

# PDC Assignment 3 Report

## Setup Environment:

Used Docker implementation with libjpeg library for image reading.

## OpenCL Implementation:

I implemented it through 2 ways;

1. Reading All images at once
2. Reading images one by one
3. Serially Converting without OpenCL (For Comparison)

### Reading Images At Once:

In this implementation I used about 25 images at once due to ram limitations, The images are loaded into a single array with offset of max width and height so the kernel knows where the next image is. Here each kernel works on different a image parallely and converts it into grayscale. This was the fastest method I tested in comparison to Implementation 2 and Serial Implementation.

### Reading Images One by One:

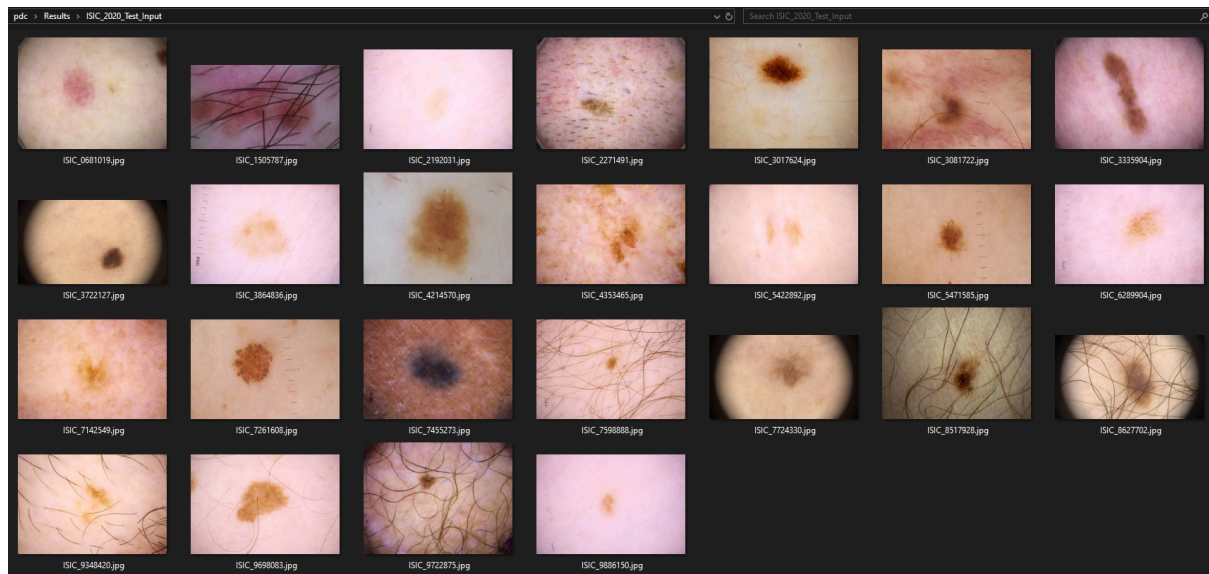
In this implementation you can take all the images if you want (35 mins processing!), It works by loading an images, dividing its pixels across kernels to convert into grayscale in parallel, reading the output buffer to reconstruct the image and save it. This method is only a little bit faster than serial execution and did not seem intuitive at all.

## Comparison Table

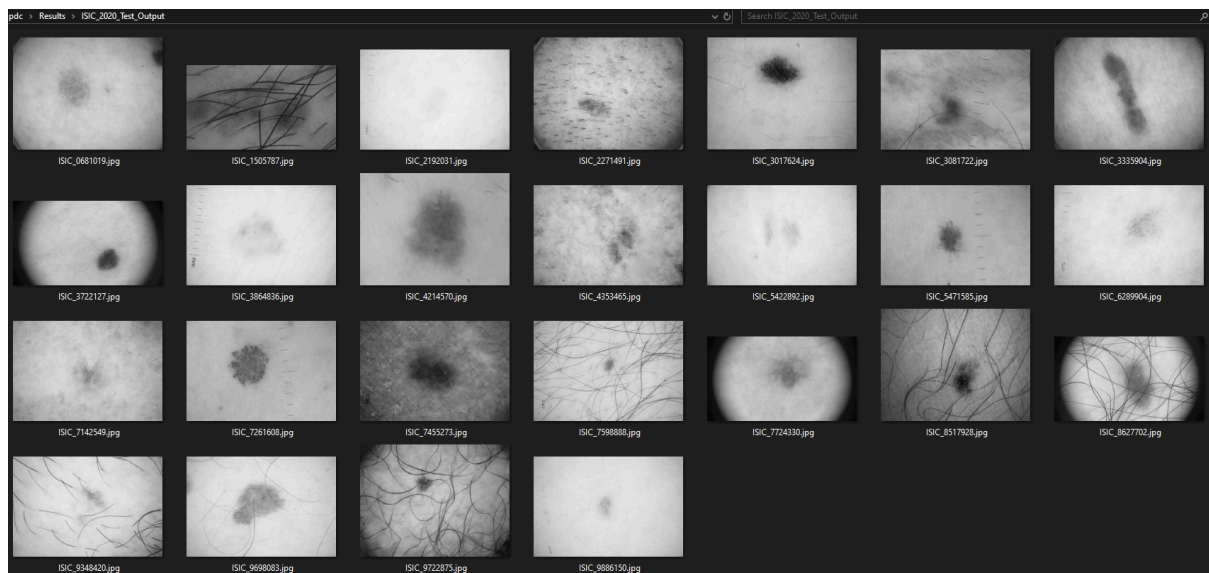
*	Sample Size	Serial	Reading images one by one	Reading All images at once
Time Taken:	5	0.501904	0.504704	0.362495
Time Taken:	15	1.251908	1.211019	0.889743
Time Taken:	25	2.305376	2.209549	1.492032

## Output Screenshots:

Input:



Output:



## Compile Command:

```
gcc -o h host.c jpeg_loader.c jpeg_saver.c -lOpenCL -ljpeg
```

(I created libraries for reading and writing images to clean up the mess and make code readable)