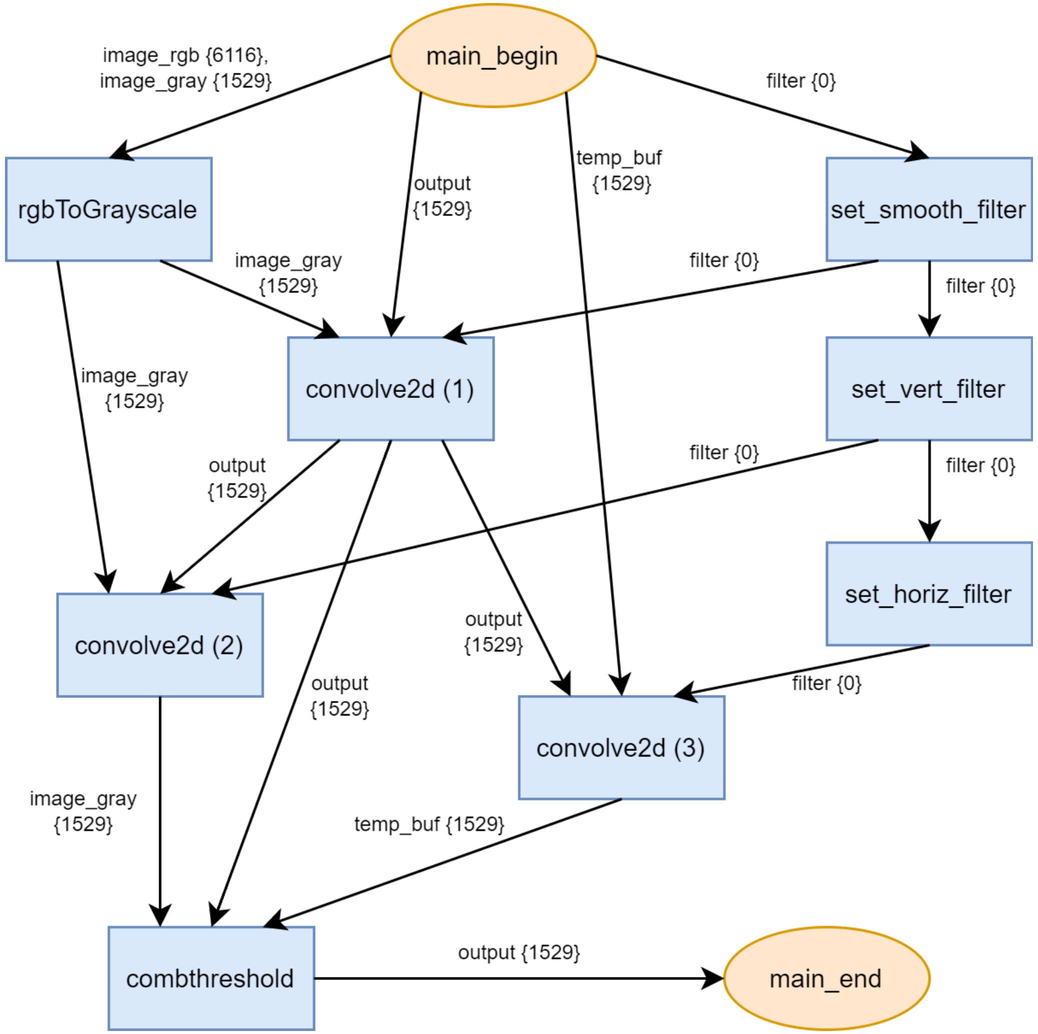


For a task, find the ratio between the data that must be communicated to the task for read purposes, and the data that the task modifies (i.e. cost of input / cost of output)

Note that some inputs may never be read, only written to

- rgbToGrayscale: 5
- convolve2d (1): 2
- convolve2d (2): 2
- convolve2d (3): 2
- combthreshold: 3
- set_smooth_filter: 0
- set_vert_filter: 0
- set_horiz_filter: 0

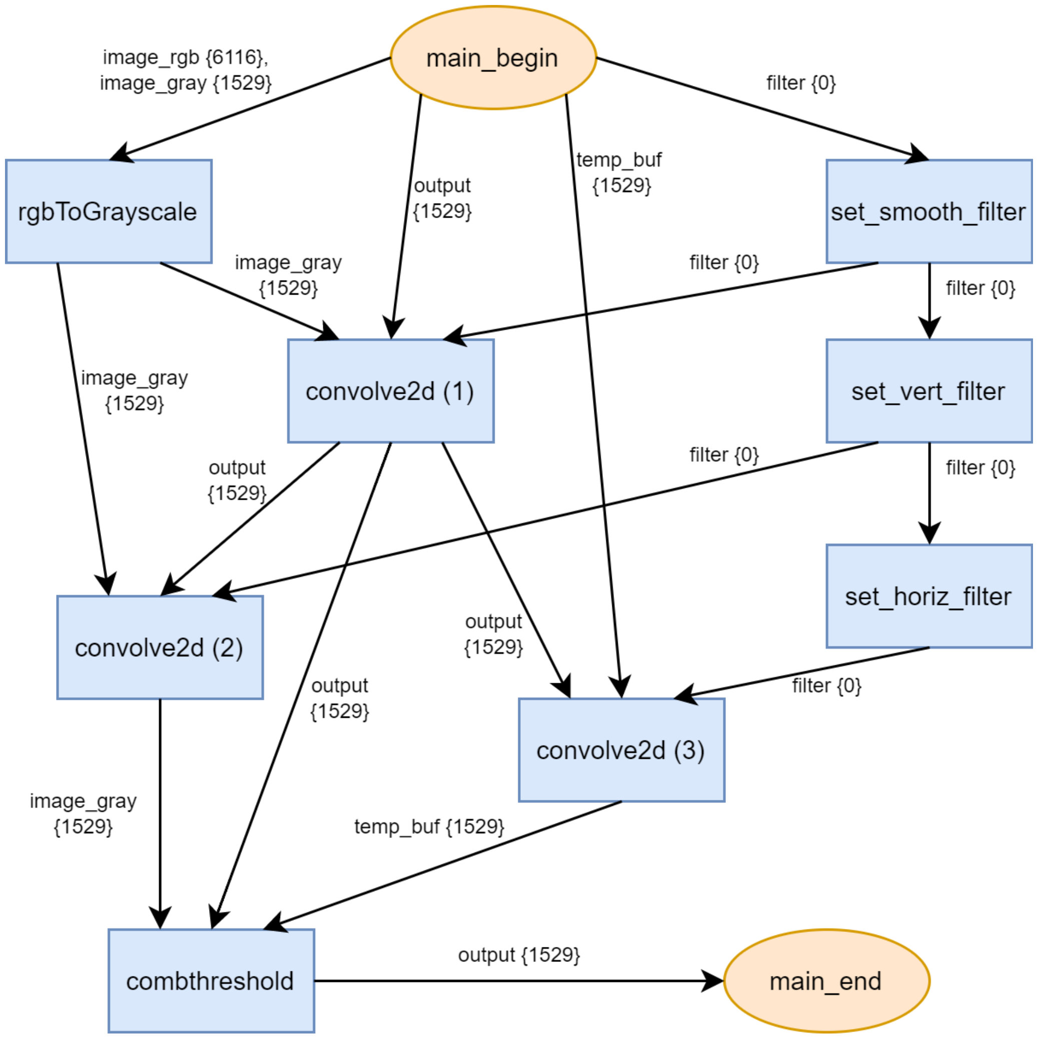
Cost of minimum cut



Cost of the data communicated in the graph's minimum cut

4587 (1529 * 3)

