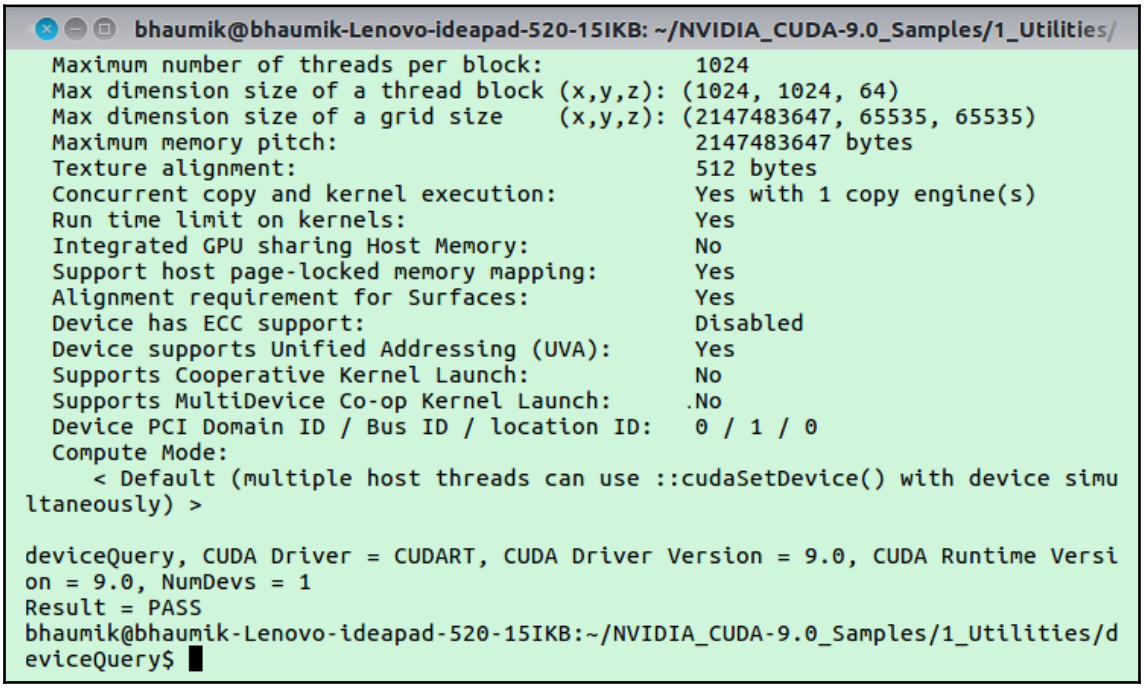
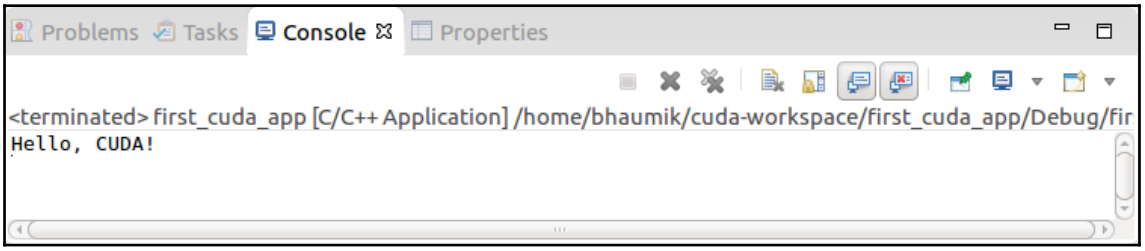
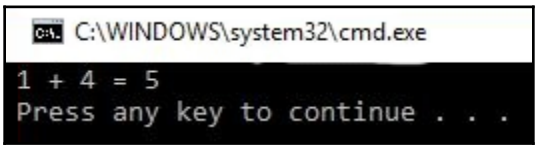
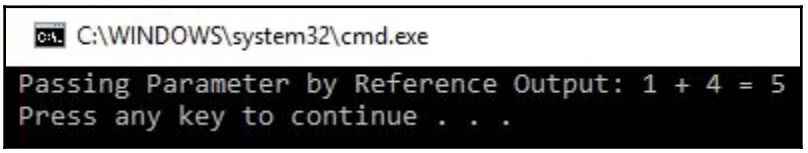
Chapter 1: Introducing CUDA and Getting Started with CUDA

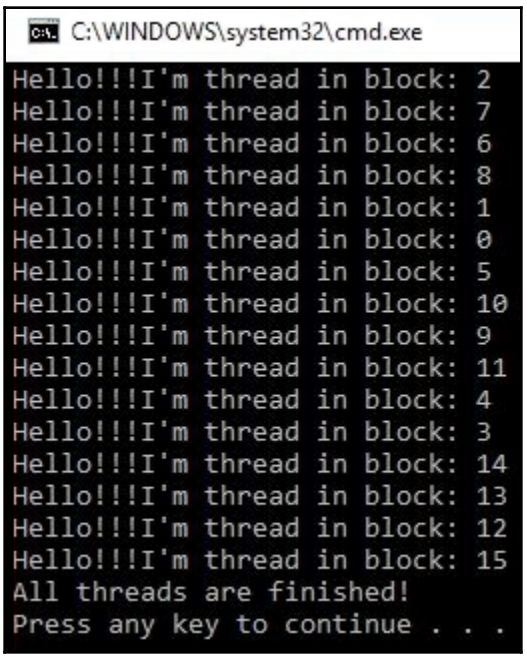




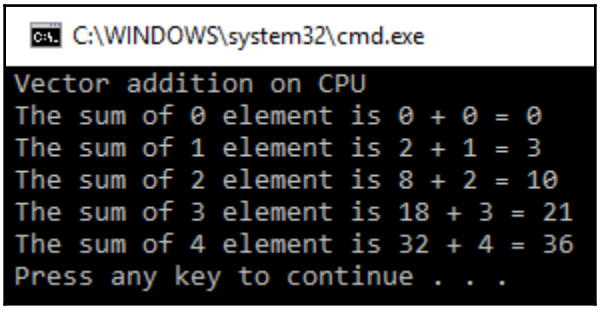
Chapter 2: Parallel Programming using CUDA C

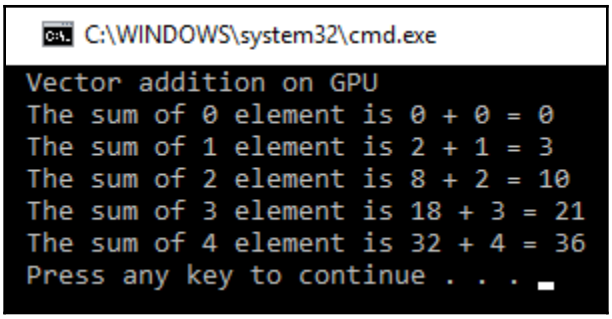


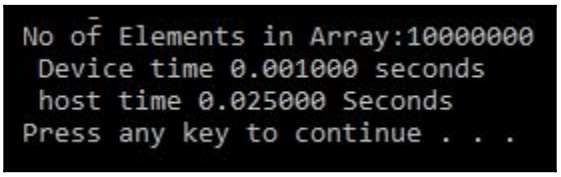


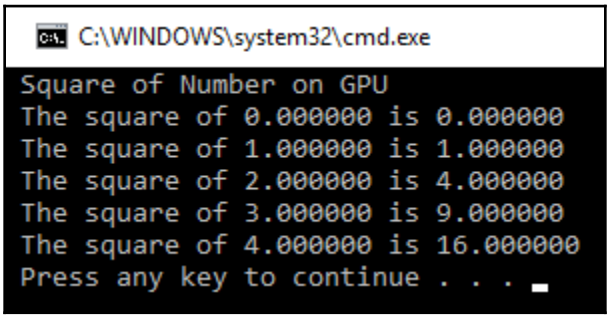


```
C:\WINDOWS\system32\cmd.exe
Detected 1 CUDA Capable device(s)

Device 0: "GeForce 940MX"
  CUDA Driver Version / Runtime Version          9.1 / 9.0
  CUDA Capability Major/Minor version number:    5.0
  Total amount of global memory:                 4096 MBytes (4294967296 bytes)
  ( 3) Multiprocessors  GPU Max Clock rate:                            1189 MHz (1.19 GHz)
  Memory Clock rate:                             2505 Mhz
  Memory Bus Width:                              64-bit
  L2 Cache Size:                                 1048576 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(65536), 2D=(65536, 65536), 3D=(4096, 4096, 4096)
  Maximum Layered 1D Texture Size, (num) layers  1D=(16384), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(16384, 16384), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 65536
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  2048
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 1 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            No
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  CUDA Device Driver Mode (TCC or WDDM):         WDDM (Windows Display Driver Model)
  Device supports Unified Addressing (UVA):      Yes
  Supports Cooperative Kernel Launch:            No
  Supports MultiDevice Co-op Kernel Launch:      No
  Device PCI Domain ID / Bus ID / location ID:   0 / 1 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >
Press any key to continue . . .
```




C:\WINDOWS\system32\cmd.exe
Vector addition on GPU
The sum of 0 element is 0 + 0 = 0
The sum of 1 element is 2 + 1 = 3
The sum of 2 element is 8 + 2 = 10
The sum of 3 element is 18 + 3 = 21
The sum of 4 element is 32 + 4 = 36
Press any key to continue . . .

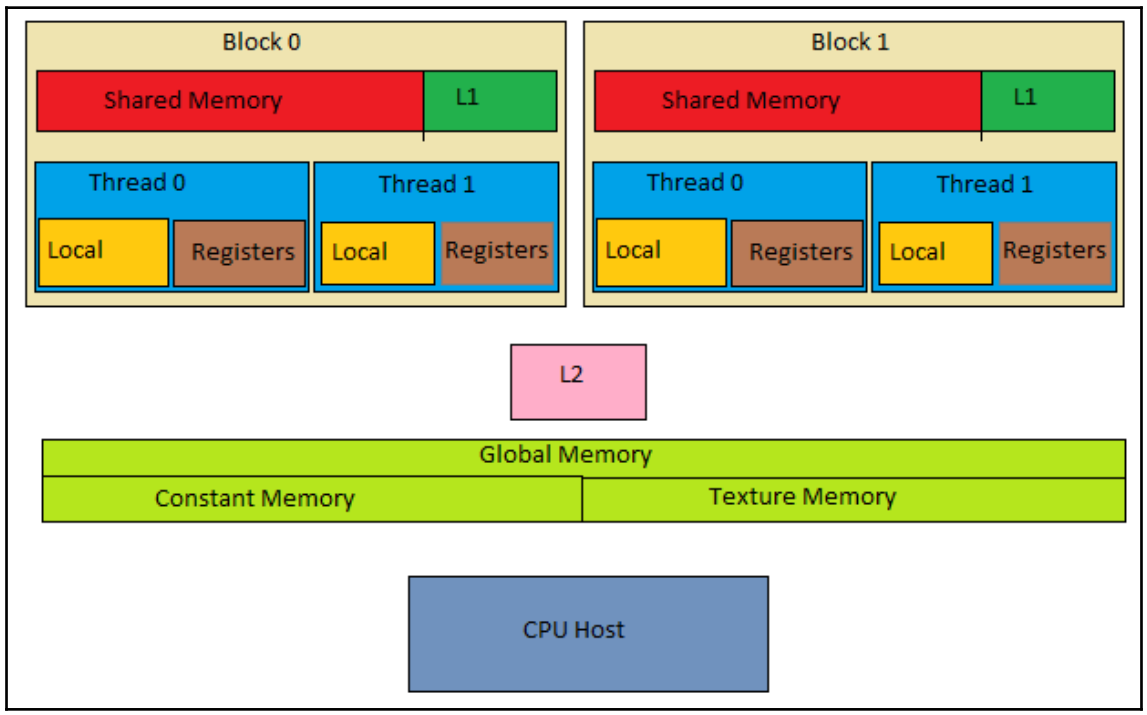

No of Elements in Array:10000000
Device time 0.001000 seconds
host time 0.025000 Seconds
Press any key to continue . . .

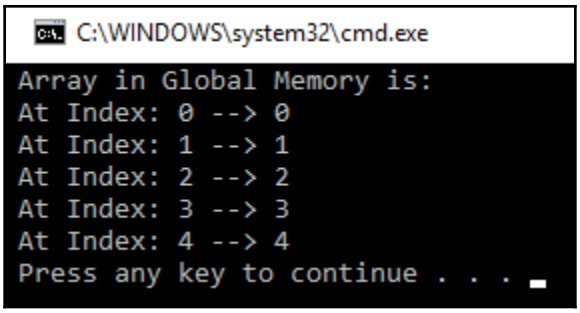

C:\WINDOWS\system32\cmd.exe
Square of Number on GPU
The square of 0.000000 is 0.000000
The square of 1.000000 is 1.000000
The square of 2.000000 is 4.000000
The square of 3.000000 is 9.000000
The square of 4.000000 is 16.000000
Press any key to continue . . .

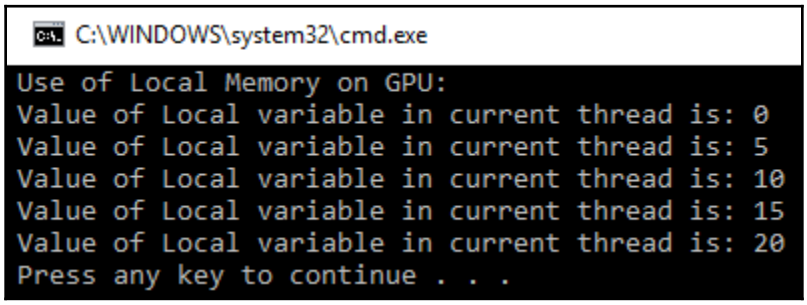
Chapter 3: Threads, Synchronization, and Memory

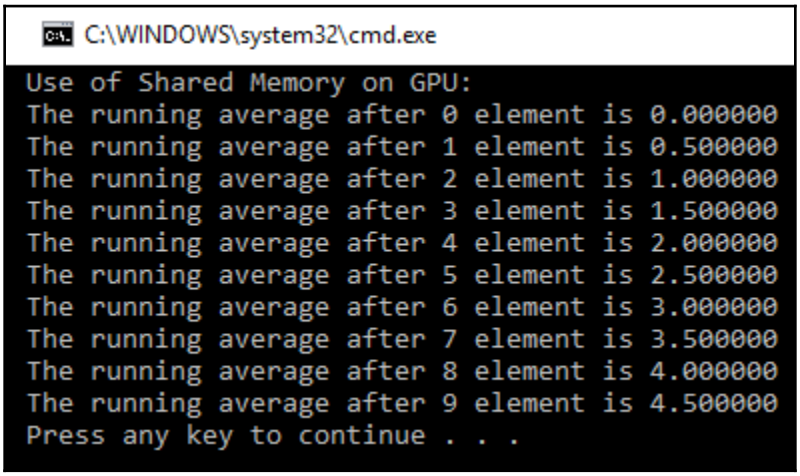
Block 0	Thread 0	Thread 1	Thread 2
Block 1	Thread 0	Thread 1	Thread 2
Block 2	Thread 0	Thread 1	Thread 2

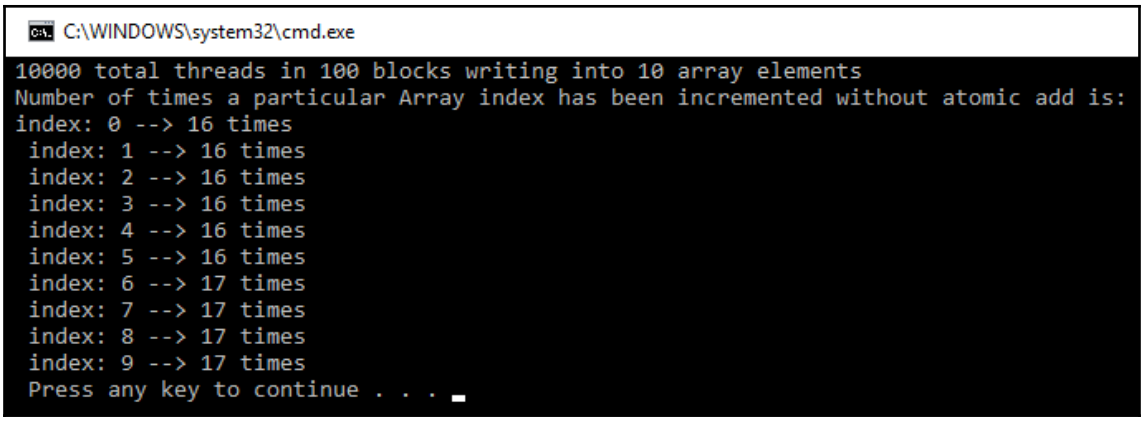
```
C:\WINDOWS\system32\cmd.exe
Vector addition on GPU
GPU has computed Sum Correctly
Press any key to continue . . .
```

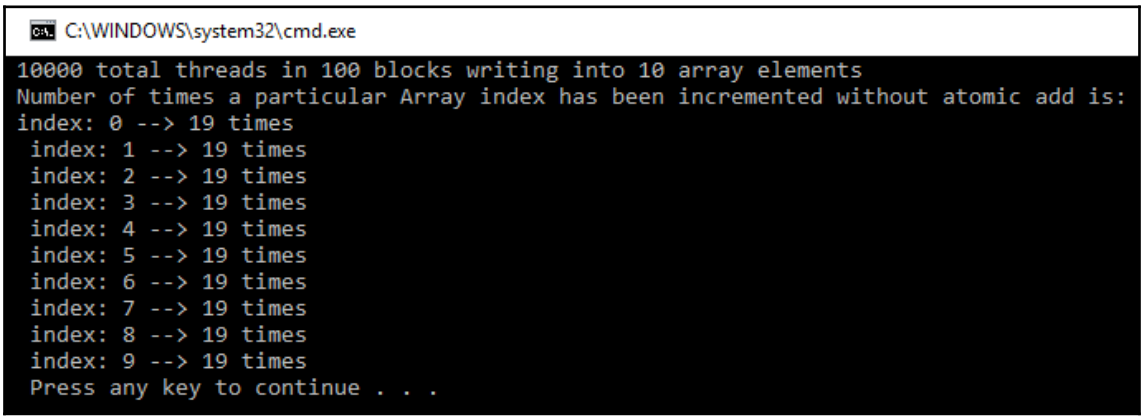



C:\WINDOWS\system32\cmd.exe
Array in Global Memory is:
At Index: 0 --> 0
At Index: 1 --> 1
At Index: 2 --> 2
At Index: 3 --> 3
At Index: 4 --> 4
Press any key to continue . . .


C:\WINDOWS\system32\cmd.exe
Use of Local Memory on GPU:
Value of Local variable in current thread is: 0
Value of Local variable in current thread is: 5
Value of Local variable in current thread is: 10
Value of Local variable in current thread is: 15
Value of Local variable in current thread is: 20
Press any key to continue . . .


C:\WINDOWS\system32\cmd.exe
Use of Shared Memory on GPU:
The running average after 0 element is 0.000000
The running average after 1 element is 0.500000
The running average after 2 element is 1.000000
The running average after 3 element is 1.500000
The running average after 4 element is 2.000000
The running average after 5 element is 2.500000
The running average after 6 element is 3.000000
The running average after 7 element is 3.500000
The running average after 8 element is 4.000000
The running average after 9 element is 4.500000
Press any key to continue . . .


C:\WINDOWS\system32\cmd.exe
10000 total threads in 100 blocks writing into 10 array elements
Number of times a particular Array index has been incremented without atomic add is:
index: 0 --> 16 times
index: 1 --> 16 times
index: 2 --> 16 times
index: 3 --> 16 times
index: 4 --> 16 times
index: 5 --> 16 times
index: 6 --> 17 times
index: 7 --> 17 times
index: 8 --> 17 times
index: 9 --> 17 times
Press any key to continue . . .


C:\WINDOWS\system32\cmd.exe
10000 total threads in 100 blocks writing into 10 array elements
Number of times a particular Array index has been incremented without atomic add is:
index: 0 --> 19 times
index: 1 --> 19 times
index: 2 --> 19 times
index: 3 --> 19 times
index: 4 --> 19 times
index: 5 --> 19 times
index: 6 --> 19 times
index: 7 --> 19 times
index: 8 --> 19 times
index: 9 --> 19 times
Press any key to continue . . .


C:\WINDOWS\system32\cmd.exe
10000 total threads in 100 blocks writing into 10 array elements
Number of times a particular Array index has been incremented is:
index: 0 --> 1000 times
index: 1 --> 1000 times
index: 2 --> 1000 times
index: 3 --> 1000 times
index: 4 --> 1000 times
index: 5 --> 1000 times
index: 6 --> 1000 times
index: 7 --> 1000 times
index: 8 --> 1000 times
index: 9 --> 1000 times
Press any key to continue . . .

