



# BEST PRACTICES FOR CUDNN

DG-09678-001\_v7.6.5 | November 2019

## Best Practices



## TABLE OF CONTENTS

Chapter 1. Introduction.....	1
Chapter 2. Best Practices For Medical Imaging.....	2
2.1. Recommended Settings In cuDNN While Performing 3D Convolutions.....	3
Chapter 3. Medical Imaging Performance.....	5
Chapter 4. Medical Imaging Limitations.....	6

# Chapter 1.

## INTRODUCTION



**Attention** These guidelines are applicable to 3D convolution and deconvolution functions starting in cuDNN v7.6.3.

This document provides guidelines for setting the cuDNN library parameters to enhance the performance of 3D convolutions. Specifically, these guidelines are focused on settings such as filter sizes, padding and dilation settings. Additionally, an application-specific use-case, namely, medical imaging, are presented to demonstrate the performance enhancement of 3D convolutions with these recommended settings.

Specifically, these guidelines are applicable to the following functions and their associated data types:

- ▶ [cudnnConvolutionForward\(\)](#)
- ▶ [cudnnConvolutionBackwardData\(\)](#)
- ▶ [cudnnConvolutionBackwardFilter\(\)](#)

For more information, see the [cuDNN Developer Guide](#) and [cuDNN API](#).

# Chapter 2.

## BEST PRACTICES FOR MEDICAL IMAGING

To optimize your performance in your model, ensure you meet the following general guidelines:

### Layout

The layout is in NCHW format.

### Filter size

The filter size is **Tx1x1**, **Tx2x2**, **Tx3x3**, **Tx5x5**, where **T** is a positive integer. There are additional limits for the value of **T** in **wgrad** and strided **dgrad**.

### Padding

(filter size // 2), for example, 0x0x0 for 1x1x1 filter, 1x1x1 for 3x3x3 filter

### Stride

Arbitrary for forward and backward filter; **dgrad/deconv**: 1x1x1 or 2x2x2 with 2x2x2 filter.

### Convolution mode

Cross-correlation for forward, arbitrary for **dgrad** and **wgrad**.

### Dilation

The dilation is 1x1x1.

### Platform

The platform is Volta with input/output channels divisible by 8.

### Batch/image size

cuDNN will fallback to non-Tensor Core kernel if it determines that the workspace required is larger than 256MB of GPU memory. The workspace required depends on many factors. For the Tensor Core kernels, the workspace size generally scales linearly with output tensor size. Therefore, this can be mitigated by using smaller image sizes or minibatch sizes.

## 2.1. Recommended Settings In cuDNN While Performing 3D Convolutions

The following table shows the specific improvements that were made in each patch release.

Table 1 Recommended settings while performing 3D convolutions

Volta						
cuDNN version		7.6.2	7.6.2	7.6.1	7.6.1	7.6.1
Convolution (3D or 2D)		3D				
Convolution or deconvolution (fprop, dgrad, or wgrad)		dgrad	fprop	wgrad	dgrad	fprop
Grouped convolution	Yes or No	No				
	Group size	NA				
Data layout format (NHWC/NCHW) <sup>1</sup>		NCDHW	NCDHW <sup>2</sup>			
Input/output precision (FP16, FP32, or FP64)		FP16 or FP32	FP16 <sup>3</sup> or FP32 <sup>4</sup>			
Accumulator (compute) precision (FP16, FP32, or FP64)		Better to be the same input and output precision	FP32			
Filter (kernel) sizes		2x2x2	<ul style="list-style-type: none"> <li>▶ T<sup>5</sup>x1x1</li> <li>▶ Tx2x2</li> <li>▶ Tx3x3</li> <li>▶ Tx5x5</li> </ul>	<ul style="list-style-type: none"> <li>▶ 1x1x1</li> <li>▶ 2x2x2</li> <li>▶ 3x3x3</li> <li>▶ 5x5x5</li> </ul>	<ul style="list-style-type: none"> <li>▶ Tx1x1</li> <li>▶ Tx2x2</li> <li>▶ Tx3x3</li> <li>▶ Tx5x5</li> </ul>	<ul style="list-style-type: none"> <li>▶ Tx1x1</li> <li>▶ Tx2x2</li> <li>▶ Tx3x3</li> <li>▶ Tx5x5</li> </ul>
Padding			Filter // 2 <sup>6</sup>			
Image sizes			256 MB WS limit <sup>7</sup>	256 MB WS limit <sup>8</sup>	256 MB WS limit <sup>9</sup>	256 MB WS limit <sup>10</sup>

<sup>1</sup> NHWC/NCHW corresponds to NDHWC/NCDHW in 3D convolution.

<sup>2</sup> With NCHW <> NHWC format transformation.

<sup>3</sup> FP16: CUDNN\_TENSOROP\_MATH

<sup>4</sup> FP32: CUDNN\_TENSOROP\_MATH\_ALLOW\_CONVERSION

<sup>5</sup> An arbitrary positive value.

