Name: *Guanshujie Fu*
NetID: *gf9*
Section: *ZJ1/ZJ2*

ECE 408/CS483 Milestone 2 Report

1. Show output of rai running Mini-DNN on the basic GPU convolution implementation for batch size of 1k images. This can either be a screen capture or a text copy of the running output. Please do not show the build output. (The running output should be everything including and after the line "*Loading fashion-mnist data...Done*").

✱ Running bash -c "time ./m2 1000" \\ Output will appear after run is complete.

```
Test batch size: 1000
Loading fashion-mnist data...Done
Loading model...Done
Conv-GPU==
Layer Time: 63.6462 ms
Op Time: 1.62626 ms
Conv-GPU==
Layer Time: 54.3965 ms
Op Time: 6.25192 ms

Test Accuracy: 0.886


real    0m9.667s
user    0m9.306s
sys     0m0.329s
```

2. For the basic GPU implementation, list Op Times, whole program execution time, and accuracy for batch size of 100, 1k, and 10k images.

Batch Size	Op Time 1	Op Time 2	Total Execution Time	Accuracy
100	*0.173583ms*	*0.634666ms*	*1.215s*	*0.86*
1000	*1.62626ms*	*6.25192ms*	*9.667s*	*0.886*
10000	*15.9891ms*	*63.1621ms*	*1m34.836s*	*0.8714*

3. List all the kernels that collectively consumed more than 90% of the kernel time and what percentage of the kernel time each kernel did consume (start with the kernel that consumed the most time, then list the next kernel, until you reach 90% or more).

For batch size of 10000:

Time (%)	*Total Time*	*Instances*	*Average*	*Minimum*	*Maximum*	*Name*
100.0	79133619	2	39566809.5	16114011	63019608	conv_forward_fernel

4. List all the CUDA API calls that collectively consumed more than 90% of the API time and what percentage of the API time each call did consume (start with the API call that consumed the most time, then list the next call, until you reach 90% or more).

For batch size of 10000:

Time (%)	*Total Time*	*Calls*	*Average*	*Minimum*	*Maximum*	*Name*
79.6	*1108373579*	*8*	*138546697.4*	*18859*	*576479762*	*cudaMemcpy*
13.3	*185045397*	*8*	*23130674.6*	*75624*	*180937369*	*cudaMalloc*

5. Explain the difference between kernels and CUDA API calls. Please give an example in your explanation for both.

Cuda API call is for the built-in functions such as cudaMalloc and cudaMemcpy. They are called by the main function.
The kernel is the cuda function we implemented, which is also called by main function.

6. Show a screenshot of the GPU SOL utilization

Batch size of 10000: