PROCEEDINGS OF SCIENCE

# Muon trigger using deep neural networks accelerated by FPGAs

**Seulgi Kim,**[a] **Jason Lee,**[a,*] **Inkyu Park,**[a] **Youngwan Son,**[a,*] **Ian James Watson**[a] **and Seungjin Yang**[a]

[a] *Department of Physics, University of Seoul,*
*163 Seoulsiripdae-ro Dongdaemun-gu, Seoul, Republic of Korea*

*E-mail:* sonkun2005@uos.ac.kr, jason.lee@physics.uos.ac.kr

Accuracy and latency are crucial to the trigger system in high luminosity particle physics experiments. We investigate the usage of deep neural networks (DNN) to improve the accuracy of the muon track segment reconstruction process at the trigger level. Track segments, made by hits within a detector module, are the initial partial reconstructed objects which are the typical building blocks for muon triggers. Currently, these segments are coarsely reconstructed on FPGAs to keep the latency manageable. DNNs are ideal for these types of pattern recognition problems, and so we examine the potential for DNN based track segment reconstruction to be accelerated by dedicated FPGAs to improve both processing speed and accuracy for the trigger system.

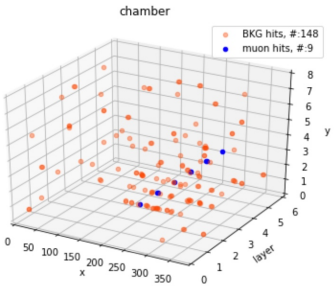
*40th International Conference on High Energy Physics - ICHEP2020*
*July 28 - August 6, 2020*
*Prague, Czech Republic (virtual meeting)*

*Speaker

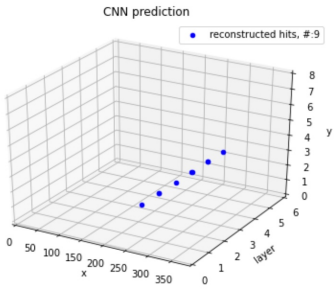




PoS(ICHEP2020)712




**(a)** All hits




**(b)** CNN prediction of 1a

| Layer | Input shape | Output shape |
|---|---|---|
| Depthwise Conv2D | (Batch, 8, 384, 6) | (Batch, 8, 384, 6) |
| Pointwise Conv2D | (Batch, 8, 384, 6) | (Batch, 8, 384, 36) |
| ReLU | (Batch, 8, 384, 36) | (Batch, 8, 384, 36) |
| Depthwise Conv2D | (Batch, 8, 384, 36) | (Batch, 8, 384, 36) |
| Pointwise Conv2D | (Batch, 8, 384, 36) | (Batch, 8, 384, 6) |

**(c)** Model structure (NHWC format)

**Figure 1:** Visualization of the chamber with all hits in 1a and its CNN prediction 1b by 1c

## 1. Introduction

Recently, a boosted decision tree (BDT) has been successfully implemented into the CMS endcap muon track finder trigger using an FPGA implementation [1]. The BDT was used to assign transverse momentum on muon trigger objects, which is a collection of segments from multiple detector modules (chambers), showing great improvements over the previous look up table method [2]. We aim to improve similar tracking based triggers by improving the segments. Segments are a collection of hits within one detector modules or chambers that represent a partial track. We implement a convolutional neural network (CNN) to improve the segment making process.

## 2. CNN based Reconstruction in FPGA

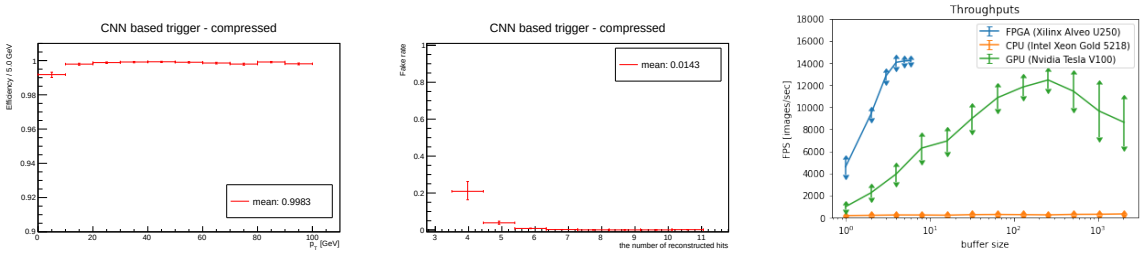
We use GEANT4 [3] to simulate a muon detector chamber similar to [4], consisting of 6 layers in the z-axis, with each layer consisting of 384 strips in the x-axis and segmented in 8 partitions in the y-axis.

The aim is to train the CNN to only reconstruct the muon segments which are marked as blue points in Fig. 1a and then accelerating the CNN inference by FPGA. Because of the restriction of layers in FPGA DPU Architecture, 2D Depthwise Separable Convolution is used to learn the relation on channel axis direction [5] [6] [7]. The model is trained in the tensorflow.keras framework, and the structure is given in 1c. [8]

Xilinx Alveo U250 FPGA is used for inference acceleration. The weights of the CNN are quantized by changing 32 bit float to 8 bit integer, in order to save memory usage and reduce latency [9].

## 3. Results

Efficiency and fake rate are used to evaluate the model, as shown in Fig. 2. The efficiency is defined as the number of matched images divided by the number of images containing muons, and the fake rate as the fraction of the number of wrong reconstructed hits. Fig 2c shows the throughput which is used to benchmark the latency of various platforms and it is measured as processed images per seconds (FPS), includes data transferring delays. The performance is very promising, showing very high efficiency with the fake rate low and with even lower latency.




**(a)** Efficiency **(b)** Fake rate **(c)** Throughput

**Figure 2:** (a): Efficiency (b): Fake rate (c): Throughput as a function of batch size is 1, 2, ..., 2048.

## Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (2019R1C1C1009200).

## References

[1] D. Acosta *et al.* [CMS], J. Phys. Conf. Ser. **1085** (2018) no.4, 042042 doi:10.1088/1742-6596/1085/4/042042

[2] A. M. Sirunyan *et al.* [CMS], JINST **15** (2020) no.10, P10017 doi:10.1088/1748-0221/15/10/P10017 [arXiv:2006.10165 [hep-ex]].

[3] J. Allison, J. Apostolakis, S. B. Lee, K. Amako, S. Chauvie, A. Mantero, J. I. Shin, T. Toshito, P. R. Truscott and T. Yamashita, *et al.* Nucl. Instrum. Meth. A **835** (2016), 186-225 doi:10.1016/j.nima.2016.06.125

[4] A. M. Sirunyan *et al.* [CMS], The Phase-2 Upgrade of the CMS Muon Detectors CERN-LHCC-2017-012

[5] Xilinx, Zynq DPU v3.2 IP Product Guide PG338 https://www.xilinx.com/support/documentation/ip_documentation/dpu/v3_2/pg338-dpu.pdf

[6] F. Chollet, Xception: Deep Learning with Depthwise Separable Convolutions [arXiv:1610.02357 [cs.CV]]

[7] J. Long, E. Shelhamer, T. Darrell, Fully Convolutional Networks for Semantic Segmentation [arXiv:1411.4038 [cs.CV]]

[8] F. Chollet*et al.*, Keras https://github.com/fchollet/keras

[9] Xilinx, Vitis AI User Guide UG1414 https://www.xilinx.com/support/documentation/sw_manuals/vitis_ai/1_2/ug1414-vitis-ai.pdf