<sup>6</sup> padding = filter // 2 constraints is no longer required in integrated kernel

<sup>7</sup> fprop: reduction

```
buffer size = ceil(k / tileN) * tileN * ceil(n*o*p*q / tileM) *
             tileM
```

<sup>8</sup> wgrad: reduction

```
buffer size = ceil(c / tileN) * tileN * ceil(k*t*r*s /
```

Volta						
cuDNN version		7.6.2	7.6.2	7.6.1	7.6.1	7.6.1
Number of channels	C	Arbitrary	0 mod 8			
	K	Arbitrary	0 mod 8			
Convolution mode			Cross-correlation			Cross-correlation
Strides		2x2x2	Arbitrary stride	1x1x1		
Dilation		1x1x1				

## Chapter 3.

# MEDICAL IMAGING PERFORMANCE

The following table shows the average speed-up of **unique cuDNN 3D convolution calls** for each network that satisfies the conditions in [Best Practices For Medical Imaging](#).

Model	Batchsize	Avg. Speed-up of unique cuDNN 3D convolution API calls (7.6.3 vs. 7.5.1)
V-Net (3D-Image segmentation)	2	4.4x
	4	4.4x
	8	4x
	16	4x
	32	4x
	64	3.4x
	128	3x
3D-UNet (3D-Image Segmentation)	2	4.4x
	4	4.1x
	8	4.4x
	16	4.3x
	32	4x
	64	4x
	128	4.2x

## Chapter 4.

# MEDICAL IMAGING LIMITATIONS

Your application will be functional but slow if the model has:

- ▶ Channel counts lower than 32 (gets worse the lower it is)
- ▶ Data gradients for convolutions with stride

If the above is in the network, use **cuDNNFind** to get the best option.



## Notice

THE INFORMATION IN THIS GUIDE AND ALL OTHER INFORMATION CONTAINED IN NVIDIA DOCUMENTATION REFERENCED IN THIS GUIDE IS PROVIDED “AS IS.” NVIDIA MAKES NO WARRANTIES, EXPRESSED, IMPLIED, STATUTORY, OR OTHERWISE WITH RESPECT TO THE INFORMATION FOR THE PRODUCT, AND EXPRESSLY DISCLAIMS ALL IMPLIED WARRANTIES OF NONINFRINGEMENT, MERCHANTABILITY, AND FITNESS FOR A PARTICULAR PURPOSE. Notwithstanding any damages that customer might incur for any reason whatsoever, NVIDIA’s aggregate and cumulative liability towards customer for the product described in this guide shall be limited in accordance with the NVIDIA terms and conditions of sale for the product.

THE NVIDIA PRODUCT DESCRIBED IN THIS GUIDE IS NOT FAULT TOLERANT AND IS NOT DESIGNED, MANUFACTURED OR INTENDED FOR USE IN CONNECTION WITH THE DESIGN, CONSTRUCTION, MAINTENANCE, AND/OR OPERATION OF ANY SYSTEM WHERE THE USE OR A FAILURE OF SUCH SYSTEM COULD RESULT IN A SITUATION THAT THREATENS THE SAFETY OF HUMAN LIFE OR SEVERE PHYSICAL HARM OR PROPERTY DAMAGE (INCLUDING, FOR EXAMPLE, USE IN CONNECTION WITH ANY NUCLEAR, AVIONICS, LIFE SUPPORT OR OTHER LIFE CRITICAL APPLICATION). NVIDIA EXPRESSLY DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY OF FITNESS FOR SUCH HIGH RISK USES. NVIDIA SHALL NOT BE LIABLE TO CUSTOMER OR ANY THIRD PARTY, IN WHOLE OR IN PART, FOR ANY CLAIMS OR DAMAGES ARISING FROM SUCH HIGH RISK USES.

NVIDIA makes no representation or warranty that the product described in this guide will be suitable for any specified use without further testing or modification. Testing of all parameters of each product is not necessarily performed by NVIDIA. It is customer’s sole responsibility to ensure the product is suitable and fit for the application planned by customer and to do the necessary testing for the application in order to avoid a default of the application or the product. Weaknesses in customer’s product designs may affect the quality and reliability of the NVIDIA product and may result in additional or different conditions and/or requirements beyond those contained in this guide. NVIDIA does not accept any liability related to any default, damage, costs or problem which may be based on or attributable to: (i) the use of the NVIDIA product in any manner that is contrary to this guide, or (ii) customer product designs.

Other than the right for customer to use the information in this guide with the product, no other license, either expressed or implied, is hereby granted by NVIDIA under this guide. Reproduction of information in this guide is permissible only if reproduction is approved by NVIDIA in writing, is reproduced without alteration, and is accompanied by all associated conditions, limitations, and notices.

## Trademarks

NVIDIA, the NVIDIA logo, and cuBLAS, CUDA, cuDNN, DALI, DIGITS, DGX, DGX-1, DGX-2, DGX Station, DLProf, Jetson, Kepler, Maxwell, NCCL, Nsight Compute, Nsight Systems, NvCaffe, PerfWorks, Pascal, SDK Manager, Tegra, TensorRT, TensorRT Inference Server, Tesla, TF-TRT, and Volta are trademarks and/or registered trademarks of NVIDIA Corporation in the United States and other countries. Other company and product names may be trademarks of the respective companies with which they are associated.

## Copyright

© 2019 NVIDIA Corporation. All rights reserved.

[www.nvidia.com](http://www.nvidia.com)