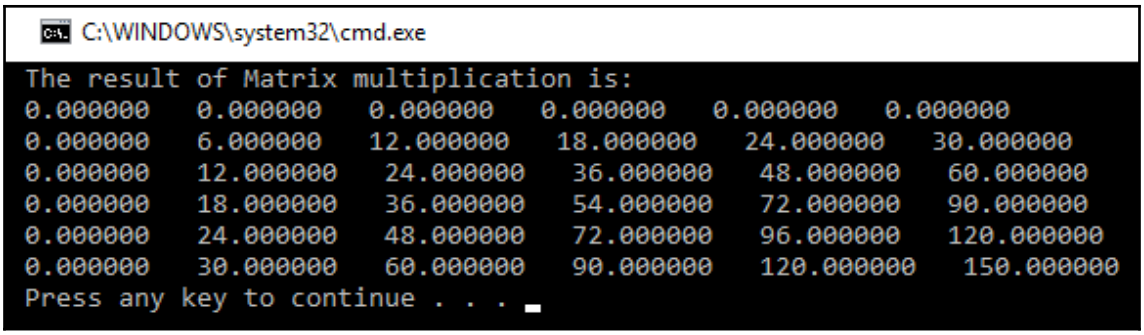
```
C:\WINDOWS\system32\cmd.exe
Use of Constant memory on GPU
The expression for input 0.000000 is 20.000000
The expression for input 1.000000 is 22.000000
The expression for input 2.000000 is 24.000000
The expression for input 3.000000 is 26.000000
The expression for input 4.000000 is 28.000000
Press any key to continue . . .
```

```
C:\WINDOWS\system32\cmd.exe
Use of Texture memory on GPU:
Texture element at 0 is : 0.000000
Texture element at 1 is : 1.000000
Texture element at 2 is : 2.000000
Texture element at 3 is : 3.000000
Texture element at 4 is : 4.000000
Texture element at 5 is : 5.000000
Texture element at 6 is : 6.000000
Texture element at 7 is : 7.000000
Texture element at 8 is : 8.000000
Texture element at 9 is : 9.000000
Press any key to continue . . .
```

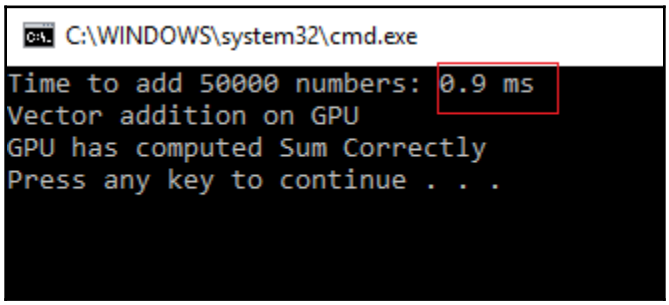
```
C:\WINDOWS\system32\cmd.exe
The computed dot product is: 1047552.000000
The dot product computed by GPU is correct
Press any key to continue . . .
```

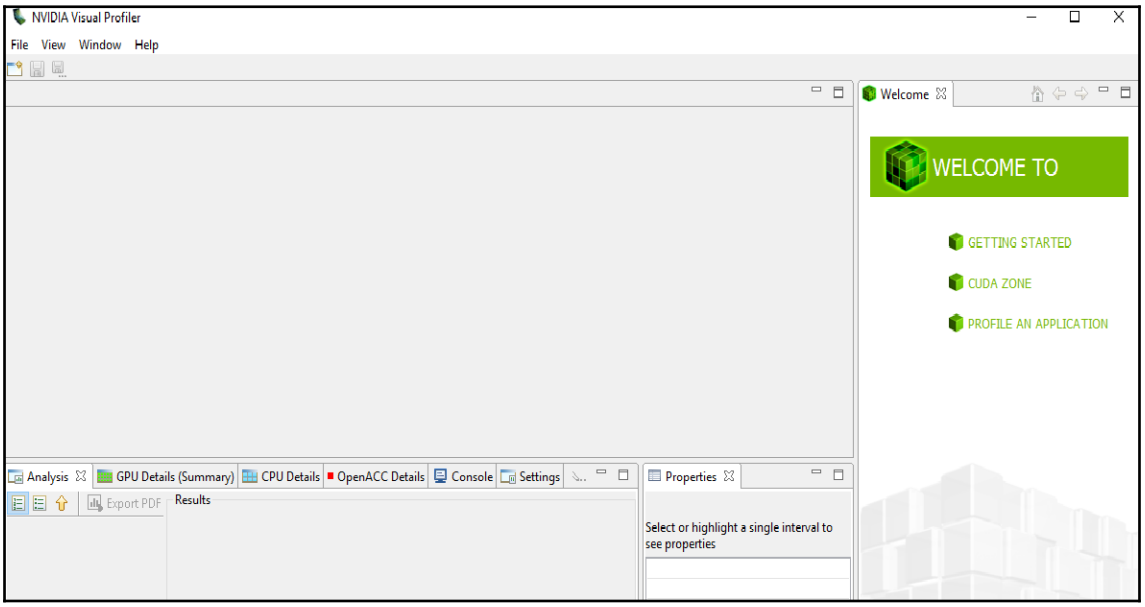
$$\begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 \\ 3 & 3 & 3 & 3 \end{bmatrix} * \begin{bmatrix} 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \\ 0 & 1 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 0*0+0*0+0*0+0*0 & 0 & 0 & 0 \\ 1*0+1*0+1*0+1*0 & 4 & 8 & 12 \\ 2*0+2*0+2*0+2*0 & 8 & 16 & 24 \\ 3*0+3*0+3*0+3*0 & 12 & 24 & 36 \end{bmatrix}$$

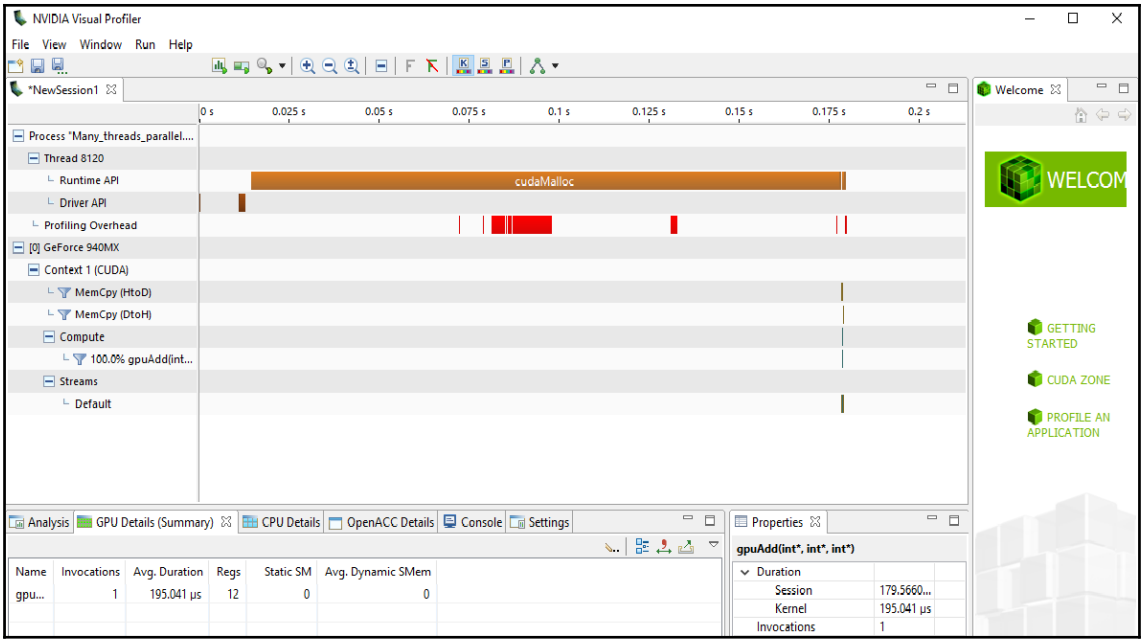
$$\begin{bmatrix} M_{00} & M_{01} \\ M_{10} & M_{11} \end{bmatrix} \longrightarrow \begin{bmatrix} M_{00} & M_{01} & M_{10} & M_{11} \end{bmatrix}$$

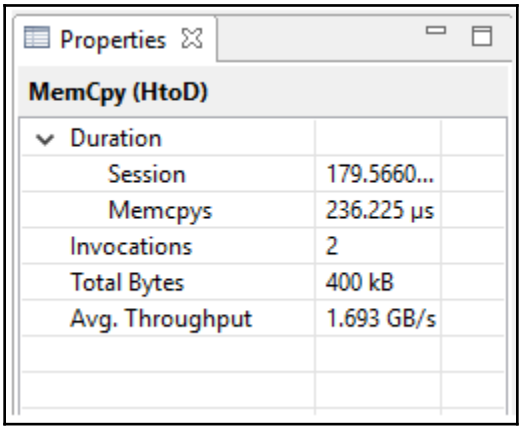

C:\WINDOWS\system32\cmd.exe
The result of Matrix multiplication is:
0.000000 0.000000 0.000000 0.000000 0.000000 0.000000
0.000000 6.000000 12.000000 18.000000 24.000000 30.000000
0.000000 12.000000 24.000000 36.000000 48.000000 60.000000
0.000000 18.000000 36.000000 54.000000 72.000000 90.000000
0.000000 24.000000 48.000000 72.000000 96.000000 120.000000
0.000000 30.000000 60.000000 90.000000 120.000000 150.000000
Press any key to continue . . .

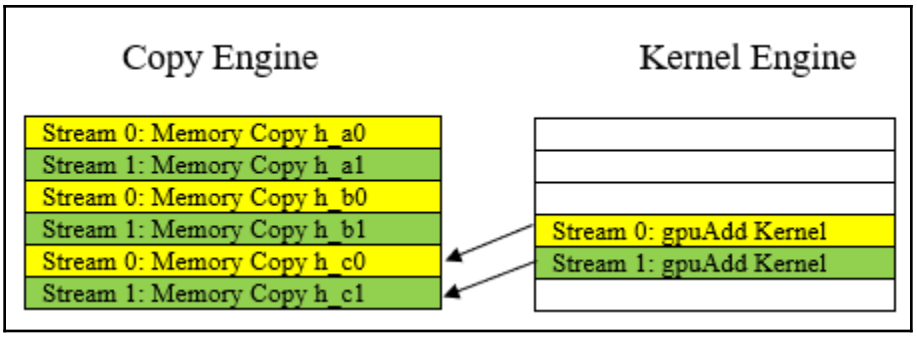
Chapter 4: Advanced Concepts in CUDA

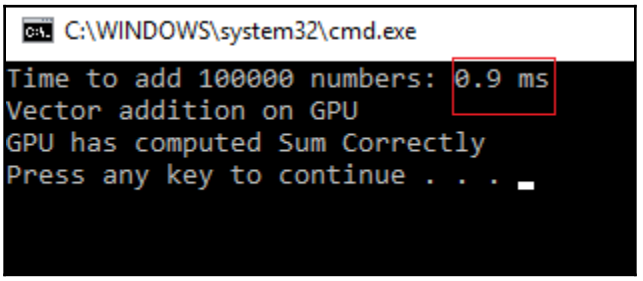


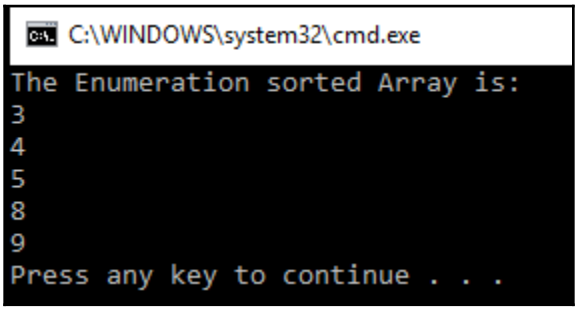


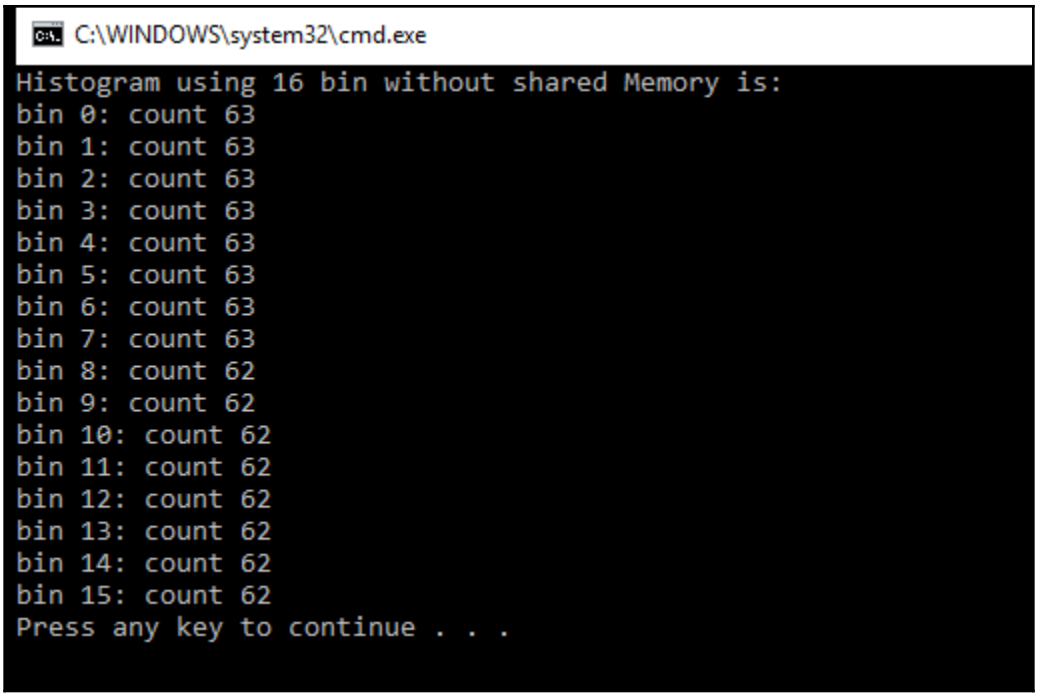

NVIDIA Visual Profiler
File View Window Run Help
*NewSession1
0 s
0.025 s
0.05 s
0.075 s
0.1 s
0.125 s
0.15 s
0.175 s
0.2 s
Process 'Many_threads_parallel....
Thread 8120
Runtime API
cudaMalloc
Driver API
Profiling Overhead
[0] GeForce 940MX
Context 1 (CUDA)
MemCpy (HtoD)
MemCpy (DtoH)
Compute
100.0% gpuAdd(int...
Streams
Default
Welcome
GETTING STARTED
CUDA ZONE
PROFILE AN APPLICATION
Analysis
GPU Details (Summary)
CPU Details
OpenACC Details
Console
Settings
Name
Invocations
Avg. Duration
Regs
Static SM
Avg. Dynamic SMem
gpu...
1
195.041 µs
12
0
0
Properties
gpuAdd(int*, int*, int*)
Duration
Session
179.5660...
Kernel
195.041 µs
Invocations
1


Properties
MemCpy (HtoD)
Duration
Session
179.5660...
Memcpys
236.225 µs
Invocations
2
Total Bytes
400 kB
Avg. Throughput
1.693 GB/s


Copy Engine
Kernel Engine
Stream 0: Memory Copy h_a0
Stream 1: Memory Copy h_a1
Stream 0: Memory Copy h_b0
Stream 1: Memory Copy h_b1
Stream 0: Memory Copy h_c0
Stream 1: Memory Copy h_c1
Stream 0: gpuAdd Kernel
Stream 1: gpuAdd Kernel

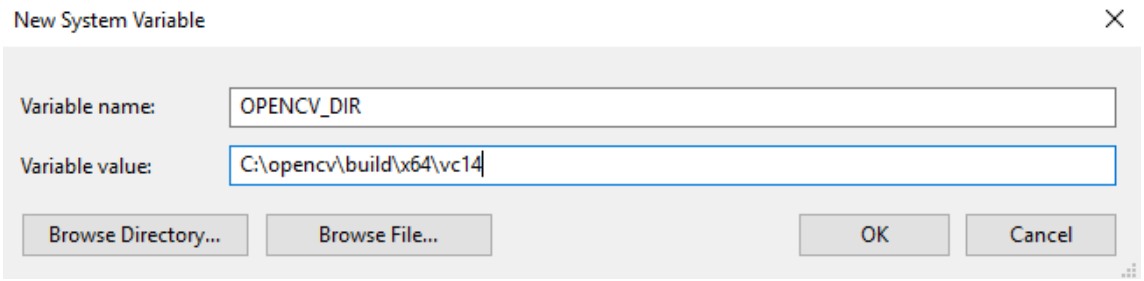

C:\WINDOWS\system32\cmd.exe
Time to add 100000 numbers: 0.9 ms
Vector addition on GPU
GPU has computed Sum Correctly
Press any key to continue . . .

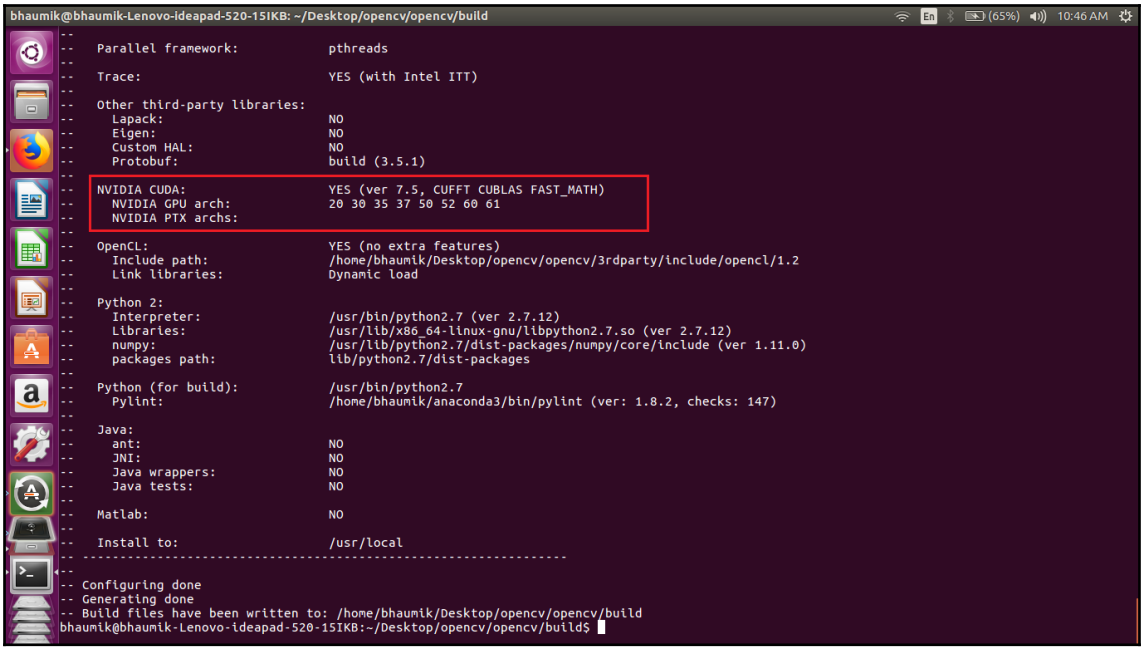

C:\WINDOWS\system32\cmd.exe
The Enumeration sorted Array is:
3
4
5
8
9
Press any key to continue . . .


C:\WINDOWS\system32\cmd.exe
Histogram using 16 bin without shared Memory is:
bin 0: count 63
bin 1: count 63
bin 2: count 63
bin 3: count 63
bin 4: count 63
bin 5: count 63
bin 6: count 63
bin 7: count 63
bin 8: count 62
bin 9: count 62
bin 10: count 62
bin 11: count 62
bin 12: count 62
bin 13: count 62
bin 14: count 62
bin 15: count 62
Press any key to continue . . .


C:\WINDOWS\system32\cmd.exe
bin 227: count 4
bin 228: count 4
bin 229: count 4
bin 230: count 4
bin 231: count 4
bin 232: count 3
bin 233: count 3
bin 234: count 3
bin 235: count 3
bin 236: count 3
bin 237: count 3
bin 238: count 3
bin 239: count 3
bin 240: count 3
bin 241: count 3
bin 242: count 3
bin 243: count 3
bin 244: count 3
bin 245: count 3
bin 246: count 3
bin 247: count 3
bin 248: count 3
bin 249: count 3
bin 250: count 3
bin 251: count 3
bin 252: count 3
bin 253: count 3
bin 254: count 3
bin 255: count 3
Press any key to continue . . .

