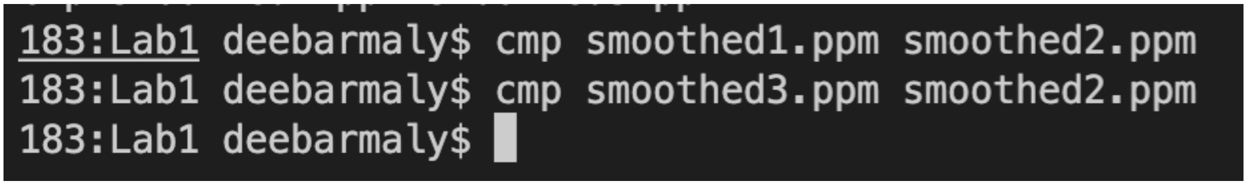
Deeb Armaly
ECE 4310
Lab 1
6 September 2022
C17525380

For this lab, three different smoothing algorithms were to be implemented. The output of the first algorithm is smoothed1.ppm, and the same naming follows for the second and third algorithms. As shown in the screenshot below, when compared, no differences appear in all three algorithms. This means the smoothed output images of all three algorithms are the same.



```
183:Lab1 deebarmaly$ cmp smoothed1.ppm smoothed2.ppm
183:Lab1 deebarmaly$ cmp smoothed3.ppm smoothed2.ppm
183:Lab1 deebarmaly$ █
```

The chart below shows the run times of each of the sorting algorithms. Each algorithm was run 10 times and the average of the ten runs are calculated below. Run times are shown in nano-seconds.

Basic 2D Convolution	**Separable Filters**	**Separable Filters and Sliding Window**
35215000	14621000	7562000
35268000	14648000	7484000
35456000	14812000	7325000
35416000	14546000	7637000
35336000	14728000	7576000
35392000	14696000	7631000
35316000	14835000	7632000
35402000	14759000	7552000
35505000	14594000	7741000
35410000	14932000	7528000
AVG: 35371600	AVG: 14717100	AVG: 7566800

As can be seen from the chart, utilizing the smoothing algorithm with separable filters lowered the average runtime by 20654500 nsec compared to basic 2D convolution. Utilizing separable filters along with a sliding window further lowered the average runtime by 27804800 nsec compared to basic 2D convolution and 7150300 nsec from just using separable filters alone.