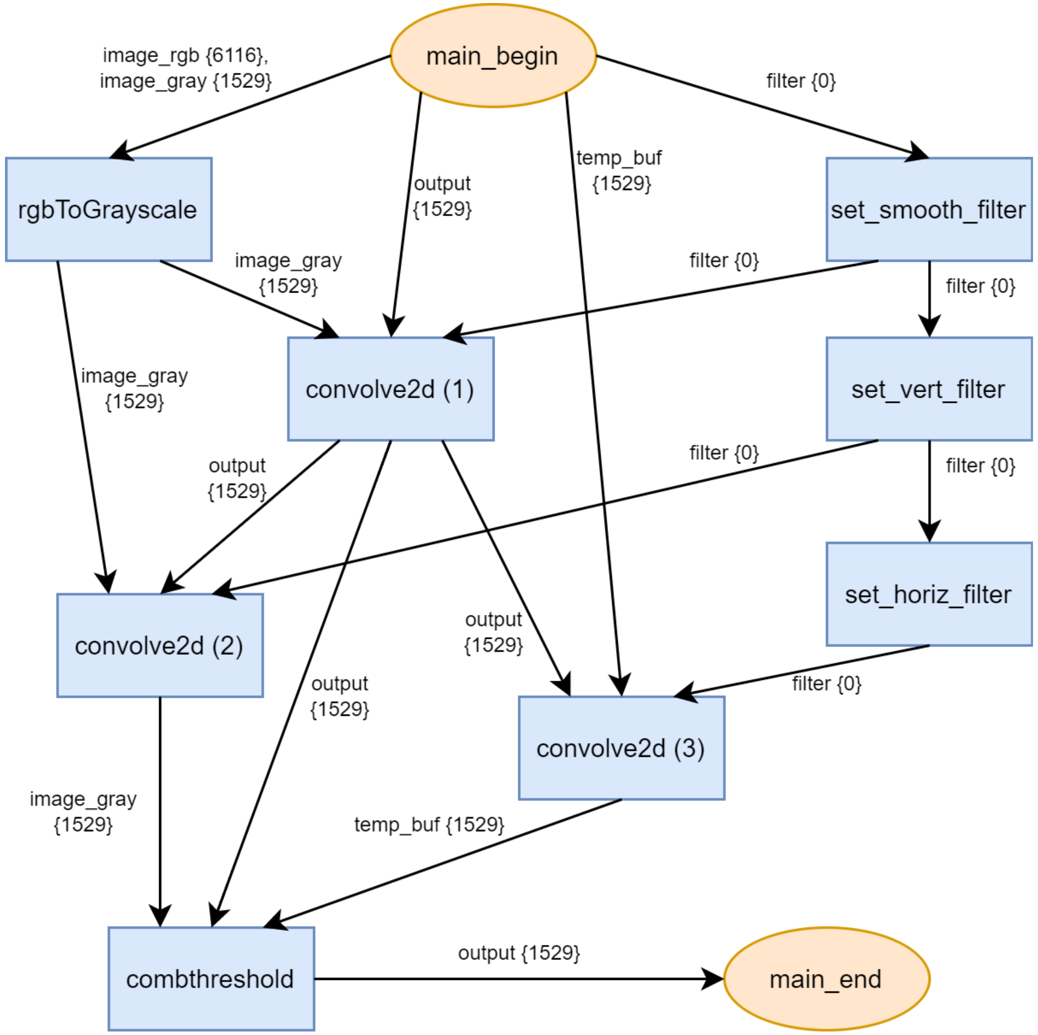
Number of tasks per hierarchical level



1st level: 10

(main_end and main_begin could also be seen as its own hierarchical level)

Critical path length



Measured in terms of tasks

6 tasks:

- main_begin -> rgbToGrayscale-> convolve2d (1) -> convolve2d (2) -> combthreshold -> main_end

or

- main_begin -> rgbToGrayscale-> convolve2d (1) -> convolve2d (3) -> combthreshold -> main_end