Chapter 5: Getting Started with OpenCV with CUDA Support

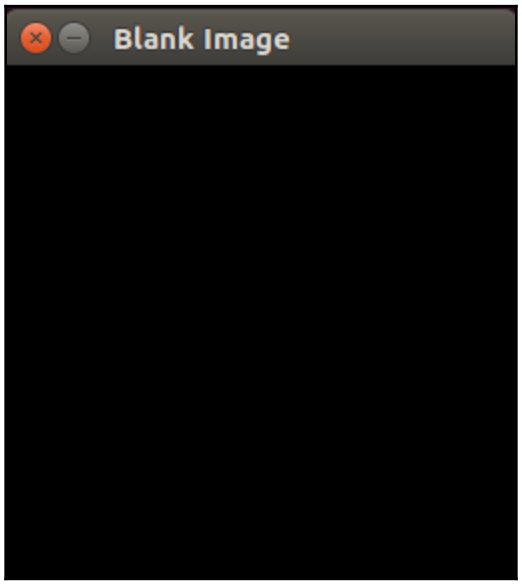


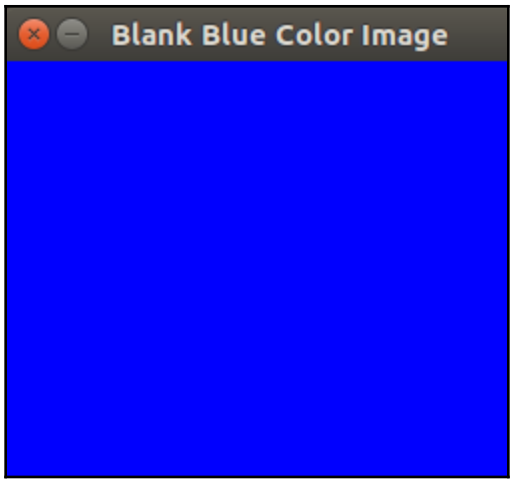


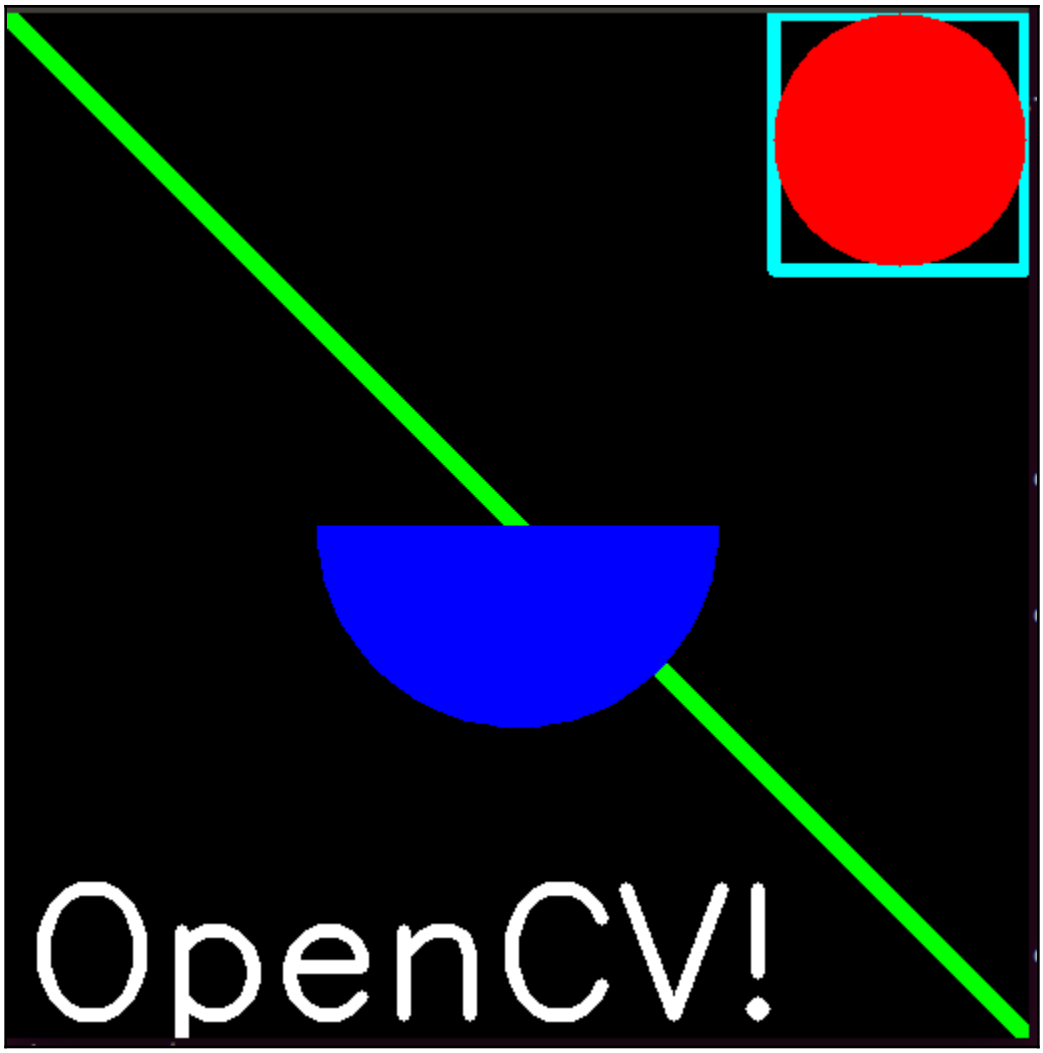

My First Opencv Program


My First Opencv Program

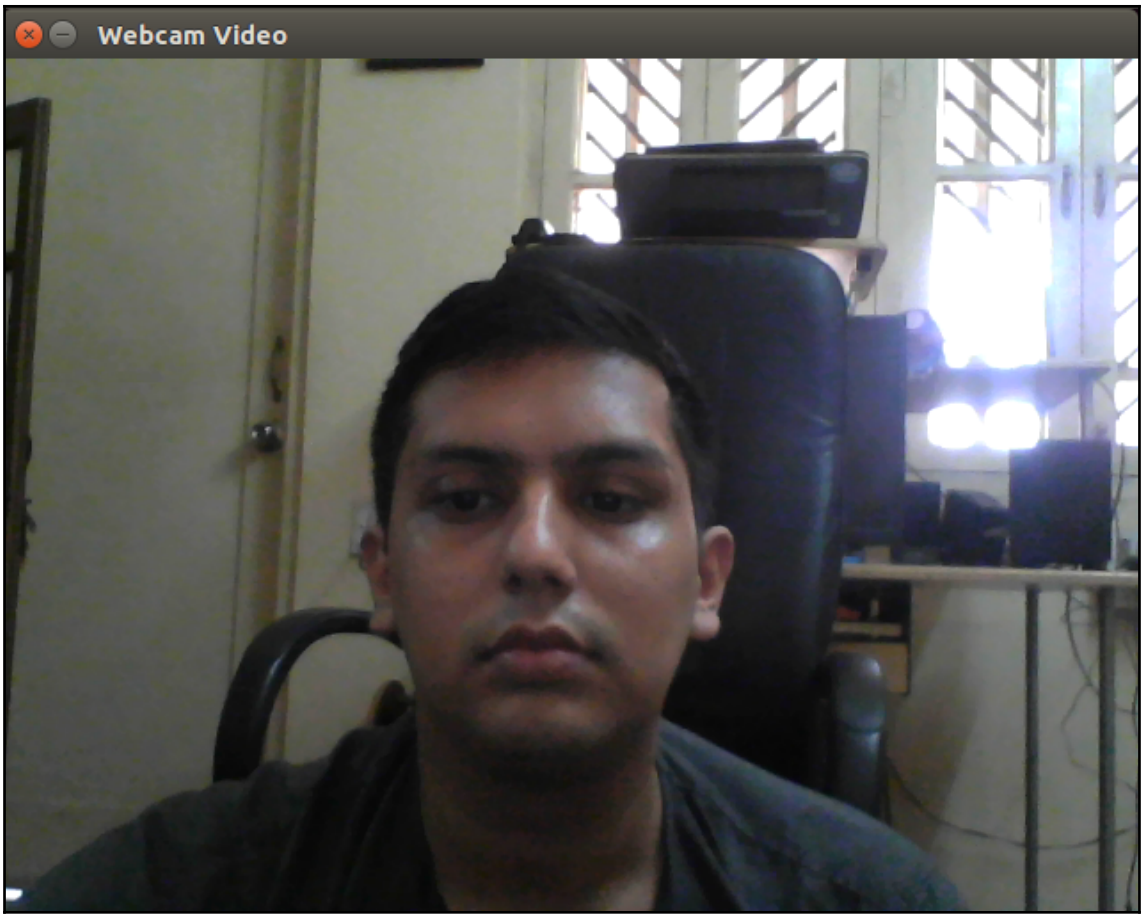

My First Opencv Program

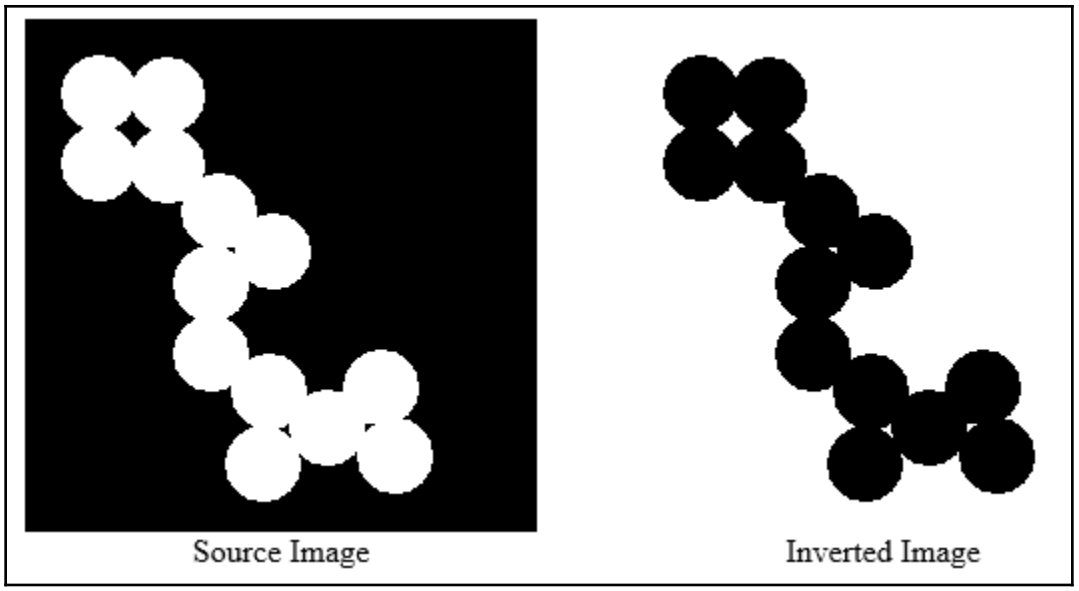

Blank Image


Blank Blue Color Image


OpenCV!

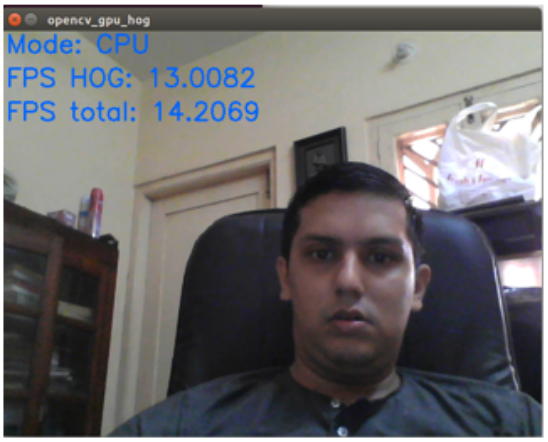

First Video


Webcam Video

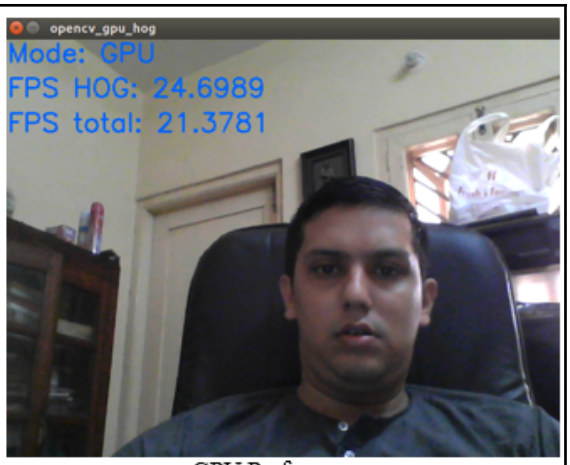



```
Performance of Thresholding on CPU:
Time: 0.169766
FPS: 5.89046
```

```
Performance of Thresholding on GPU:
Time: 0.000550593
FPS: 1816.22
```



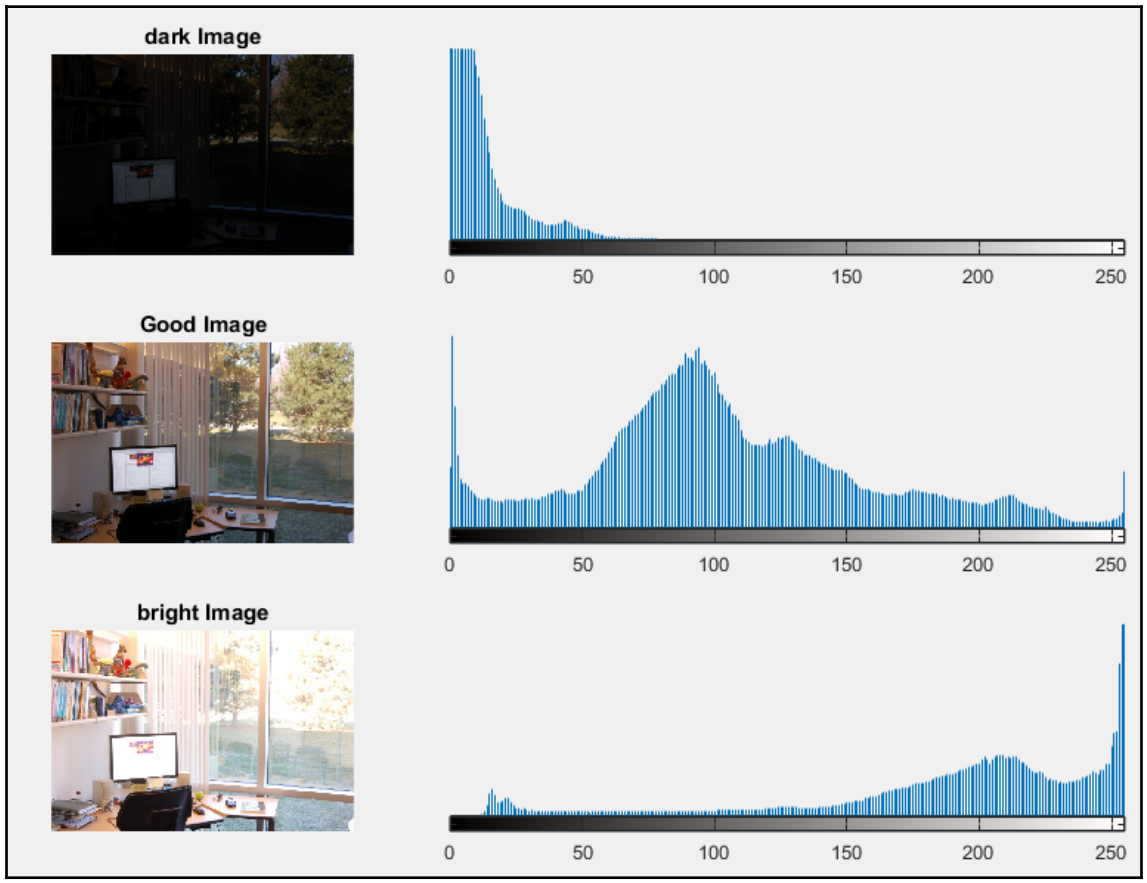
CPU Performance



GPU Performance

Chapter 6: Basic Computer Vision Operations Using OpenCV and CUDA

```
bhaumik@bhaumik-Lenovo-ideapad-520-15IKB:~/Desktop/opencv/Chapter 6$ g++ -std=c+
+11 individual_pixel.cpp `pkg-config --libs --cflags opencv` -o pixel
bhaumik@bhaumik-Lenovo-ideapad-520-15IKB:~/Desktop/opencv/Chapter 6$ ./pixel
Pixel Intensity of gray scale Image at (100,50) is:9
Pixel Intensity of color Image at (100,50) is:[175, 179, 177]
```




Original Image
Histogram Equalized Image

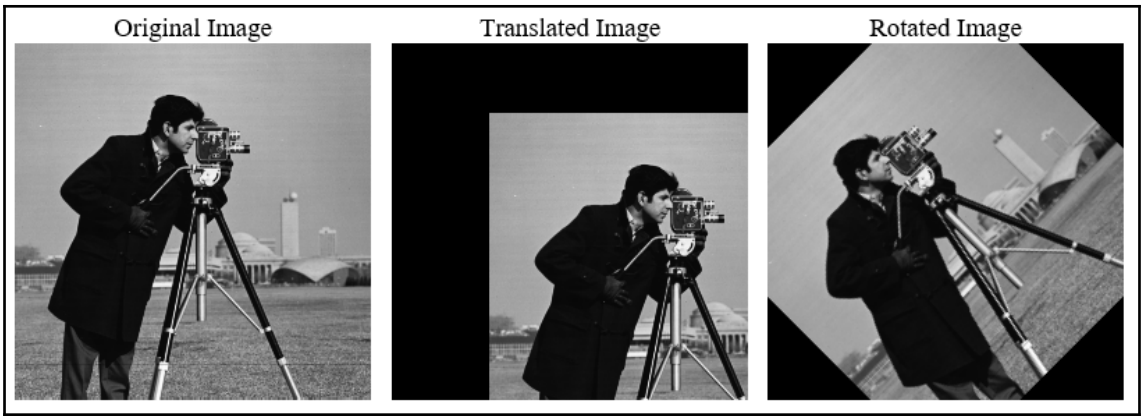

Original Image
Histogram Equalized Image
Histogram Equalized Image

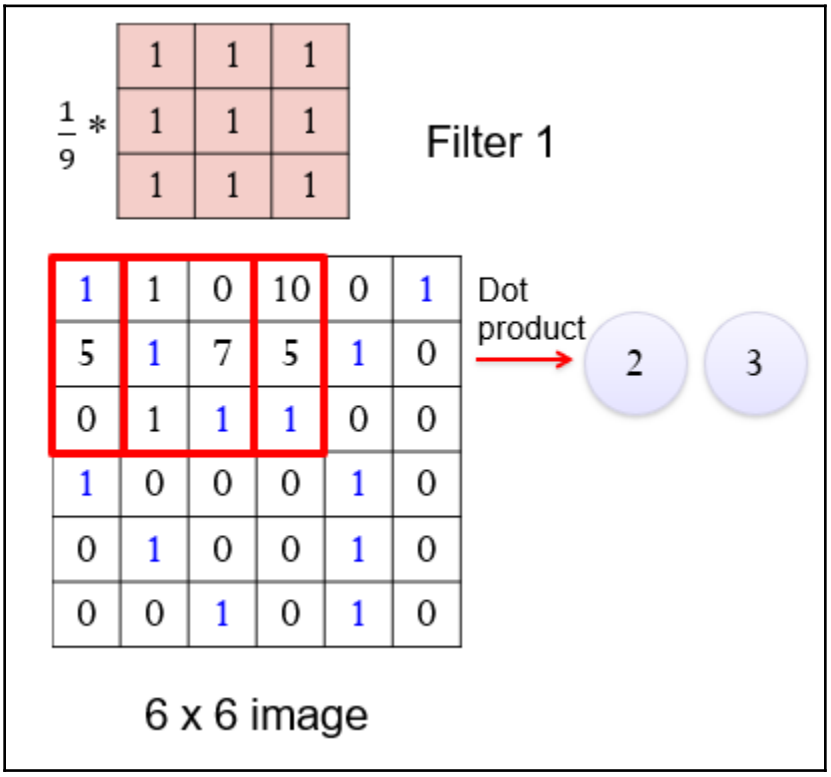
Original Image

Image Resize (200,200)

Image Resize downscaled by 2

$$\begin{bmatrix} 1 & 0 & tx \\ 0 & 1 & ty \end{bmatrix}$$


Original Image
Translated Image
Rotated Image


1/9 *
1 1 1
1 1 1
1 1 1
Filter 1
1 1 0 10 0 1
5 1 7 5 1 0
0 1 1 1 0 0
1 0 0 0 1 0
0 1 0 0 1 0
0 0 1 0 1 0
Dot product
2
3
6 x 6 image

Averaging Filter 3x3

$$\frac{1}{9} * \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Averaging Filter 5x5

$$\frac{1}{25} * \begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

Original Image

Average Filter (3x3)

Average Filter (5x5)

Average Filter (7x7)

Gaussian Filter 5x5

$$\frac{1}{273} * \begin{bmatrix} 1 & 4 & 7 & 4 & 1 \\ 4 & 16 & 26 & 16 & 4 \\ 7 & 26 & 41 & 26 & 7 \\ 4 & 16 & 26 & 16 & 4 \\ 1 & 4 & 7 & 4 & 1 \end{bmatrix}$$

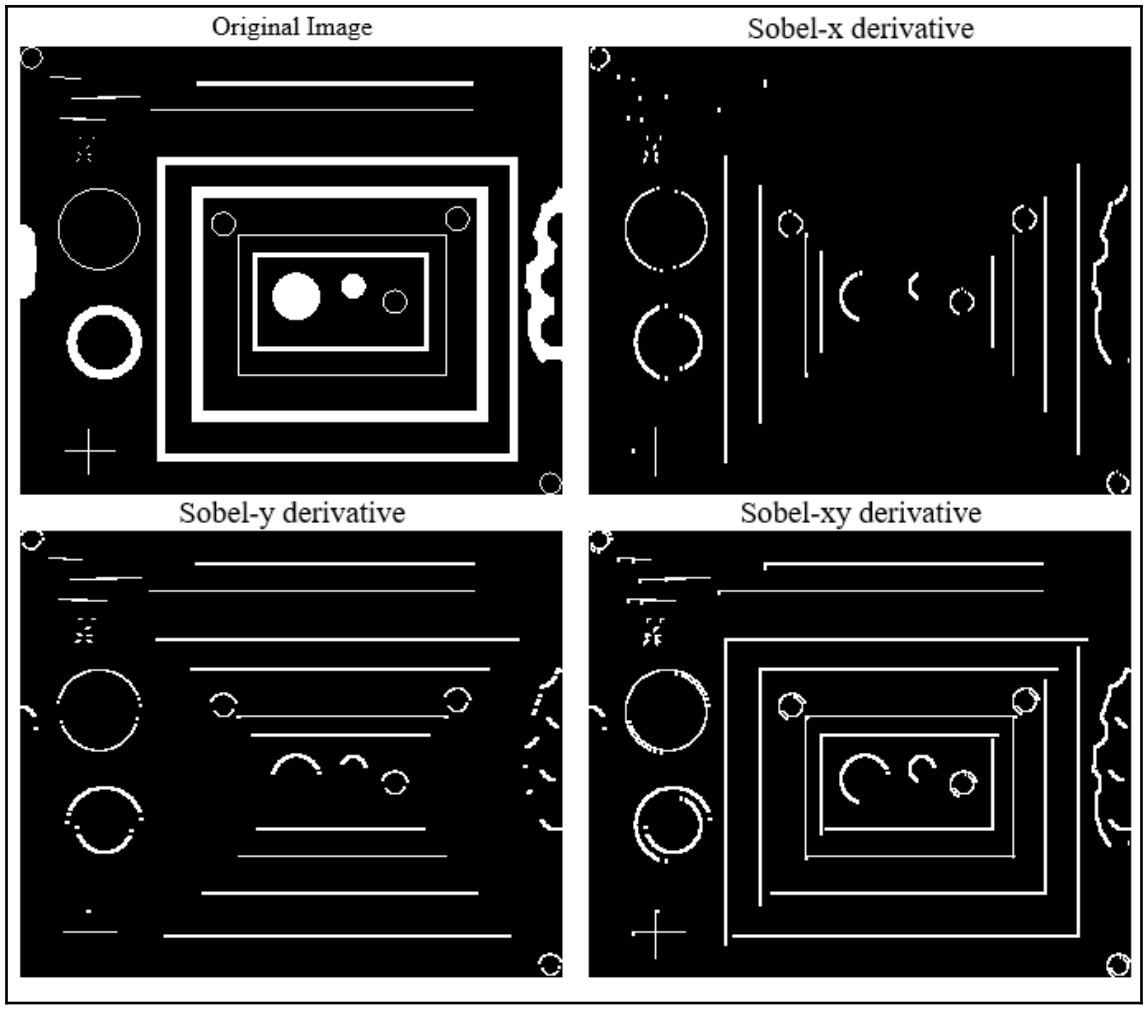
Original Image

Gaussian Filter (3x3)

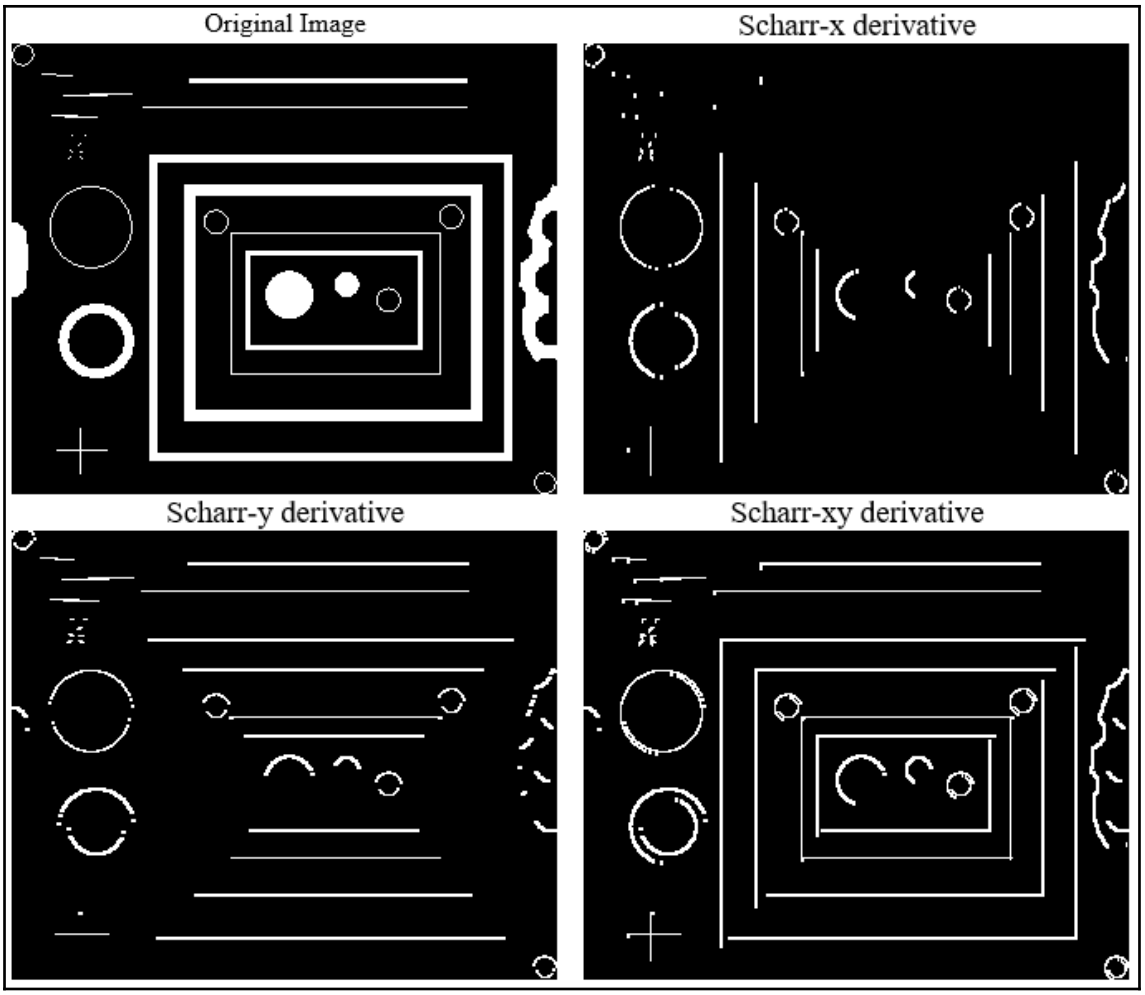
Gaussian Filter (5x5)

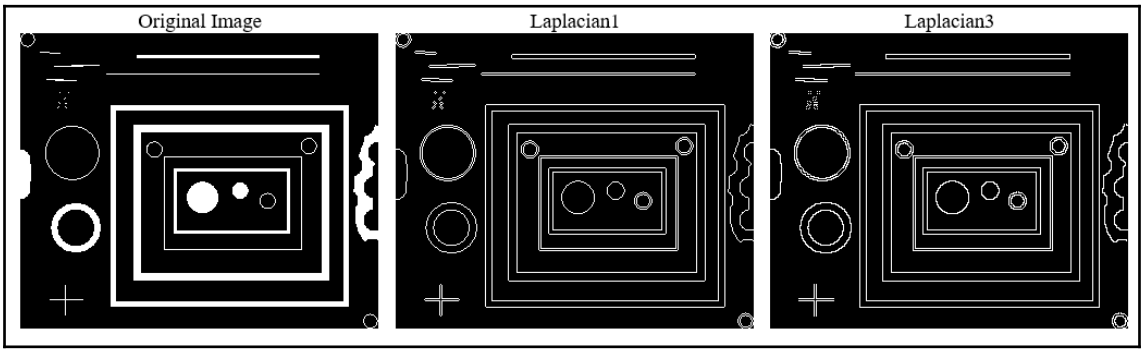
Gaussian Filter (7x7)

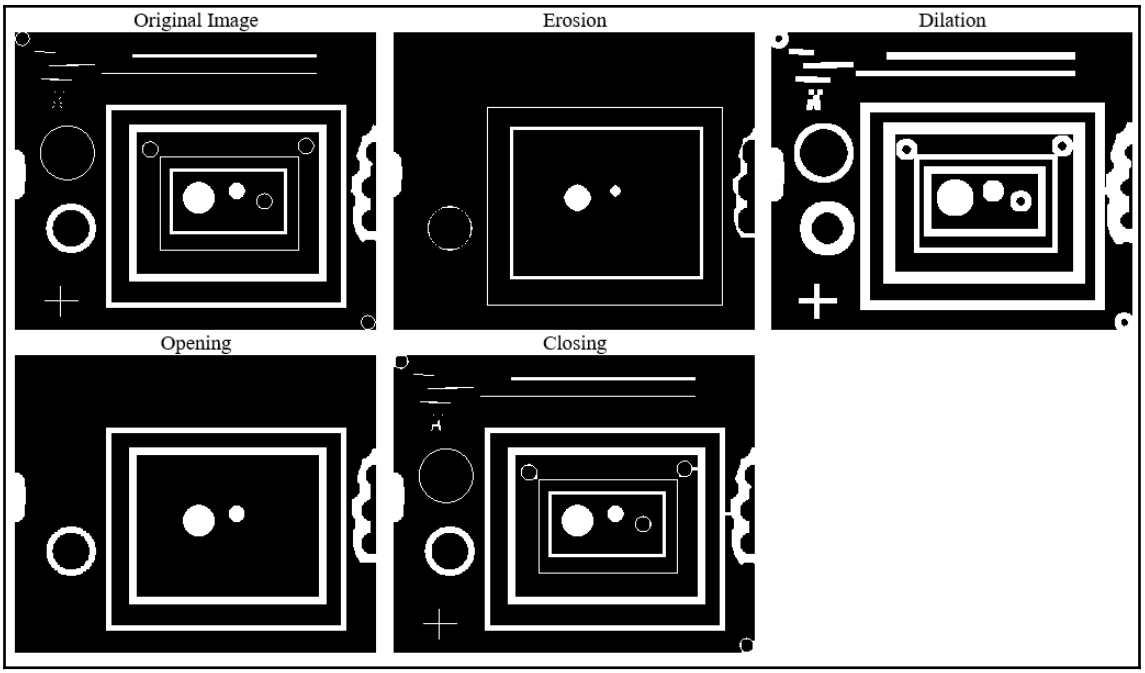
$$Sx = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \qquad Sy = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

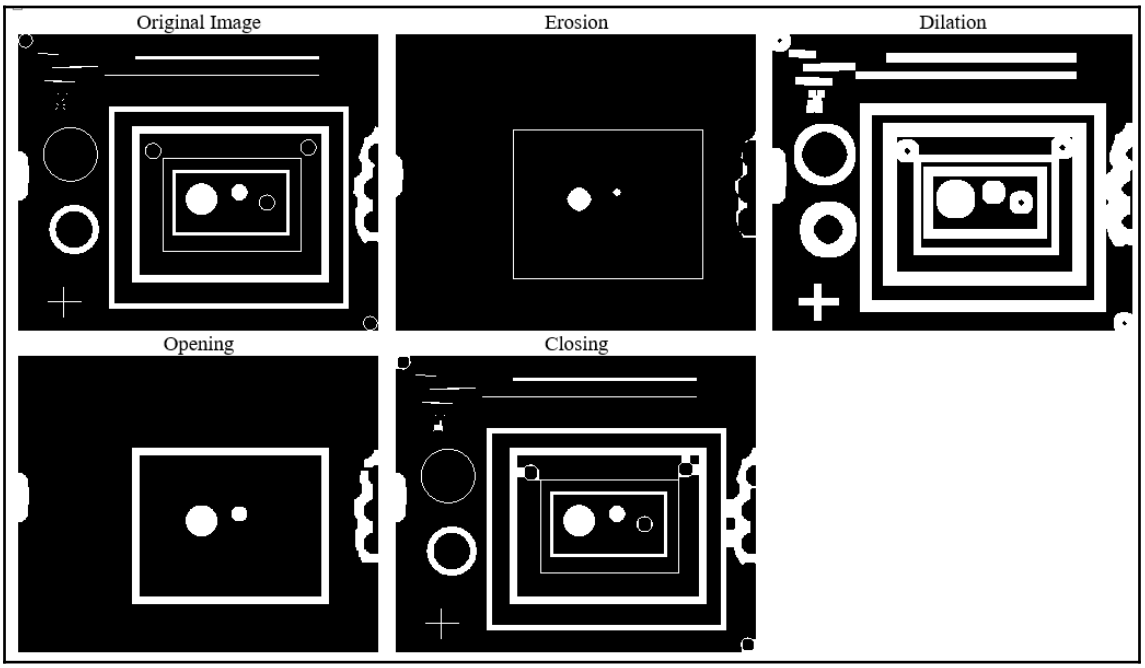


$$Sx = \begin{bmatrix} 3 & 0 & -3 \\ 10 & 0 & -10 \\ 3 & 0 & -3 \end{bmatrix} \qquad Sy = \begin{bmatrix} 3 & 10 & 3 \\ 0 & 0 & 0 \\ -3 & -10 & -3 \end{bmatrix}$$

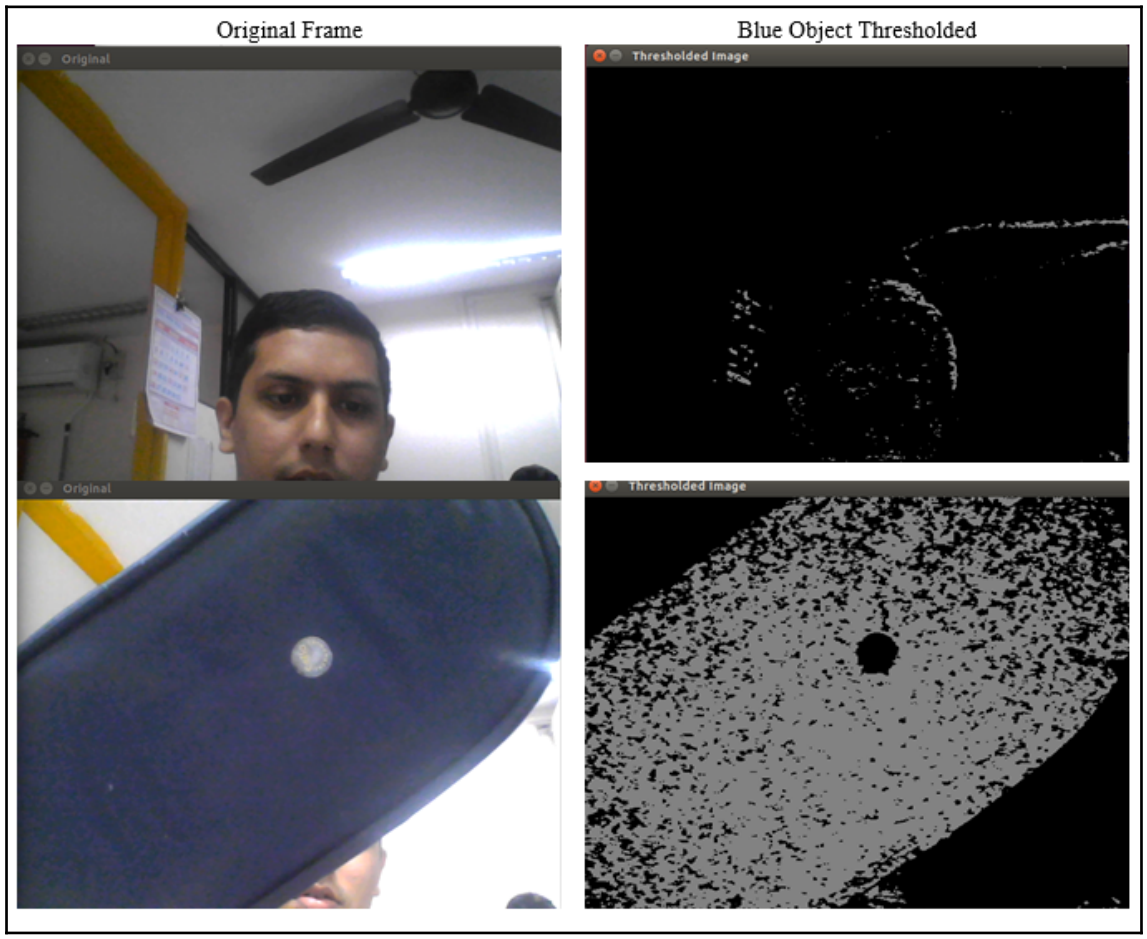

Original Image
Scharr-x derivative
Scharr-y derivative
Scharr-xy derivative

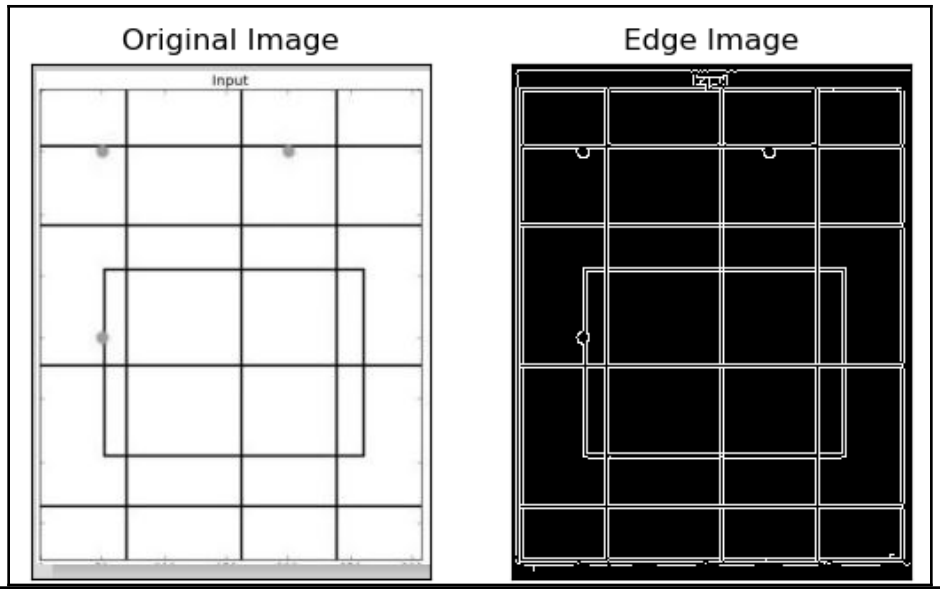

Original Image
Laplacian1
Laplacian3

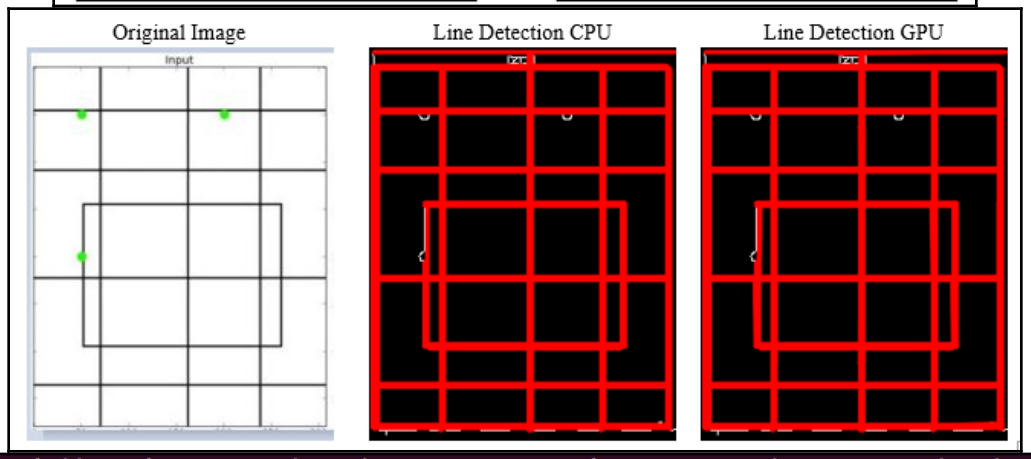

Original Image
Erosion
Dilation
Opening
Closing


Original Image
Erosion
Dilation
Opening
Closing

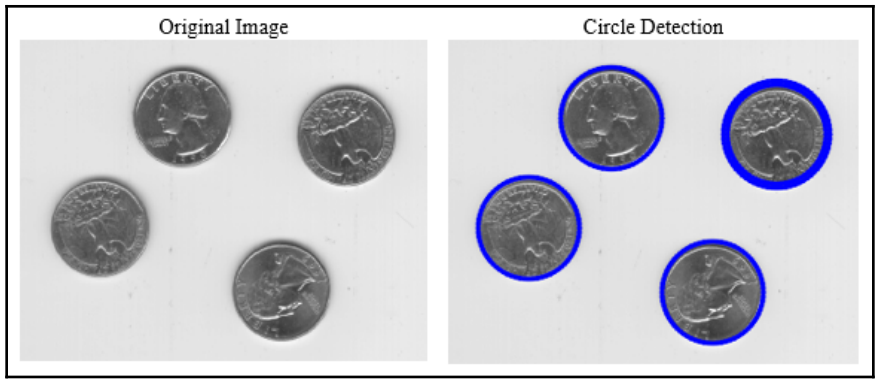
Chapter 7: Object Detection and Tracking Using OpenCV and CUDA

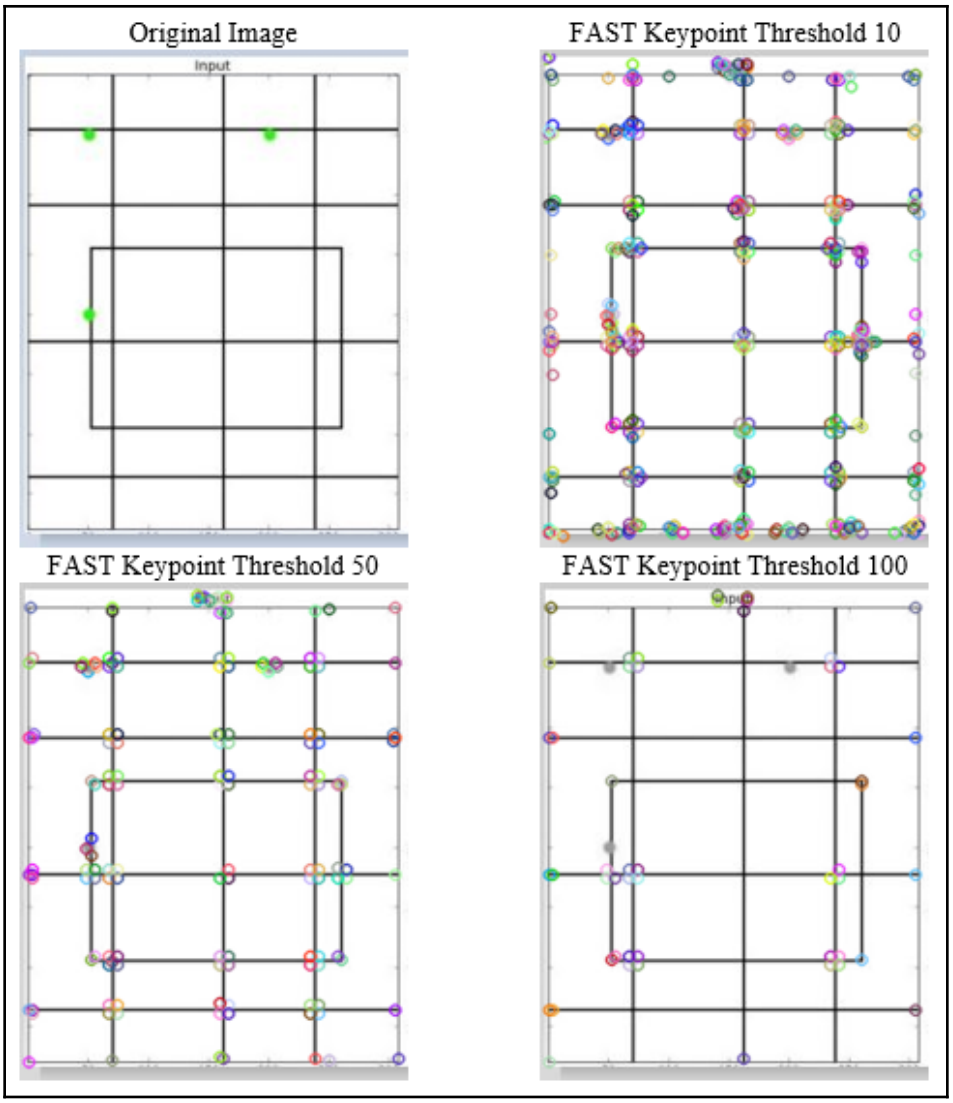


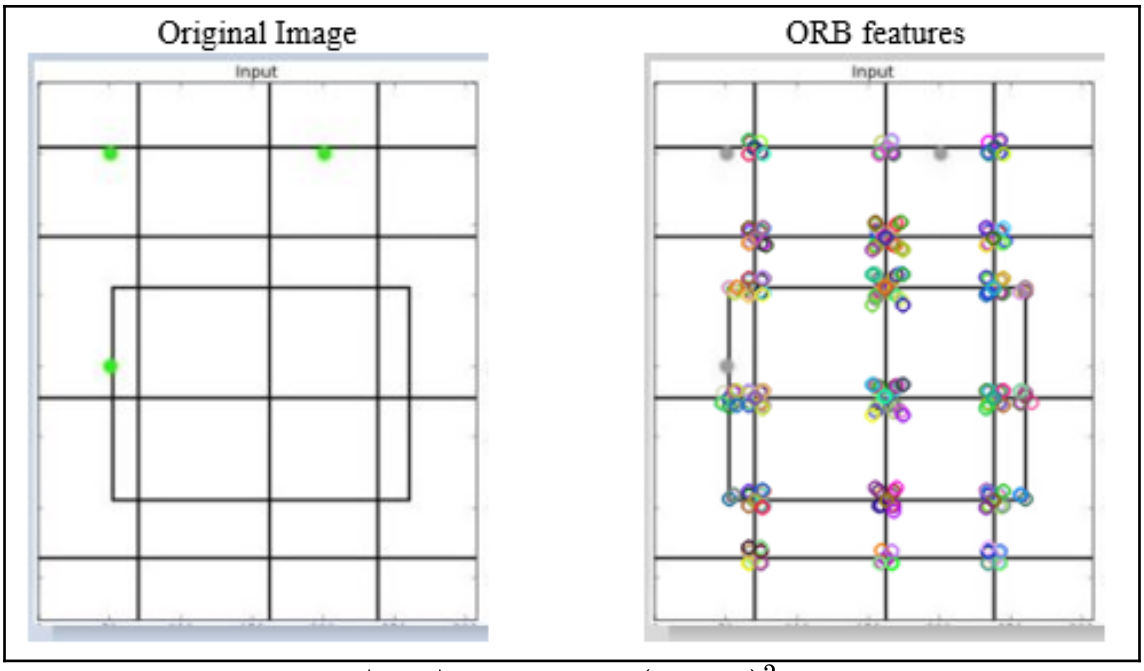




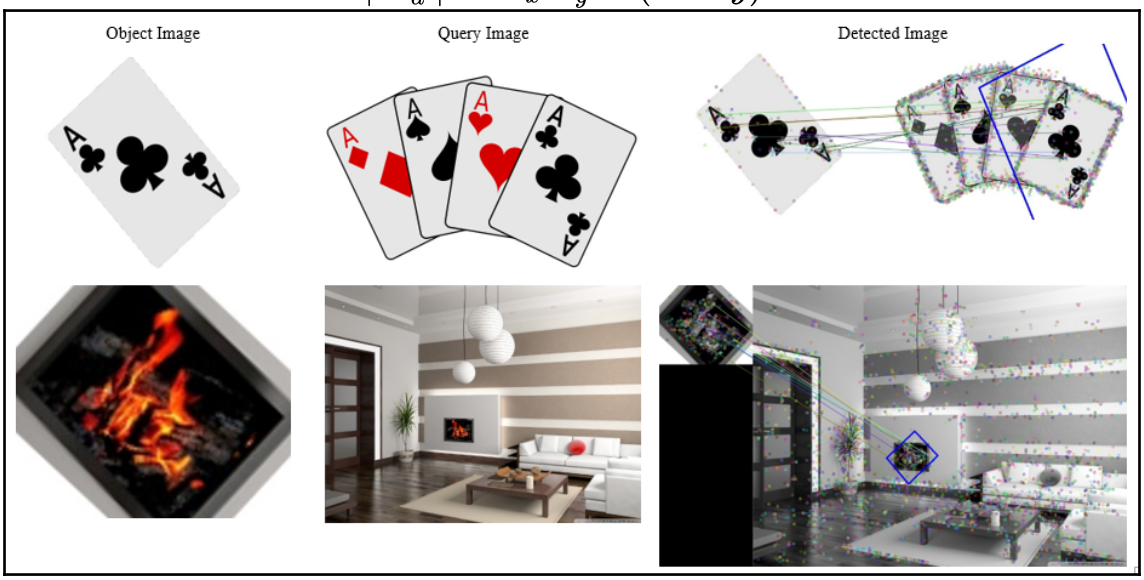

```
bhaumik@bhaumik-Lenovo-ideapad-520-15IKB:~/Desktop/opencv/chapter 7$ ./hough_lin
e
CPU Time : 4.01799 ms
CPU FPS : 248.88
GPU Time : 1.5828 ms
GPU FPS : 631.791
```

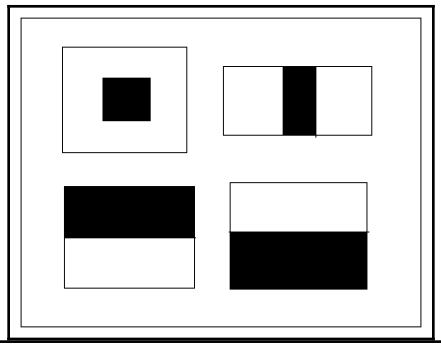

Original Image
Circle Detection


Original Image
Input
FAST Keypoint Threshold 10
FAST Keypoint Threshold 50
FAST Keypoint Threshold 100

