# Kernel (1)

First, we have Tried to use the 3Darray to CUDA, but we have faced some **challenges forced us to use the linearized version again:**

1. The Kernel worked well for small image, but in case of large images the compilation passed but when running the .out file nothing is executed [It Fails in step of Memory allocation]

We found out that this static allocation is done in the stack which is small that can’t be used to hold all this data.

**Unsigned char image [HEIGHT][WIDTH] [CHANNELS]**

1. We need the Image Dimensions to be available at compilation which isn’t partial instead we need to take the image dimensions form the first image read from the input directory.

For Constant Memory Allocation we first use 2D Static Indexing but still same problem to use different filters we need to change the filter dim in the code and recompile which is time consuming, so we used Static allocation with max limit 400.

A screenshot of a computer screen

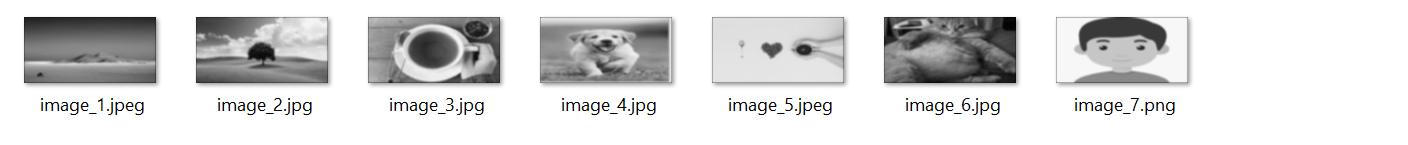
Description automatically generated

## Padding:

Without Padding we just Added Boundary Conditions for out-of-range input. The output was the same size as input because initially we allocated with this, but we got the outer frame to be zero [Default Value]

A black screen with colorful text

Description automatically generated



We added Ghost Cells

A screen shot of a computer program

Description automatically generated

A close up of a cup

Description automatically generated

No Black Border ☺☺