**Blue Waters Petascale Semester Curriculum v1.0**

**Unit 7: CUDA**

**Lesson 2: Image Processing**

**Exercise Instructions for Students**

*Developed by Michael D. Shah for the Shodor Education Foundation, Inc.*



*Except where otherwise noted, this work by The Shodor Education Foundation, Inc. is licensed under CC BY-SA 4.0. To view a copy of this license, visit*[*https://creativecommons.org/licenses/by-sa/4.0*](https://creativecommons.org/licenses/by-sa/4.0)

*Browse and search the full curriculum at*[*http://shodor.org/petascale/materials/semester-curriculum*](http://shodor.org/petascale/materials/semester-curriculum)

*We welcome your improvements! You can submit your proposed changes to this material and the rest of the curriculum in our GitHub repository at*[*https://github.com/shodor-education/petascale-semester-curriculum*](https://github.com/shodor-education/petascale-semester-curriculum)

*We want to hear from you! Please let us know your experiences using this material by sending email to* [*petascale@shodor.org*](mailto:petascale@shodor.org)

Some sample code has been provided along the lecture.

1. Students should first start by modifying the 'make\_brighter' function.

\_\_global\_\_

void make\_brighter(unsigned char\* pixel\_array, int size){

int id = threadIdx.x+blockDim.x\*blockIdx.x;

// Check to make sure we do not have more threads

// than the index of our array

if(id < size){

pixel\_array[id] \*= 6;

}

}

* Modify the function to make an image darker for example.

2. Students should then implement the grayscale kernel filter.

3. Try creating a filter that filters out the 'green' and 'blue' components of the image.

* (i.e. A kernel that zeroes out the red and green components)