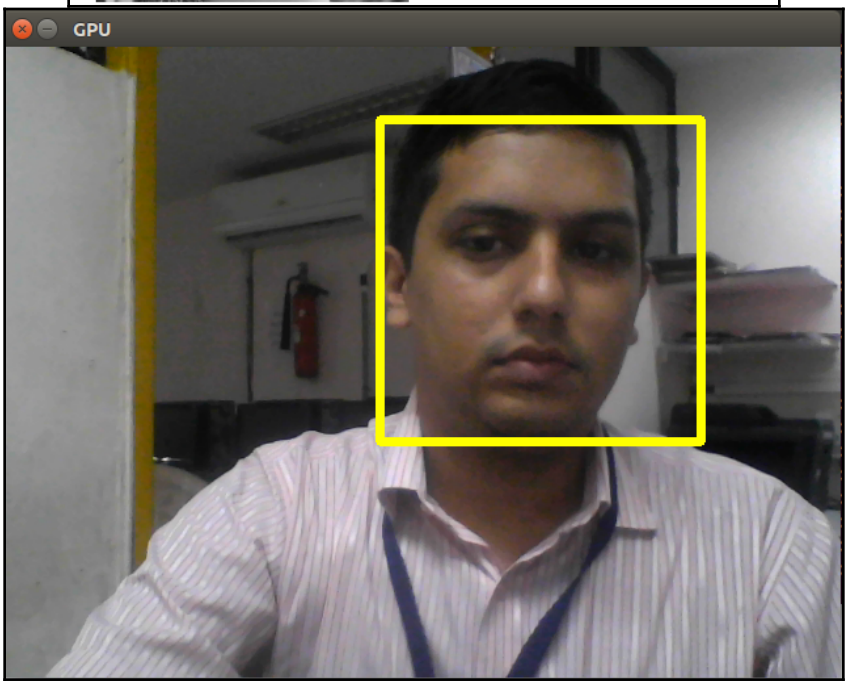


$$|H_a| = D_x D_y - (wDxy)^2$$

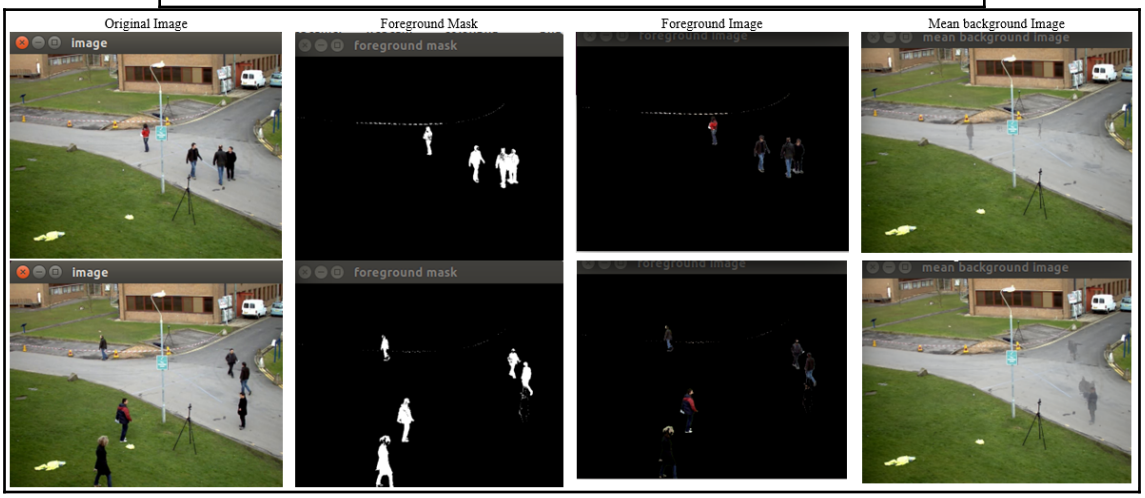


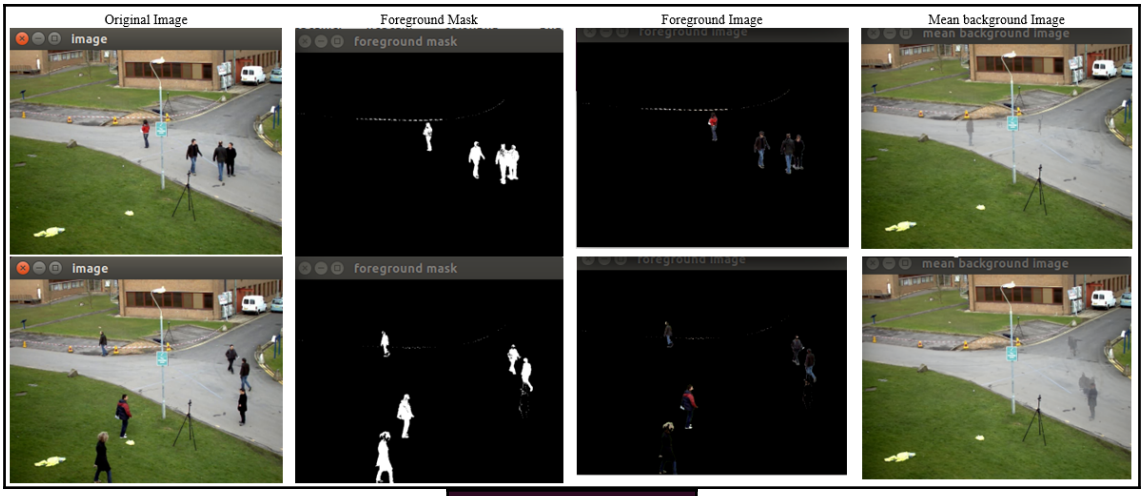



Original Image
Face Detection

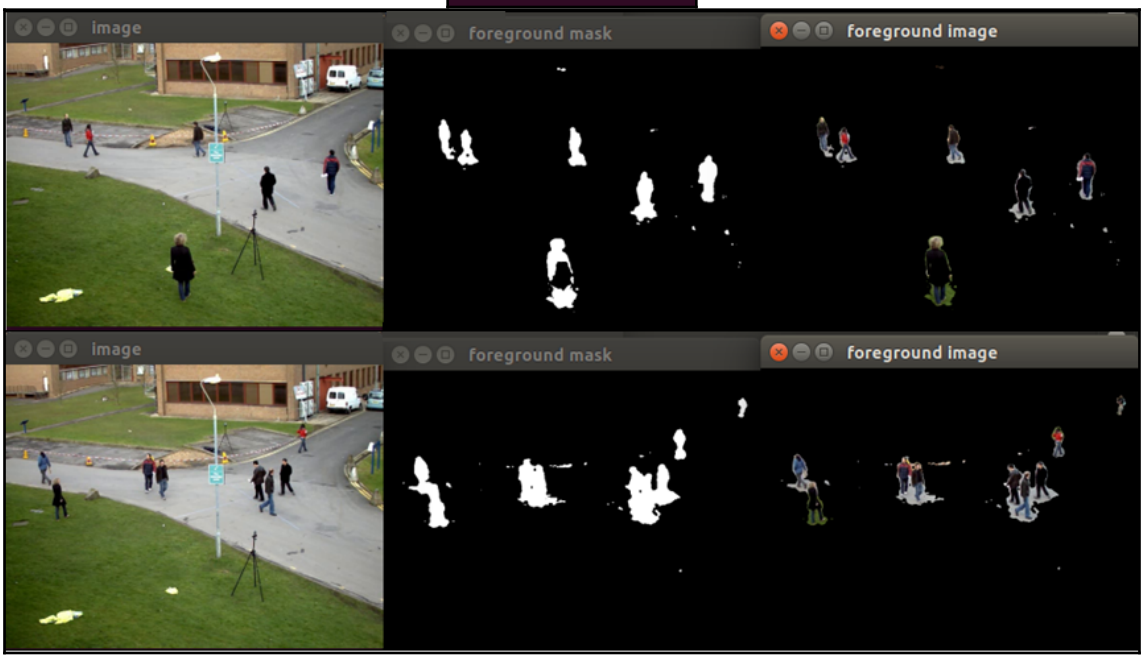

GPU

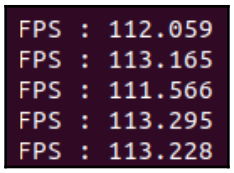

Original Image
Eye Detection


Original Image
Foreground Mask
Foreground Image
Mean background Image
image
foreground mask
mean background image

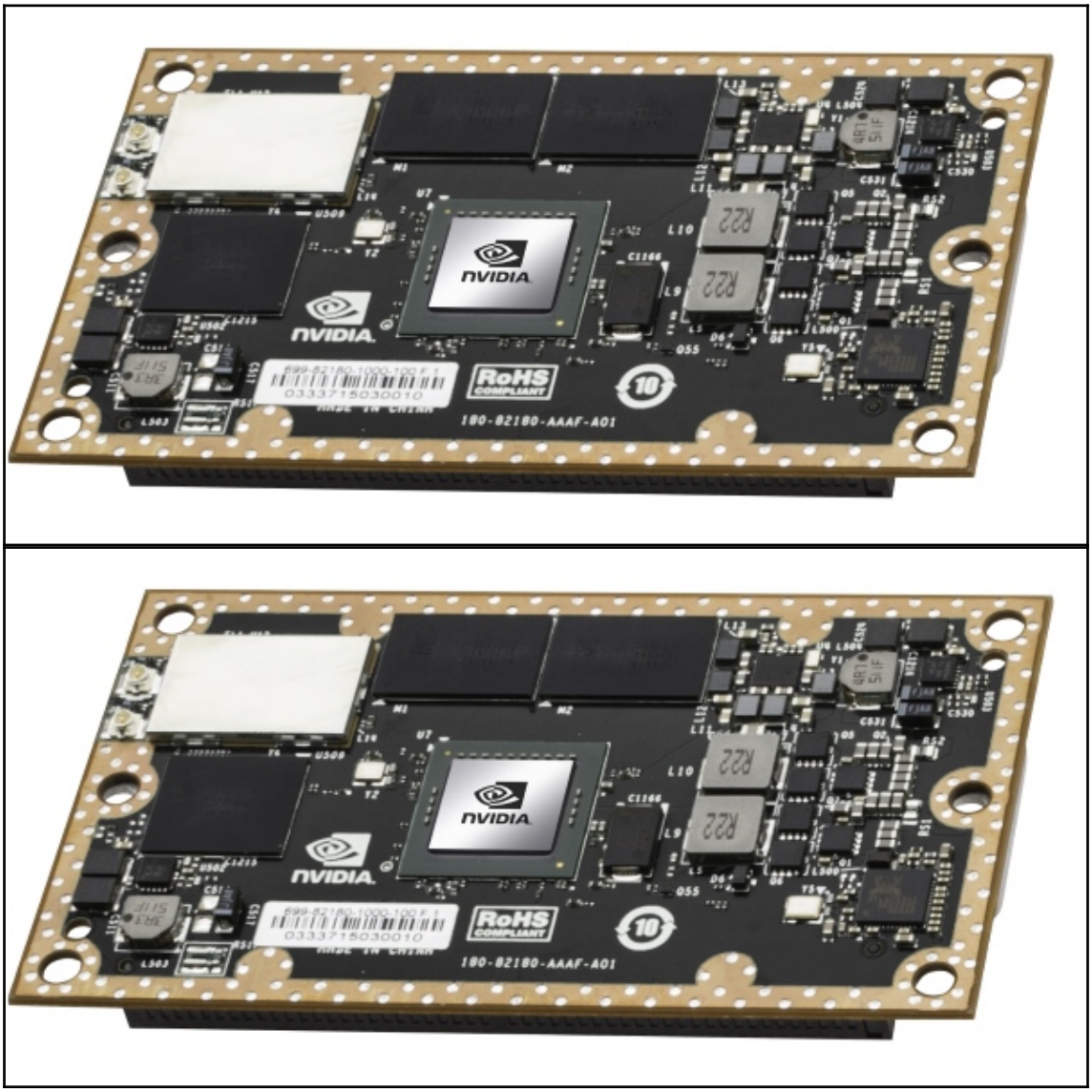


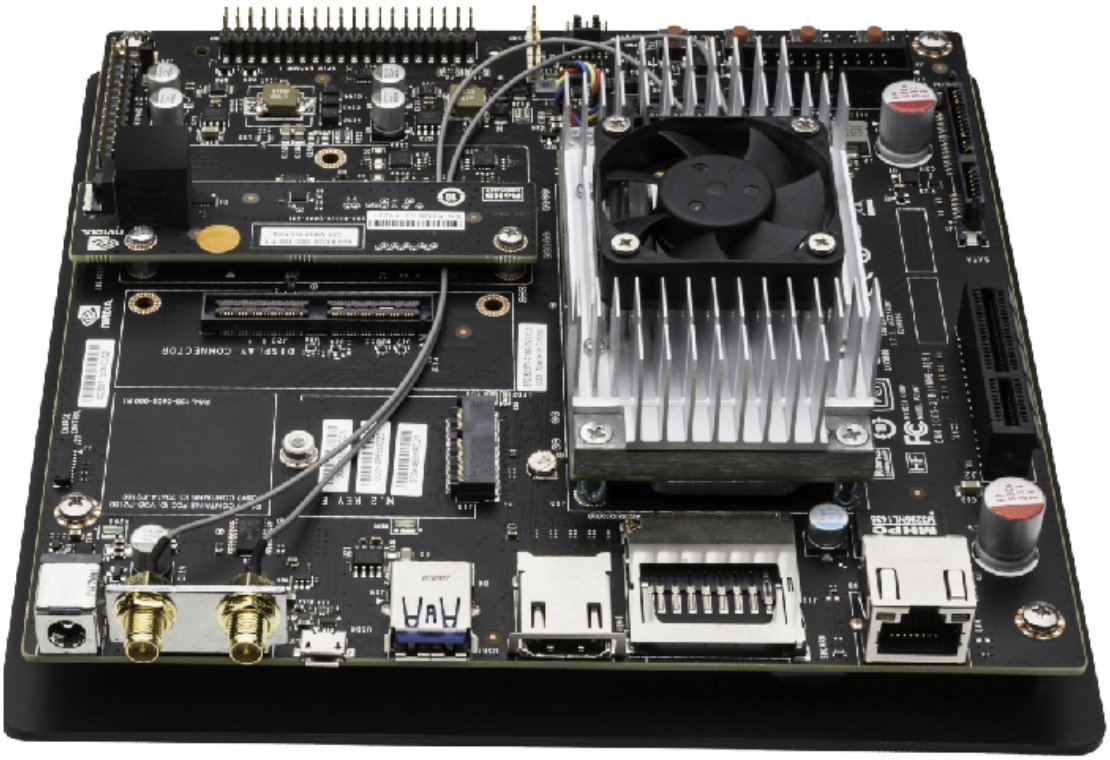

```
FPS : 333.944
FPS : 332.342
FPS : 331.221
FPS : 330.282
```




FPS : 112.059
FPS : 113.165
FPS : 111.566
FPS : 113.295
FPS : 113.228

Chapter 8: Introduction to the Jetson TX1 Development Board and Installing OpenCV on Jetson TX1





JetPack

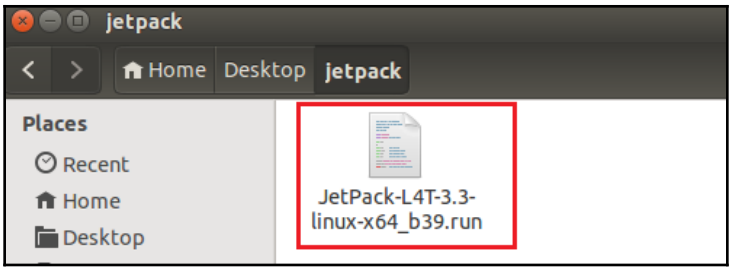
NVIDIA JetPack SDK is the most comprehensive solution for building AI applications. Use the JetPack installer to flash your Jetson Developer Kit with the latest OS image, to install developer tools for both host PC and Developer Kit, and to install the libraries and APIs, samples, and documentation needed to jumpstart your development environment.

JetPack 3.3 with the latest BSPs (**L4T 28.2.1 for Jetson TX2/TX2i** and **L4T 28.2 for Jetson TX1**) is the latest production software release for NVIDIA Jetson TX2, Jetson TX2i, and Jetson TX1. It bundles all the Jetson platform software, including TensorRT, cuDNN, CUDA Toolkit, VisionWorks, GStreamer, and OpenCV, all built on top of L4T with LTS Linux kernel.

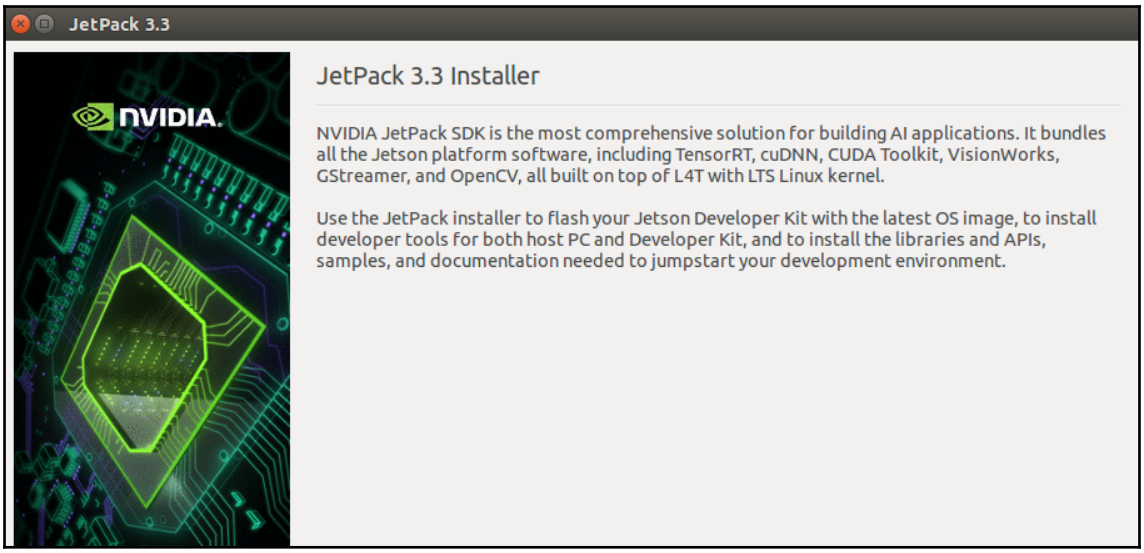
The highlight of this release is TensorRT 4.0, enabling support for TensorFlow's TensorRT integration feature. Additionally, cuDNN has a small point release to support the new TensorRT version, while all other JetPack components remain unchanged from JetPack 3.2.1.

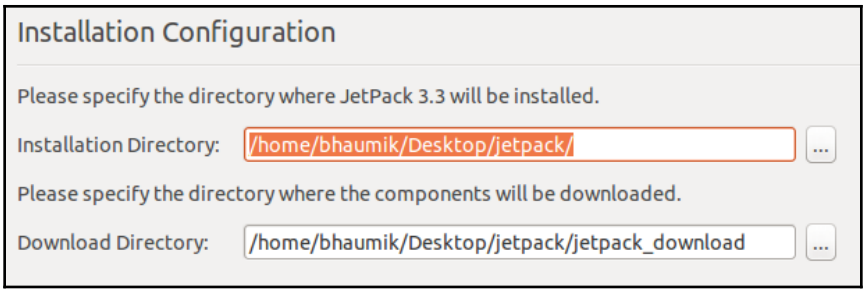
View the full 3.3 Release Notes here.

Download Jetpack

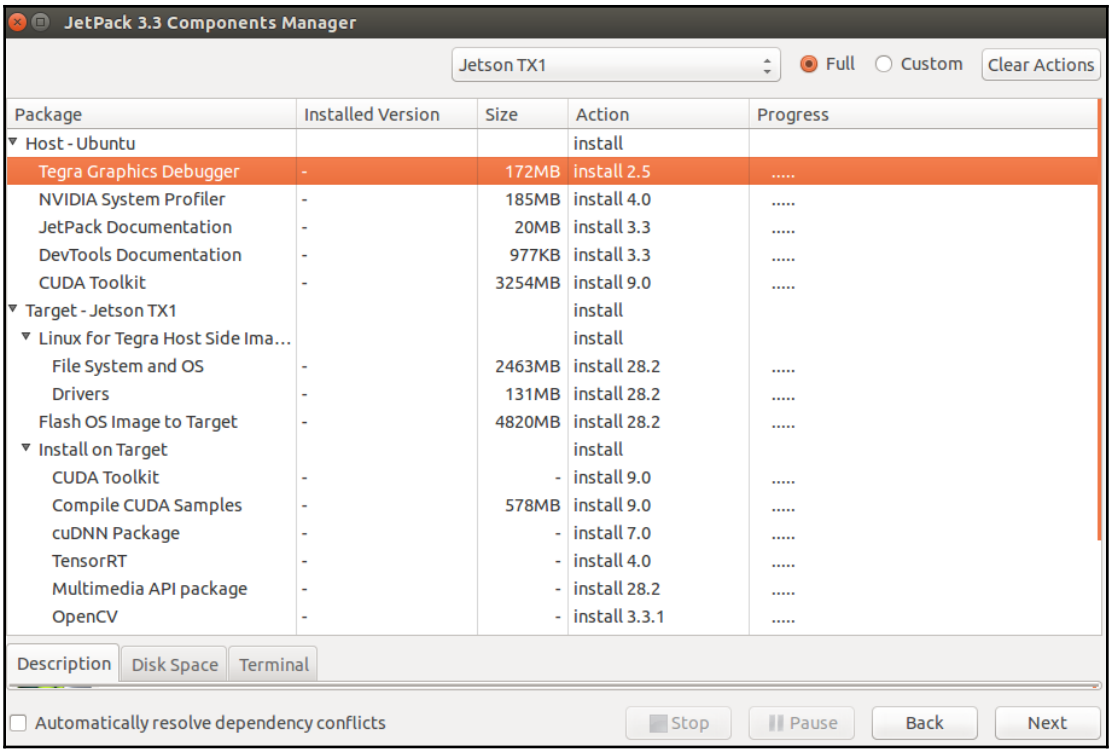


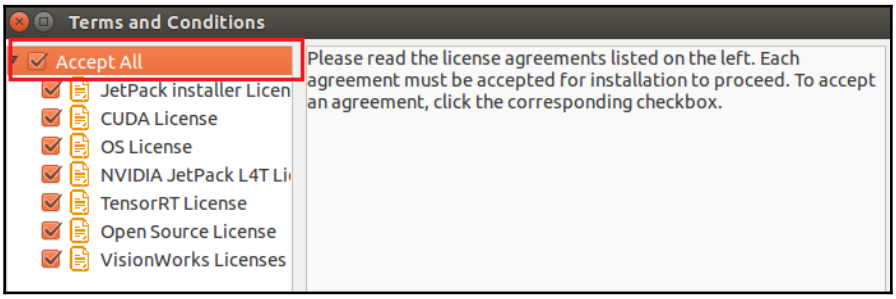

```
bhaumik@bhaumik-VirtualBox:~$ cd Desktop/jetpack
bhaumik@bhaumik-VirtualBox:~/Desktop/jetpack$ chmod +x JetPack-L4T-3.3-linux-x64
_b39.run
bhaumik@bhaumik-VirtualBox:~/Desktop/jetpack$ ./JetPack-L4T-3.3-linux-x64_b39.ru
n
Creating directory _installer
Verifying archive integrity... All good.
Uncompressing JetPack    57%
```

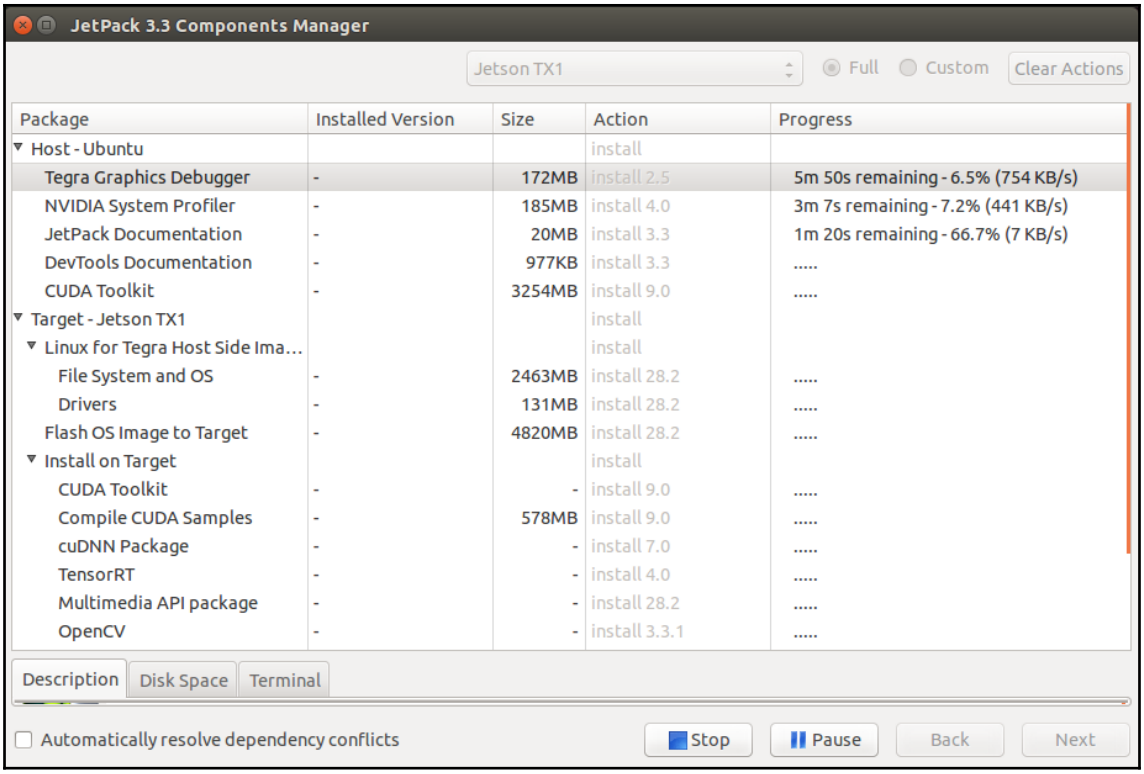


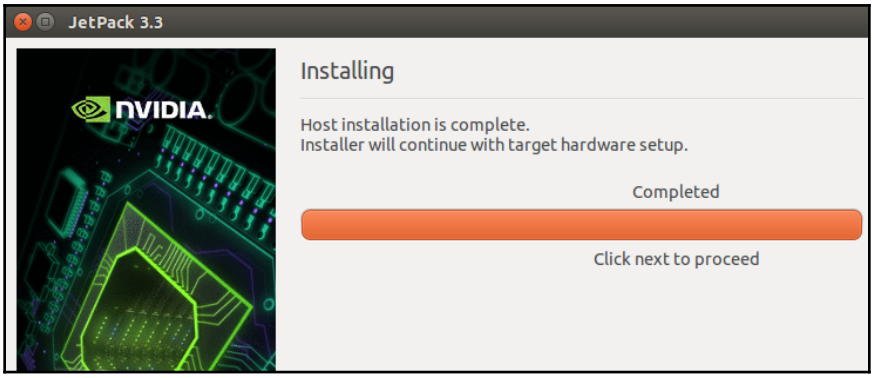

Installation Configuration
Please specify the directory where JetPack 3.3 will be installed.
Installation Directory:
/home/bhaumik/Desktop/jetpack/
...
Please specify the directory where the components will be downloaded.
Download Directory:
/home/bhaumik/Desktop/jetpack/jetpack_download
...

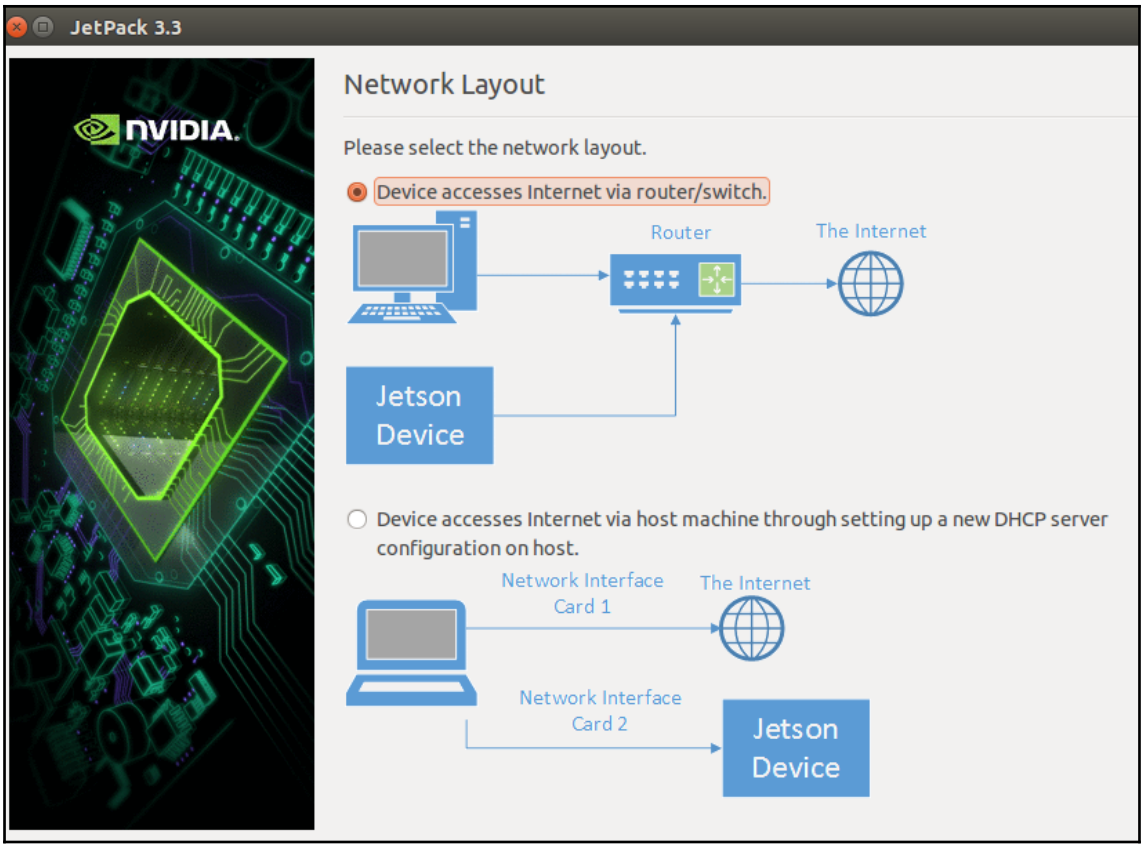

JetPack 3.3
NVIDIA.
Select Development Environment
Please specify the development environment you would like to set up:
Jetson TX2
Jetson TX2i
Jetson TX1

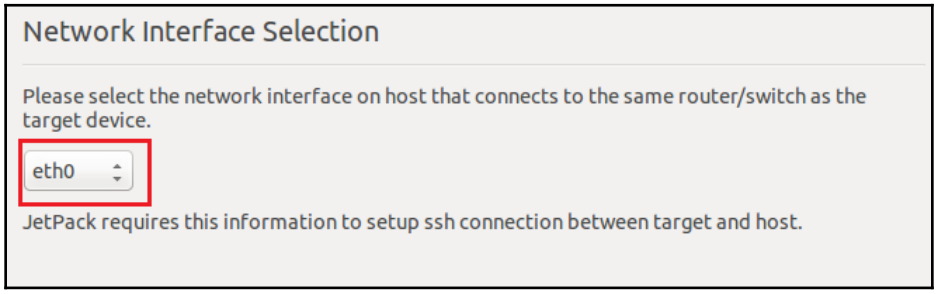

JetPack 3.3 Components Manager
Jetson TX1
Full
Custom
Clear Actions
Package
Installed Version
Size
Action
Progress
Host - Ubuntu
install
Tegra Graphics Debugger
-
172MB
install 2.5
.....
NVIDIA System Profiler
-
185MB
install 4.0
.....
JetPack Documentation
-
20MB
install 3.3
.....
DevTools Documentation
-
977KB
install 3.3
.....
CUDA Toolkit
-
3254MB
install 9.0
.....
Target - Jetson TX1
install
Linux for Tegra Host Side Ima...
install
File System and OS
-
2463MB
install 28.2
.....
Drivers
-
131MB
install 28.2
.....
Flash OS Image to Target
-
4820MB
install 28.2
.....
Install on Target
install
CUDA Toolkit
-
-
install 9.0
.....
Compile CUDA Samples
-
578MB
install 9.0
.....
cuDNN Package
-
-
install 7.0
.....
TensorRT
-
-
install 4.0
.....
Multimedia API package
-
-
install 28.2
.....
OpenCV
-
-
install 3.3.1
.....
Description
Disk Space
Terminal
Automatically resolve dependency conflicts
Stop
Pause
Back
Next

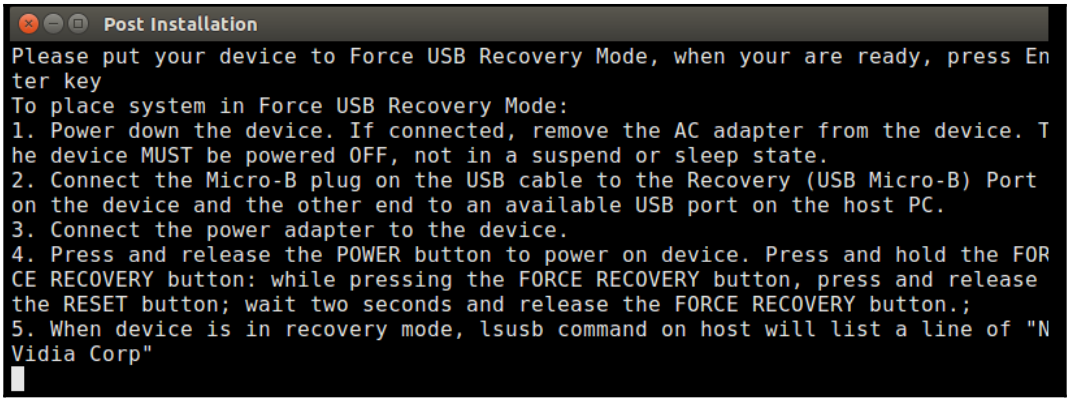

Terms and Conditions
Accept All
JetPack installer Licen
CUDA License
OS License
NVIDIA JetPack L4T Li
TensorRT License
Open Source License
VisionWorks Licenses
Please read the license agreements listed on the left. Each agreement must be accepted for installation to proceed. To accept an agreement, click the corresponding checkbox.

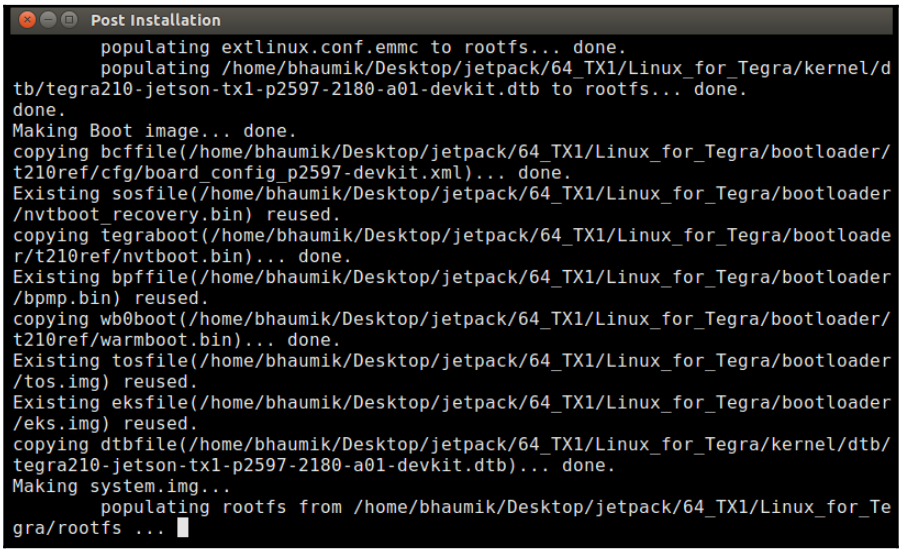

JetPack 3.3 Components Manager
Jetson TX1
Full
Custom
Clear Actions
Package
Installed Version
Size
Action
Progress
Host - Ubuntu
install
Tegra Graphics Debugger
-
172MB
install 2.5
5m 50s remaining - 6.5% (754 KB/s)
NVIDIA System Profiler
-
185MB
install 4.0
3m 7s remaining - 7.2% (441 KB/s)
JetPack Documentation
-
20MB
install 3.3
1m 20s remaining - 66.7% (7 KB/s)
DevTools Documentation
-
977KB
install 3.3
.....
CUDA Toolkit
-
3254MB
install 9.0
.....
Target - Jetson TX1
install
Linux for Tegra Host Side Ima...
install
File System and OS
-
2463MB
install 28.2
.....
Drivers
-
131MB
install 28.2
.....
Flash OS Image to Target
-
4820MB
install 28.2
.....
Install on Target
install
CUDA Toolkit
-
-
install 9.0
.....
Compile CUDA Samples
-
578MB
install 9.0
.....
cuDNN Package
-
-
install 7.0
.....
TensorRT
-
-
install 4.0
.....
Multimedia API package
-
-
install 28.2
.....
OpenCV
-
-
install 3.3.1
.....
Description
Disk Space
Terminal
Automatically resolve dependency conflicts
Stop
Pause
Back
Next


JetPack 3.3
NVIDIA
Installing
Host installation is complete.
Installer will continue with target hardware setup.
Completed
Click next to proceed

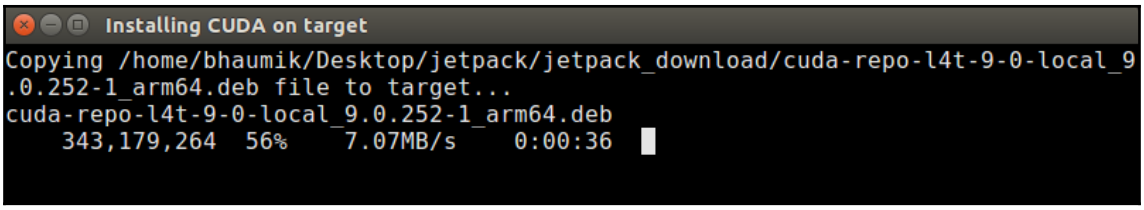

JetPack 3.3
NVIDIA.
Network Layout
Please select the network layout.
Device accesses Internet via router/switch.
Router
The Internet
Jetson Device
Device accesses Internet via host machine through setting up a new DHCP server configuration on host.
Network Interface Card 1
The Internet
Network Interface Card 2
Jetson Device

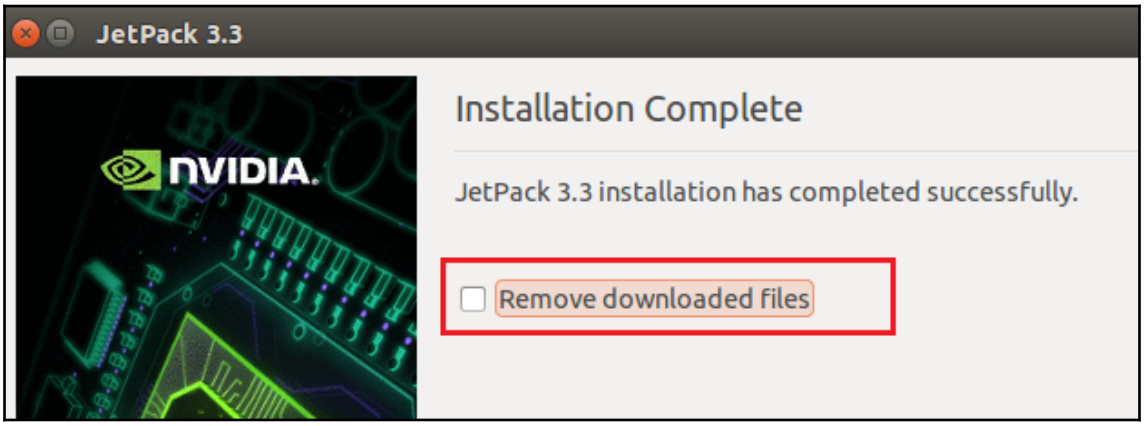

Network Interface Selection
Please select the network interface on host that connects to the same router/switch as the target device.
eth0
JetPack requires this information to setup ssh connection between target and host.


Post Installation
Please put your device to Force USB Recovery Mode, when your are ready, press En
ter key
To place system in Force USB Recovery Mode:
1. Power down the device. If connected, remove the AC adapter from the device. T
he device MUST be powered OFF, not in a suspend or sleep state.
2. Connect the Micro-B plug on the USB cable to the Recovery (USB Micro-B) Port
on the device and the other end to an available USB port on the host PC.
3. Connect the power adapter to the device.
4. Press and release the POWER button to power on device. Press and hold the FOR
CE RECOVERY button: while pressing the FORCE RECOVERY button, press and release
the RESET button; wait two seconds and release the FORCE RECOVERY button.;
5. When device is in recovery mode, lsusb command on host will list a line of "N
Vidia Corp"


Post Installation
populating extlinux.conf.emmc to rootfs... done.
populating /home/bhaumik/Desktop/jetpack/64_TX1/Linux_for_Tegra/kernel/d
tb/tegra210-jetson-tx1-p2597-2180-a01-devkit.dtb to rootfs... done.
done.
Making Boot image... done.
copying bcffile(/home/bhaumik/Desktop/jetpack/64_TX1/Linux_for_Tegra/bootloader/
t210ref/cfg/board_config_p2597-devkit.xml)... done.
Existing sosfile(/home/bhaumik/Desktop/jetpack/64_TX1/Linux_for_Tegra/bootloader
/nvtboot_recovery.bin) reused.
copying tegraboot(/home/bhaumik/Desktop/jetpack/64_TX1/Linux_for_Tegra/bootloade
r/t210ref/nvtboot.bin)... done.
Existing bpffile(/home/bhaumik/Desktop/jetpack/64_TX1/Linux_for_Tegra/bootloader
/bpmp.bin) reused.
copying wb0boot(/home/bhaumik/Desktop/jetpack/64_TX1/Linux_for_Tegra/bootloader/
t210ref/warmboot.bin)... done.
Existing tosfile(/home/bhaumik/Desktop/jetpack/64_TX1/Linux_for_Tegra/bootloader
/tos.img) reused.
Existing eksfile(/home/bhaumik/Desktop/jetpack/64_TX1/Linux_for_Tegra/bootloader
/eks.img) reused.
copying dtbfile(/home/bhaumik/Desktop/jetpack/64_TX1/Linux_for_Tegra/kernel/dtb/
tegra210-jetson-tx1-p2597-2180-a01-devkit.dtb)... done.
Making system.img...
populating rootfs from /home/bhaumik/Desktop/jetpack/64_TX1/Linux_for_Te
gra/rootfs ...


JetPack 3.3
NVIDIA
Post Installation Jetson TX1
Following actions will be performed at this stage.
- Push and install 64Bit CUDA on target
- Push and install TensorRT on target
- Push and install 64Bit OpenCV on target
- Push and install 64Bit cuDNN on target
- Push and install 64Bit VisionWorks on target
- Push and install 64Bit VisionWorks SFM on target
- Push and install 64Bit VisionWorks Tracking on target
- Cross-compile 64Bit CUDA samples and push to target
- Push and install MMAPI on target


Installing CUDA on target
Copying /home/bhaumik/Desktop/jetpack/jetpack_download/cuda-repo-l4t-9-0-local_9
.0.252-1_arm64.deb file to target...
cuda-repo-l4t-9-0-local_9.0.252-1_arm64.deb
343,179,264 56% 7.07MB/s 0:00:36


JetPack 3.3
NVIDIA.
Installation Complete
JetPack 3.3 installation has completed successfully.
Remove downloaded files

Chapter 9: Deploying Computer Vision Applications on Jetson TX1

```
ubuntu@tegra-ubuntu:~/Desktop/opencv/Chapter  9$ nvcc kernel.cu -o device
ubuntu@tegra-ubuntu:~/Desktop/opencv/Chapter  9$ ./device
./device Starting...

 CUDA Device Query (Runtime API) version (CUDART static linking)

Detected 1 CUDA Capable device(s)

Device 0: "NVIDIA Tegra X1"
  CUDA Driver Version / Runtime Version          9.0 / 9.0
  CUDA Capability Major/Minor version number:    5.3
  Total amount of global memory:                 3984 MBytes (4177342464 bytes)
  ( 2) Multiprocessors  GPU Max Clock rate:                            998 MHz (1.00 GHz)
  Memory Clock rate:                             13 Mhz
  Memory Bus Width:                              64-bit
  L2 Cache Size:                                 262144 bytes
  Maximum Texture Dimension Size (x,y,z)         1D=(65536), 2D=(65536, 65536), 3D=(4096, 4096, 4096)
  Maximum Layered 1D Texture Size, (num) layers  1D=(16384), 2048 layers
  Maximum Layered 2D Texture Size, (num) layers  2D=(16384, 16384), 2048 layers
  Total amount of constant memory:               65536 bytes
  Total amount of shared memory per block:       49152 bytes
  Total number of registers available per block: 32768
  Warp size:                                     32
  Maximum number of threads per multiprocessor:  2048
  Maximum number of threads per block:           1024
  Max dimension size of a thread block (x,y,z): (1024, 1024, 64)
  Max dimension size of a grid size    (x,y,z): (2147483647, 65535, 65535)
  Maximum memory pitch:                          2147483647 bytes
  Texture alignment:                             512 bytes
  Concurrent copy and kernel execution:          Yes with 1 copy engine(s)
  Run time limit on kernels:                     Yes
  Integrated GPU sharing Host Memory:            Yes
  Support host page-locked memory mapping:       Yes
  Alignment requirement for Surfaces:            Yes
  Device has ECC support:                        Disabled
  Device supports Unified Addressing (UVA):      Yes
  Supports Cooperative Kernel Launch:            No
  Supports MultiDevice Co-op Kernel Launch:      No
  Device PCI Domain ID / Bus ID / location ID:   0 / 0 / 0
  Compute Mode:
     < Default (multiple host threads can use ::cudaSetDevice() with device simultaneously) >
ubuntu@tegra-ubuntu:~/Desktop/opencv/Chapter  9$
```

```
ubuntu@tegra-ubuntu:~/Desktop/opencv/Chapter  9$ nvcc 01_performance_cuda_events
.cu -o gpu_add
ubuntu@tegra-ubuntu:~/Desktop/opencv/Chapter  9$ ./gpu_add
Time to add 50000 numbers: 3.4 ms
Vector addition on GPU
GPU has computed Sum Correctly
```

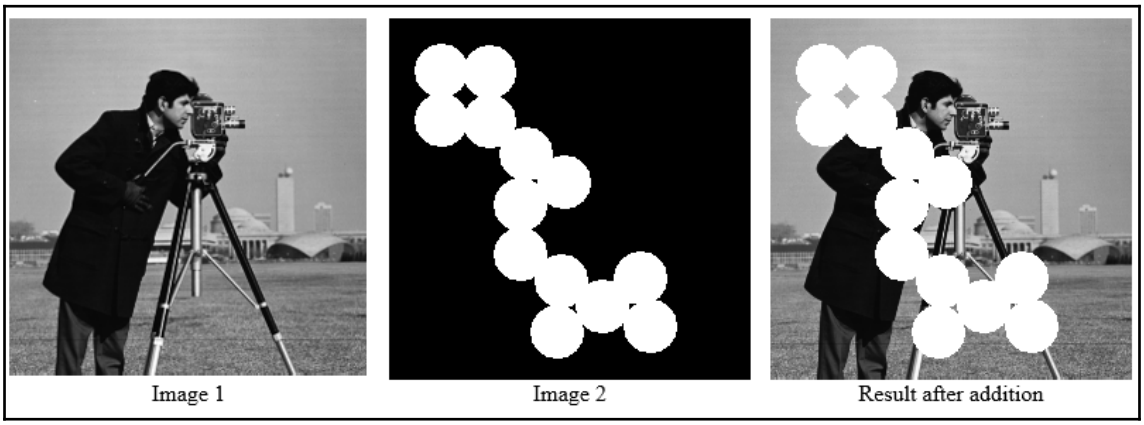


Image 1 | Image 2 | Result after addition

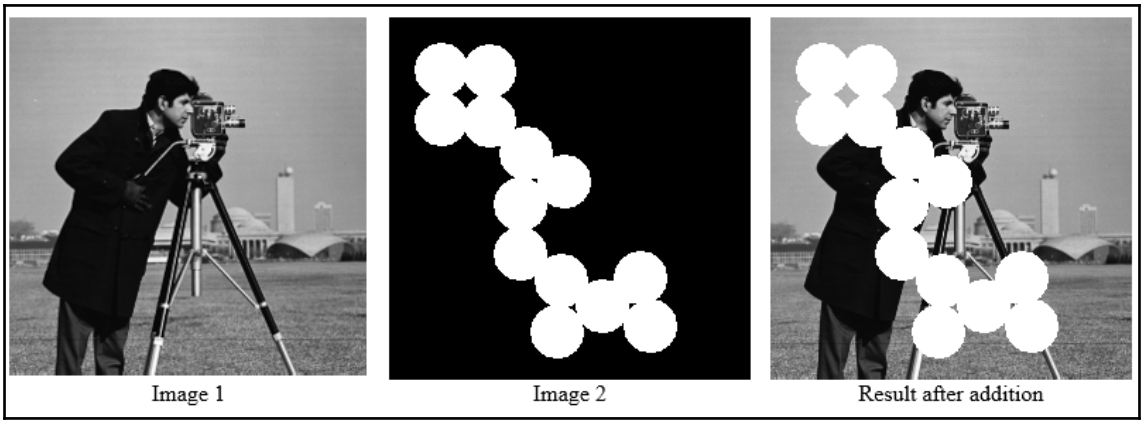
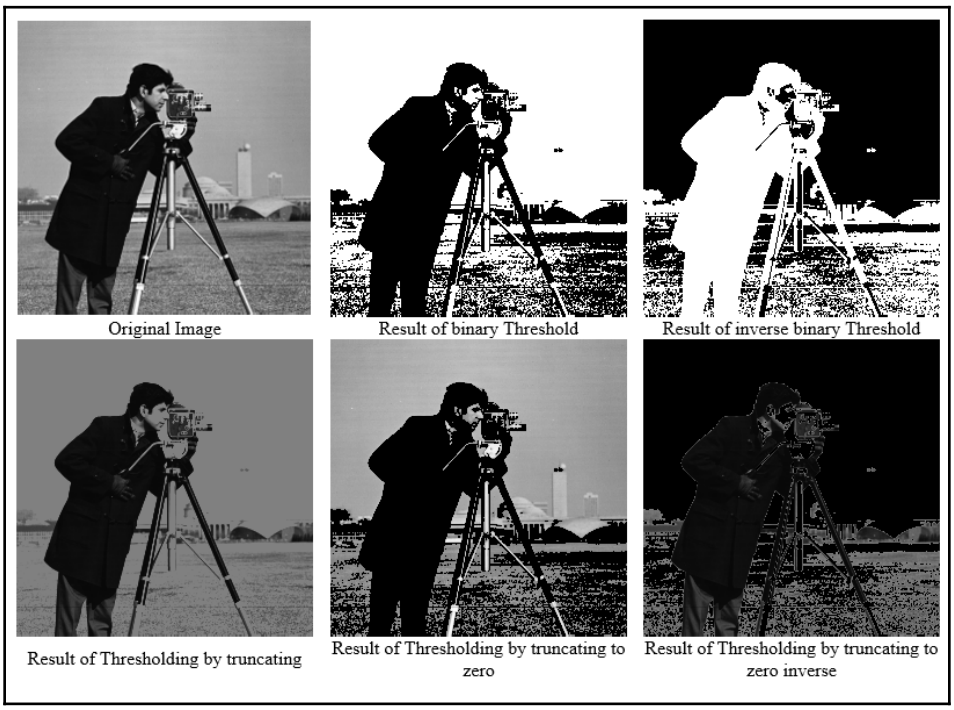
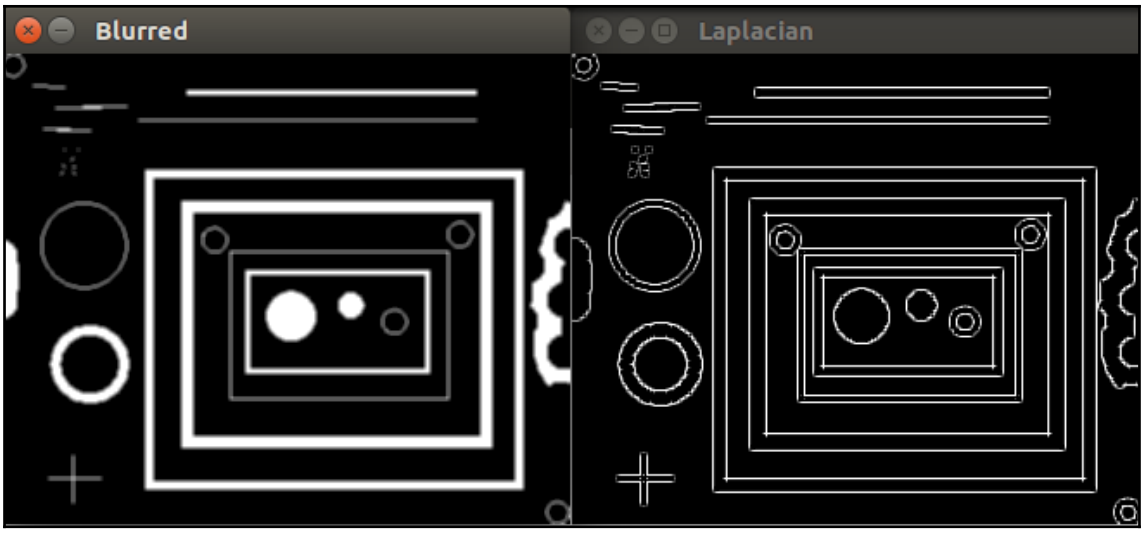


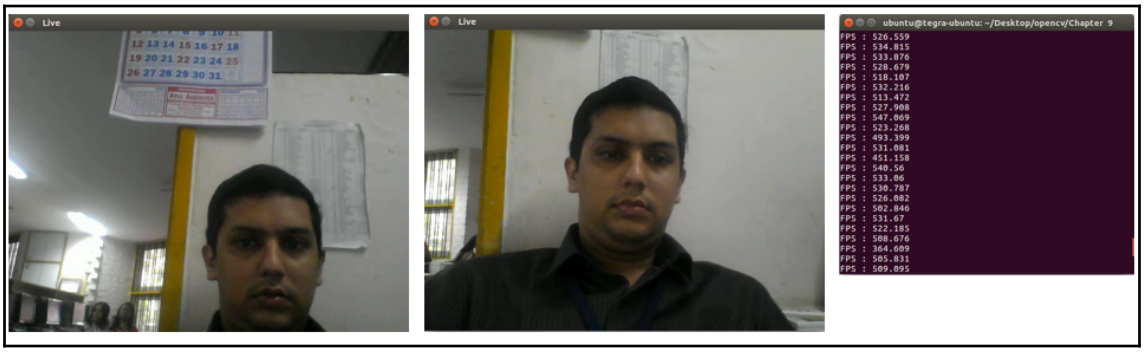
Image 1 | Image 2 | Result after addition

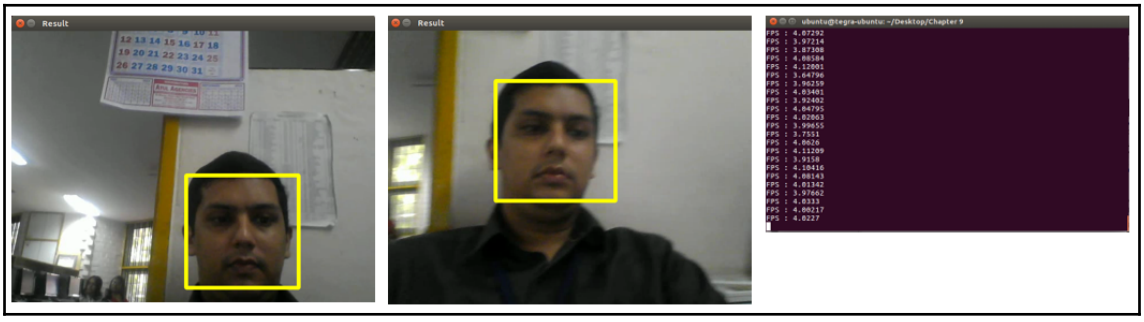
```
ubuntu@tegra-ubuntu:~/Desktop/opencv/Chapter  9$ ./image_add
Performance of Addition on Jetson TX!:
Time: 0.000262104
FPS: 3815.28
```




```
Performance of Thresholding
Time: 0.000326671
FPS: 3061.18
```


Blurred
Laplacian


Live
Live
ubuntu@tegra-ubuntu: ~/Desktop/opencv/Chapter 9
FPS : 526.559
FPS : 534.815
FPS : 533.876
FPS : 528.679
FPS : 518.107
FPS : 532.216
FPS : 513.472
FPS : 527.908
FPS : 547.069
FPS : 523.268
FPS : 493.399
FPS : 531.081
FPS : 451.158
FPS : 540.56
FPS : 533.86
FPS : 530.787
FPS : 526.882
FPS : 502.846
FPS : 531.67
FPS : 522.185
FPS : 508.676
FPS : 364.609
FPS : 505.831
FPS : 509.095


Result
Result
ubuntu@tegra-ubuntu: ~/Desktop/Chapter 9
FPS : 4.07292
FPS : 3.97214
FPS : 3.87308
FPS : 4.08184
FPS : 4.12601
FPS : 3.64796
FPS : 3.96219
FPS : 4.03401
FPS : 3.92402
FPS : 4.04795
FPS : 4.02863
FPS : 3.99655
FPS : 3.7551
FPS : 4.0626
FPS : 4.11209
FPS : 3.9158
FPS : 4.10416
FPS : 4.08143
FPS : 4.01342
FPS : 3.97662
FPS : 4.0333
FPS : 4.00217
FPS : 4.0227


Original Image
Eye Detection


image
foreground mask
foreground image
mean background image
image
foreground mask
foreground image
mean background image
image
foreground mask
foreground image
mean background image

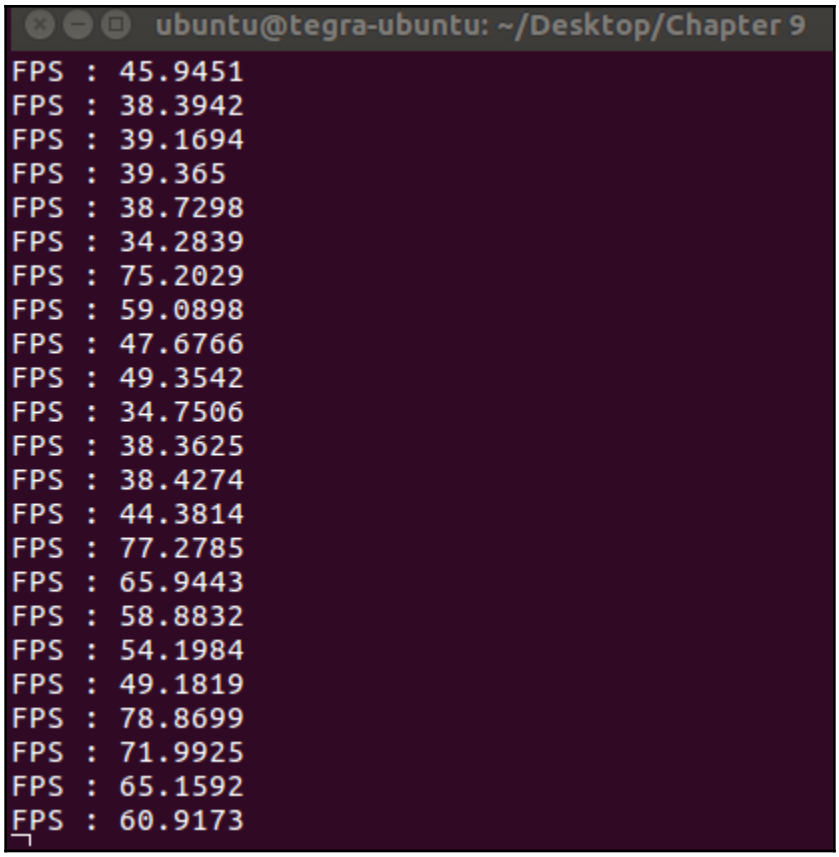
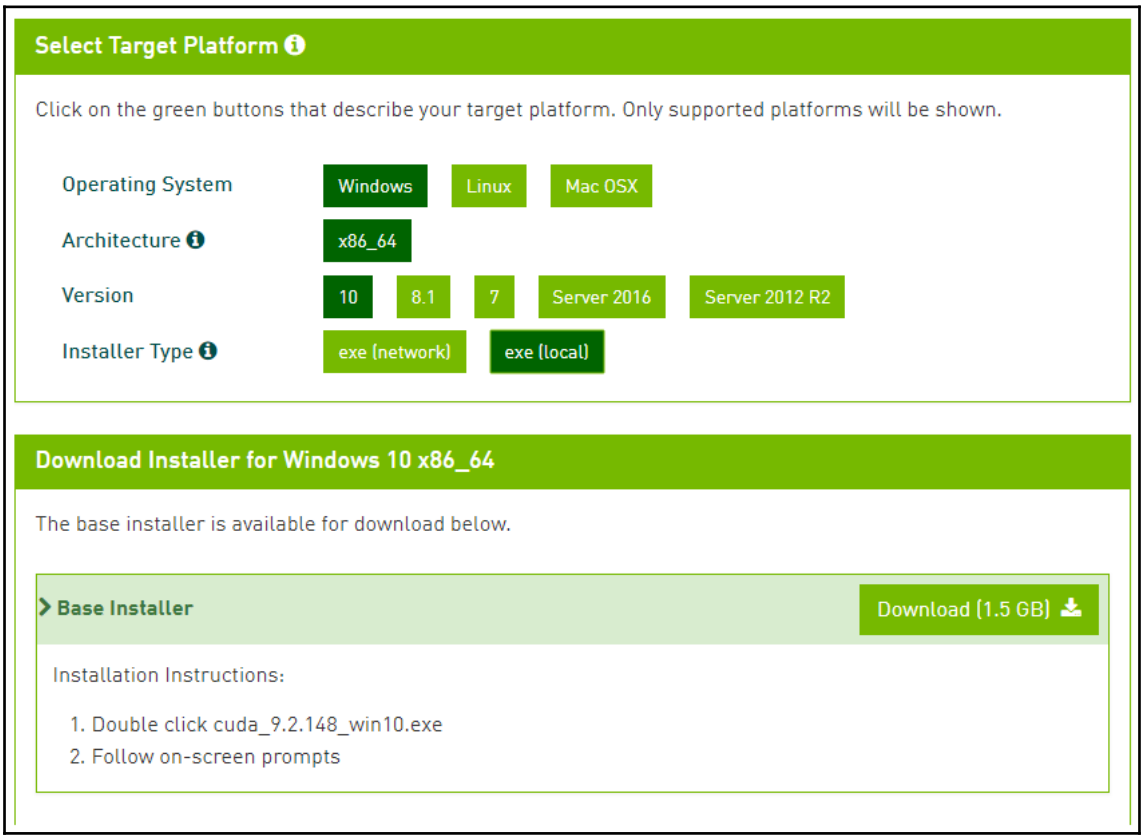
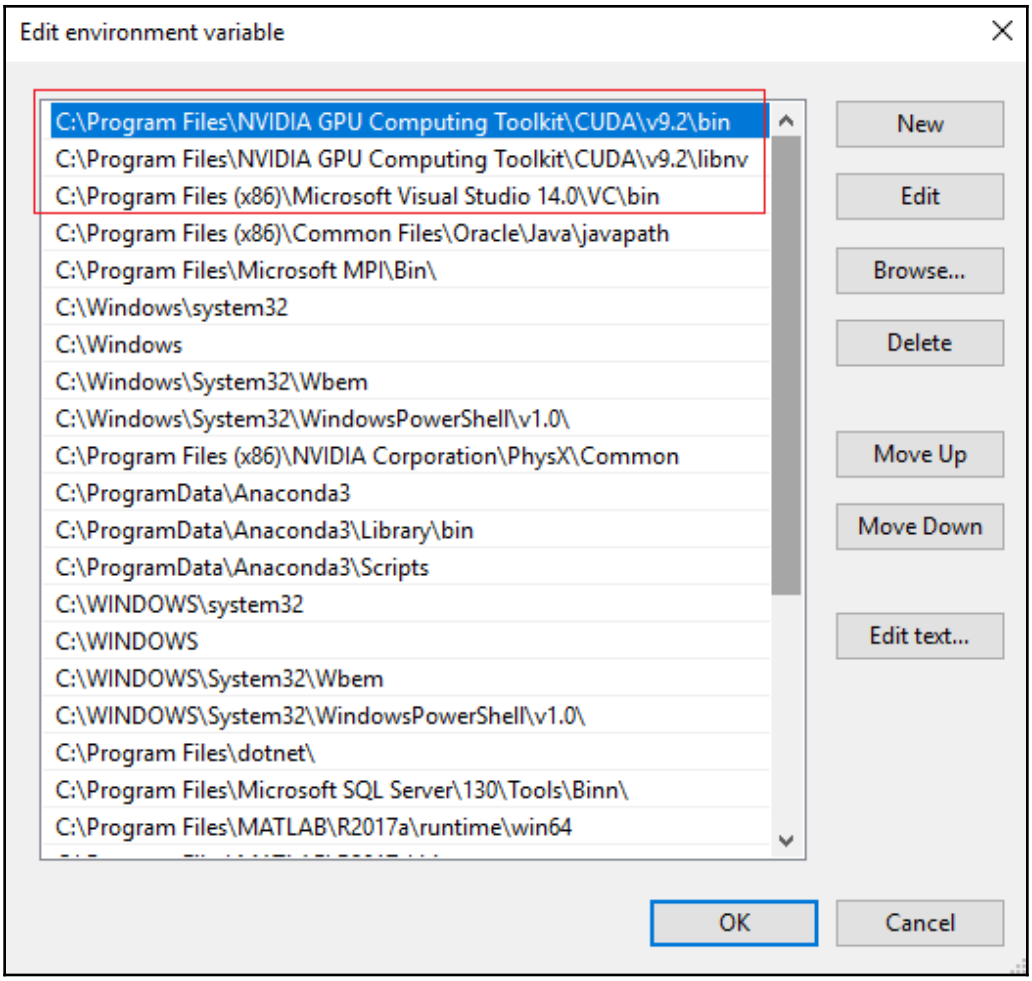
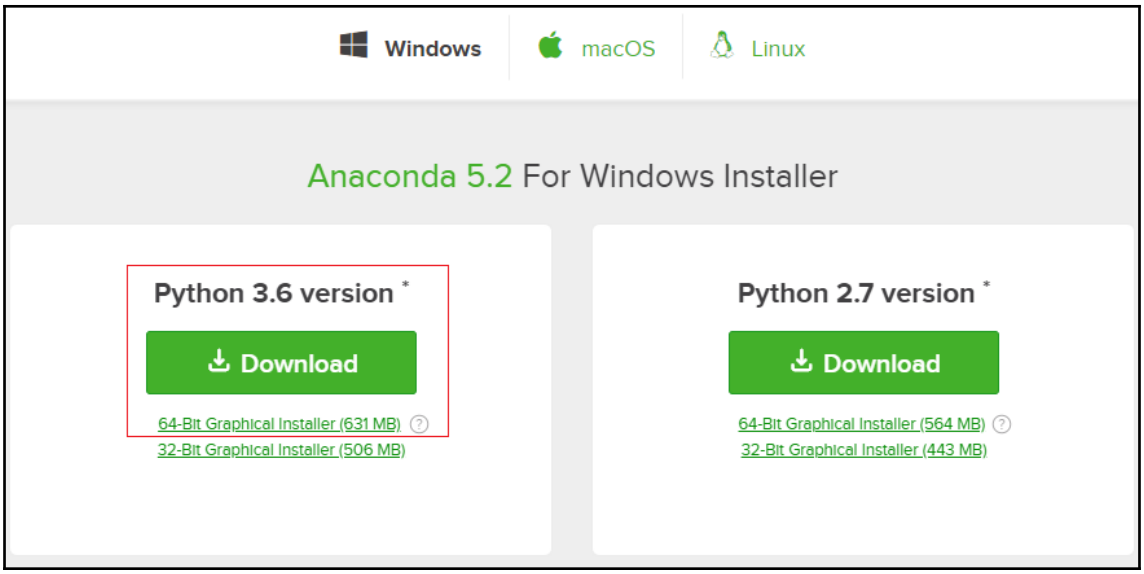

ubuntu@tegra-ubuntu: ~/Desktop/Chapter 9
FPS : 45.9451
FPS : 38.3942
FPS : 39.1694
FPS : 39.365
FPS : 38.7298
FPS : 34.2839
FPS : 75.2029
FPS : 59.0898
FPS : 47.6766
FPS : 49.3542
FPS : 34.7506
FPS : 38.3625
FPS : 38.4274
FPS : 44.3814
FPS : 77.2785
FPS : 65.9443
FPS : 58.8832
FPS : 54.1984
FPS : 49.1819
FPS : 78.8699
FPS : 71.9925
FPS : 65.1592
FPS : 60.9173


Image read in Python

Chapter 10: Getting Started with PyCUDA

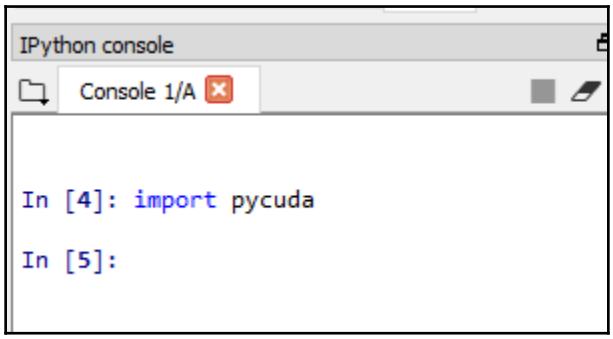


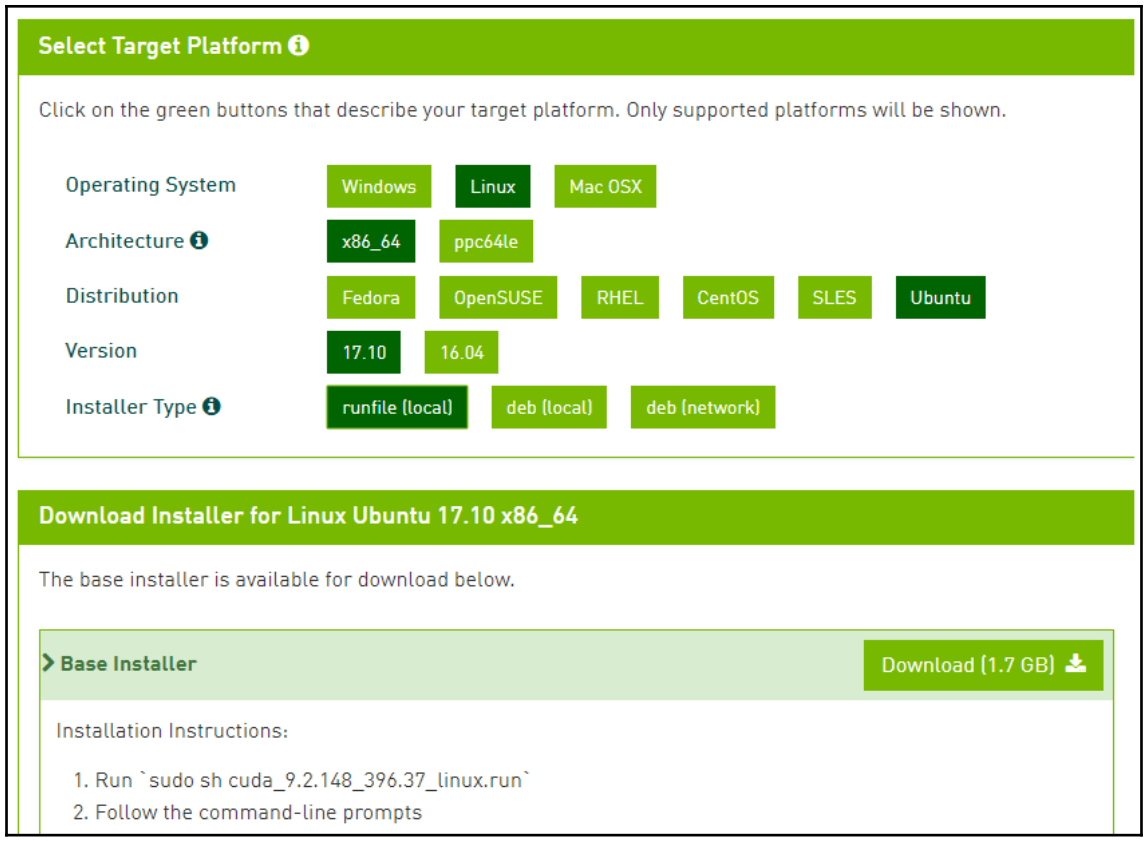

Edit environment variable
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.2\bin
C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v9.2\libnv
C:\Program Files (x86)\Microsoft Visual Studio 14.0\VC\bin
C:\Program Files (x86)\Common Files\Oracle\Java\javapath
C:\Program Files\Microsoft MPI\Bin\
C:\Windows\system32
C:\Windows
C:\Windows\System32\Wbem
C:\Windows\System32\WindowsPowerShell\v1.0\
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\ProgramData\Anaconda3
C:\ProgramData\Anaconda3\Library\bin
C:\ProgramData\Anaconda3\Scripts
C:\WINDOWS\system32
C:\WINDOWS
C:\WINDOWS\System32\Wbem
C:\WINDOWS\System32\WindowsPowerShell\v1.0\
C:\Program Files\dotnet\
C:\Program Files\Microsoft SQL Server\130\Tools\Binn\
C:\Program Files\MATLAB\R2017a\runtime\win64
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel

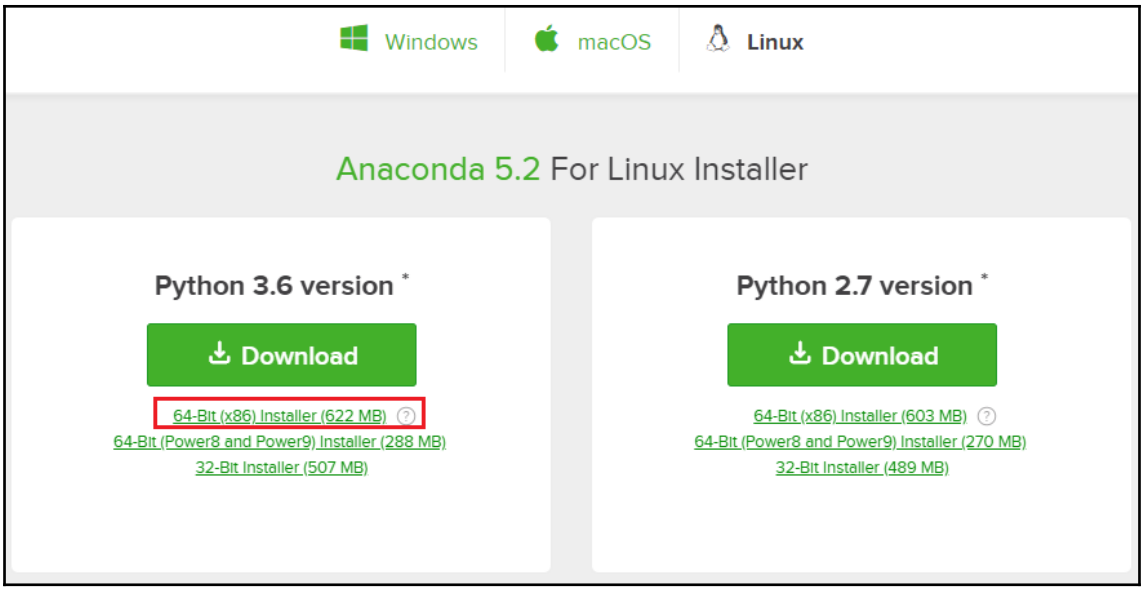


pycuda-2017.1.1+cuda9185-cp34-cp34m-win32.whl
pycuda-2017.1.1+cuda9185-cp34-cp34m-win_amd64.whl
pycuda-2017.1.1+cuda9185-cp35-cp35m-win32.whl
pycuda-2017.1.1+cuda9185-cp35-cp35m-win_amd64.whl
pycuda-2017.1.1+cuda9185-cp36-cp36m-win32.whl
pycuda-2017.1.1+cuda9185-cp36-cp36m-win_amd64.whl
pycuda-2018.1+cuda92148-cp27-cp27m-win32.whl
pycuda-2018.1+cuda92148-cp27-cp27m-win_amd64.whl
pycuda-2018.1+cuda92148-cp34-cp34m-win32.whl
pycuda-2018.1+cuda92148-cp34-cp34m-win_amd64.whl
pycuda-2018.1+cuda92148-cp35-cp35m-win32.whl
pycuda-2018.1+cuda92148-cp35-cp35m-win_amd64.whl
pycuda-2018.1+cuda92148-cp36-cp36m-win32.whl
pycuda-2018.1+cuda92148-cp36-cp36m-win_amd64.whl
pycuda-2018.1+cuda92148-cp37-cp37m-win32.whl
pycuda-2018.1+cuda92148-cp37-cp37m-win_amd64.whl

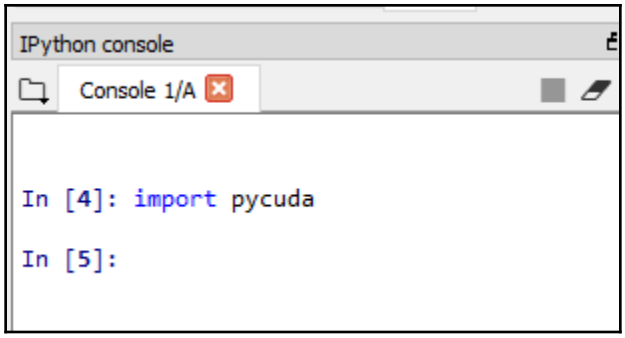
```
PS C:\Users\bhaum\Downloads> pip install .\pycuda-2017.1.1+cuda92148-cp36-cp36m-win_amd64.whl
Processing c:\users\bhaum\downloads\pycuda-2017.1.1+cuda92148-cp36-cp36m-win_amd64.whl
Requirement already satisfied: pytest>=2 in c:\programdata\anaconda3\lib\site-packages (from pycuda==2017.1.1+cuda92148)

Requirement already satisfied: pytools>=2011.2 in c:\programdata\anaconda3\lib\site-packages (from pycuda==2017.1.1+cuda
92148)
Requirement already satisfied: decorator>=3.2.0 in c:\programdata\anaconda3\lib\site-packages (from pycuda==2017.1.1+cud
a92148)
Requirement already satisfied: appdirs>=1.4.0 in c:\programdata\anaconda3\lib\site-packages (from pycuda==2017.1.1+cuda9
2148)
Requirement already satisfied: py>=1.5.0 in c:\programdata\anaconda3\lib\site-packages (from pytest>=2->pycuda==2017.1.1
+cuda92148)
Requirement already satisfied: six>=1.10.0 in c:\programdata\anaconda3\lib\site-packages (from pytest>=2->pycuda==2017.1
.1+cuda92148)
Requirement already satisfied: setuptools in c:\programdata\anaconda3\lib\site-packages (from pytest>=2->pycuda==2017.1.
1+cuda92148)
Requirement already satisfied: attrs>=17.2.0 in c:\programdata\anaconda3\lib\site-packages (from pytest>=2->pycuda==2017
.1.1+cuda92148)
Requirement already satisfied: pluggy<0.7,>=0.5 in c:\programdata\anaconda3\lib\site-packages (from pytest>=2->pycuda==2
017.1.1+cuda92148)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-packages (from pytest>=2->pycuda==2017.1.1+
cuda92148)
Requirement already satisfied: numpy>=1.6.0 in c:\programdata\anaconda3\lib\site-packages (from pytools>=2011.2->pycuda=
=2017.1.1+cuda92148)
Installing collected packages: pycuda
  Found existing installation: pycuda 2017.1.1+cuda9185
    Uninstalling pycuda-2017.1.1+cuda9185:
      Successfully uninstalled pycuda-2017.1.1+cuda9185
Successfully installed pycuda-2017.1.1+cuda92148
You are using pip version 9.0.1, however version 18.0 is available.
You should consider upgrading via the 'python -m pip install --upgrade pip' command.
```




Select Target Platform
Click on the green buttons that describe your target platform. Only supported platforms will be shown.
Operating System
Windows
Linux
Mac OSX
Architecture
x86_64
ppc64le
Distribution
Fedora
OpenSUSE
RHEL
CentOS
SLES
Ubuntu
Version
17.10
16.04
Installer Type
runfile (local)
deb (local)
deb (network)
Download Installer for Linux Ubuntu 17.10 x86_64
The base installer is available for download below.
Base Installer
Download (1.7 GB)
Installation Instructions:
1. Run `sudo sh cuda_9.2.148_396.37_linux.run`
2. Follow the command-line prompts

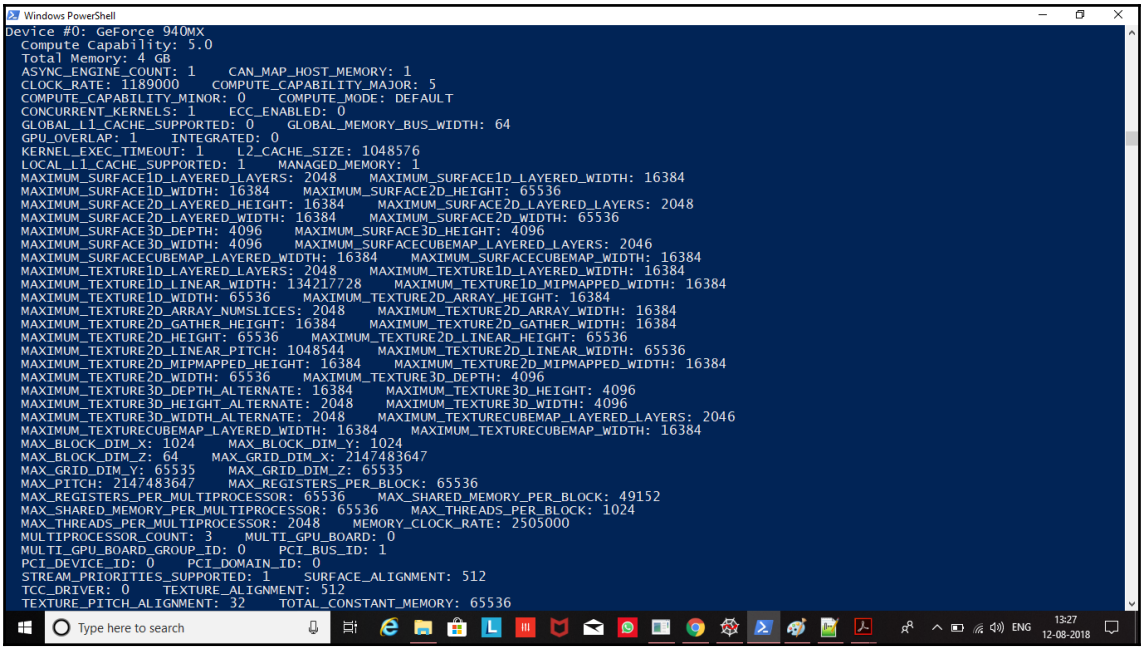

Windows
macOS
Linux
Anaconda 5.2 For Linux Installer
Python 3.6 version *
Download
64-Bit (x86) Installer (622 MB)
64-Bit (Power8 and Power9) Installer (288 MB)
32-Bit Installer (507 MB)
Python 2.7 version *
Download
64-Bit (x86) Installer (603 MB)
64-Bit (Power8 and Power9) Installer (270 MB)
32-Bit Installer (489 MB)

```
bhaumik@bhaumik-Lenovo-ideapad-520-15IKB:~$ conda install -c lukepfister pycuda
Solving environment: done


==> WARNING: A newer version of conda exists. <==
  current version: 4.4.10
  latest version: 4.5.8

Please update conda by running

    $ conda update -n base conda



## Package Plan ##

  environment location: /home/bhaumik/anaconda3

  added / updated specs:
    - pycuda


The following packages will be downloaded:

    package                    |            build
    ---------------------------|-----------------
    ca-certificates-2018.03.07 |                0         124 KB
    pycuda-2017.1              |            py36_0         627 KB  lukepfister
    appdirs-1.4.3              |   py36h28b3542_0          16 KB
    pytools-2018.4             |            py36_0         110 KB  lukepfister
    mako-1.0.4                 |            py36_0         116 KB  lukepfister
    openssl-1.0.2o             |       h20670df_0         3.4 MB
    certifi-2018.4.16          |            py36_0         142 KB
    ------------------------------------------------------------
                                           Total:         4.6 MB
```



Chapter 11: Working with PyCUDA

```
PS G:\cude opencv book material\CUDA book code\Chapter11> python .\hello_pycuda.py
Hello,PyCUDA!!!
PS G:\cude opencv book material\CUDA book code\Chapter11> _
```




```
Windows PowerShell
PS G:\cude opencv book material\CUDA book code\Chapter11> python .\thread_execution.py
I am in block no: 2
I am in block no: 7
I am in block no: 0
I am in block no: 8
I am in block no: 1
I am in block no: 3
I am in block no: 5
I am in block no: 4
I am in block no: 6
I am in block no: 9
PS G:\cude opencv book material\CUDA book code\Chapter11>
```

```
In [11]: runfile('G:/cude opencv book material/CUDA book code/Chapter11/add_n.py',
wdir='G:/cude opencv book material/CUDA book code/Chapter11')
Addition on GPU:
1.2273334 + 1.3404454 = 2.5677788
```

```
Addition on GPU:
0.7673203 + 0.5080069 = 1.2753272
0.28488383 + 0.15324554 = 0.43812937
-0.3220178 + -0.10700232 = -0.4290201
-0.501334 + -1.7047318 = -2.206066
0.023053076 + 0.34796545 = 0.37101853
-0.44806996 + -2.071736 = -2.5198061
2.9193559 + 0.7721601 = 3.691516
-0.8016763 + -0.31726292 = -1.1189392
0.607628 + -1.2539302 = -0.6463022
-0.80436313 + 1.2789425 = 0.47457933
```

```
In [9]: runfile('G:/cude opencv book material/CUDA book code/Chapter11/add_number.py',
wdir='G:/cude opencv book material/CUDA book code/Chapter11')
Addition of 1000000 element of GPU
0.009422s
Addition of 1000000 element of CPU
0.41515421867370605 s
```

```
Time of Squaring on GPU without inout
0.149003s
original array:
[[3. 2. 1. 4. 3.]
 [1. 1. 1. 2. 2.]
 [2. 1. 1. 3. 2.]
 [3. 4. 2. 3. 1.]
 [4. 1. 4. 3. 1.]]
Square with kernel:
[[ 9.  4.  1. 16.  9.]
 [ 1.  1.  1.  4.  4.]
 [ 4.  1.  1.  9.  4.]
 [ 9. 16.  4.  9.  1.]
 [16.  1. 16.  9.  1.]]
```

```
Square with InOut:
[[ 9.  4.  1. 16.  9.]
 [ 1.  1.  1.  4.  4.]
 [ 4.  1.  1.  9.  4.]
 [ 9. 16.  4.  9.  1.]
 [16.  1. 16.  9.  1.]]
Time of Squaring on GPU with inout
0.004260s
```

```
original array:
[[1 1 3 1 4]
 [4 3 4 2 1]
 [4 1 1 4 4]
 [2 3 2 4 1]
 [2 1 2 4 3]]
Squared with gpuarray:
[[ 1.  1.  9.  1. 16.]
 [16.  9. 16.  4.  1.]
 [16.  1.  1. 16. 16.]
 [ 4.  9.  4. 16.  1.]
 [ 4.  1.  4. 16.  9.]]
Time of Squaring on GPU with gpuarray
0.058682s
```

```
In [3]: runfile('G:/cude opencv book material/CUDA book code/Chapter11/gpu_dot.py',
wdir='G:/cude opencv book material/CUDA book code/Chapter11')
Answer of Dot Product using numpy
633.0
Time taken for Dot Product using numpy
0.03769350051879883 s
Answer of Dot Product on GPU
633.0
Time taken for Dot Product on GPU
0.000108s
The computed dor product is correct
```


```
In [3]: runfile('G:/cude opencv book material/CUDA book code/Chapter11/gpu_dot.py',
wdir='G:/cude opencv book material/CUDA book code/Chapter11')
Answer of Dot Product using numpy
633.0
Time taken for Dot Product using numpy
0.03769350051879883 s
Answer of Dot Product on GPU
633.0
Time taken for Dot Product on GPU
0.000108s
The computed dor product is correct
```


$$\begin{bmatrix} 4 & 1 & 1 \\ 1 & 2 & 3 \\ 1 & 1 & 3 \end{bmatrix} + \begin{bmatrix} 3 & 3 & 4 \\ 2 & 3 & 3 \\ 4 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 4*3+1*2+1*4 & 16 & 20 \\ 1*3+2*2+3*4 & 12 & 13 \\ 1*3+1*2+3*4 & 9 & 10 \end{bmatrix}$$

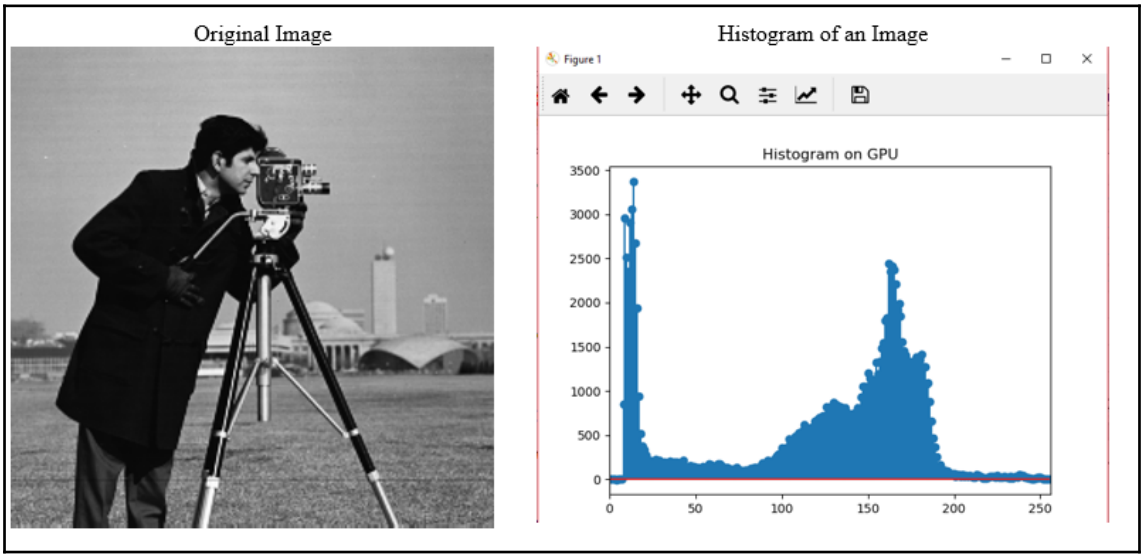
```
In [7]: runfile('G:/cude opencv book material/CUDA book code/Chapter11/matrix_mmulfinal.py', wdir='G:/cude opencv book material/CUDA book code/Chapter11')
**********************************************************************************************
Matrix A:
[[4. 1. 1.]
 [1. 2. 3.]
 [1. 1. 3.]]
**********************************************************************************************
Matrix B:
[[3. 3. 4.]
 [2. 3. 3.]
 [4. 1. 1.]]
**********************************************************************************************
Matrix Multiplication result:
[[18. 16. 20.]
 [19. 12. 13.]
 [17.  9. 10.]]


The computed matrix multiplication is correct
```

```
In [4]: runfile('G:/cude opencv book material/CUDA book code/Chapter12/
element_wise_addition.py', wdir='G:/cude opencv book material/CUDA book code/Chapter12')
Addition of 1000000 element of GPU
0.000629s
The sum computed on GPU is correct
```

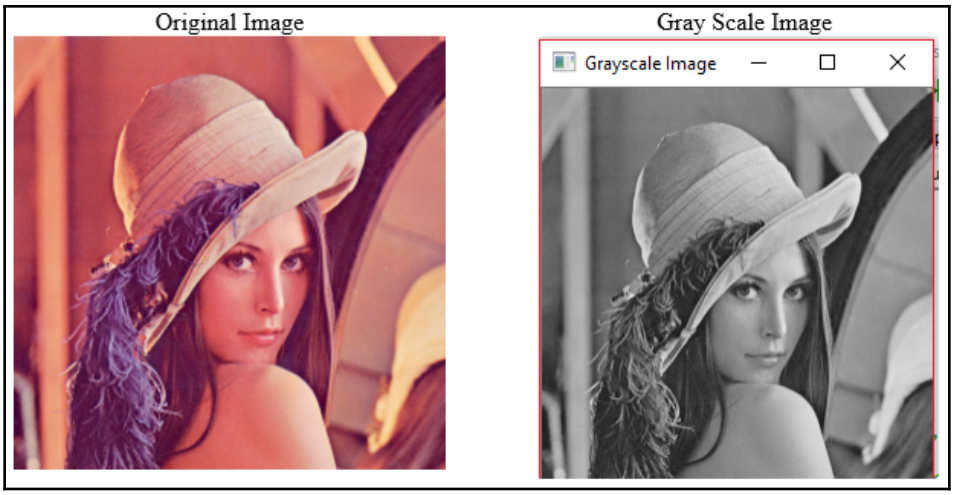
```
In [6]: runfile('G:/cude opencv book material/CUDA book code/Chapter12/gpu_dot.py',
wdir='G:/cude opencv book material/CUDA book code/Chapter12')
Vector A
[0 1 2 3 4]
Vector B
[0 1 2 3 4]
The computed dot product using reduction:
30
Dot Product on GPU
2.546338s
```

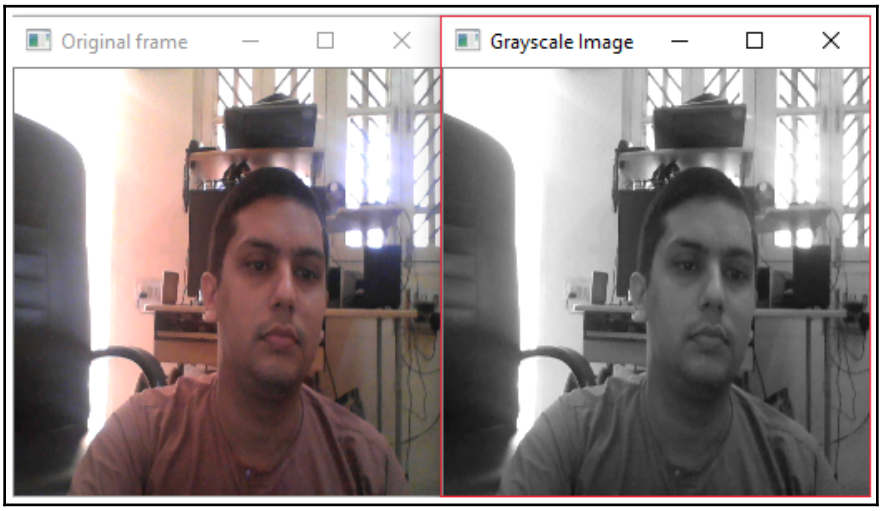
```
In [3]: runfile('G:/cude opencv book material/CUDA book code/Chapter12/gpu_scan.py',
wdir='G:/cude opencv book material/CUDA book code/Chapter12')
The input data:
[7 5 9 2 9 7 7 7 2 6]
The computed cumulative sum using Scan:
[ 7 12 21 23 32 39 46 53 55 61]
Cumulative Sum on GPU
0.002032s
```

Chapter 12: Basic Computer Vision Applications Using PyCUDA




```
In [4]: runfile('G:/cude opencv book material/CUDA book code/Chapter12/
histogram_without_shared.py', wdir='G:/cude opencv book material/CUDA book code/
Chapter12')
Time for Calculating Histogram on GPU without shared memory
0.001040s
Time for Calculating Histogram using OpenCV
0.001040s

In [5]: runfile('G:/cude opencv book material/CUDA book code/Chapter12/histogram.py',
wdir='G:/cude opencv book material/CUDA book code/Chapter12')
Time for Calculating Histogram on GPU with shared memory
0.000839s
```


Original Image
Gray Scale Image
Grayscale Image


Original frame
Grayscale Image