A DEEP LEARNING ACCELERATOR FOR LOSSLESS IMAGE COMPRESSION USING TVM/VTA STACK

Malan Lakshan Evans

(219331A)

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

March 2023

A DEEP LEARNING ACCELERATOR FOR LOSSLESS IMAGE COMPRESSION USING TVM/VTA STACK

Malan Lakshan Evans

(219331A)

Thesis submitted in partial fulfilment of the requirement for the degree Master of Science in Computer Science

Department of Computer Science and Engineering

University of Moratuwa
Sri Lanka

March 2023

Declaration

I declare that this is my own work, and this MSc Research Project Report does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works.

………………………. 16/07/2023

Malan Evans Date

The above candidate has carried out research for the Masters' thesis under my supervision.

Name of the supervisor: Professor Sanath Jayasena

………………………. 16/07/2023

Prof. Sanath Jayasena Date

Acknowledgment

I would like to thank my supervisor Professor Sanath Jayasena for his dedicated support and guidance from the moment he accepted this research project. He always shared his advice with me to drive this research project work in the right direction. I initially thought it would be a bit hard for me to contact him regularly due to his obvious busy schedule. But it was never the case. I have been so fortunate to have him as the supervisor for this research project, not only because he is one of the most senior professors in the department. But because he is also very kind, flexible and humble and it was so easy for me to communicate with him freely. I could not have completed this research project successfully if it was not for his guidance.

I would like to thank my former employer, Synopsys Lanka (Private) Limited for encouraging me to pursue this master's programme. I specially thank my former manager, Mr. Aabid Rushdi for his support, guidance and motivation which sparked my interest in this Computer Science master's degree.

I also would love to mention the support from my very own family. I really believe that my parents who encouraged me through every step of my life have contributed to the success of this research project. I would love to extend my gratitude to my wife's parents as well for motivating me always to strive. I must acknowledge my wife for standing beside me always encouraging me, guiding me, and motivating me. I was about to give up this research work many times when I was kept failing. She was the reason why I could put the failures behind me to achieve this height.

Abstract

Image compression is a requirement for storing and transmitting images. Many fields like digital photography demand lossless image compression as it's a requirement to reconstruct compressed images without a loss. Deep learning has opened so much room for improvements in image processing-related tasks. There are lossless image compression algorithms which use deep learning to achieve impressive compression ratio values. However, the time efficiency of lossless image compression might be affected due to using deep learning. Low-cost edge computing devices cannot use GPUs to accelerate deep learning algorithms. Deep Learning Accelerators (DLA) are the most feasible solution to eliminate time efficiency issues of deep learning-based algorithms for edge computing devices. Deep learning-based lossless image compression solution can be implemented in a System on Chip (SoC) with a Field Programmable Gate Array (FPGA). We propose a lossless image compression system in which a properly trained deep Convolutional Neural Network (CNN) is used to predict residual errors of LOCO-I-based pixel value prediction. Adaptive Arithmetic coding is applied to further improve the compression ratio. The main contribution of our approach is implementing the trained deep CNN in hardware using TVM/VTA stack. Finally, our proposed solution implements an end-to-end lossless image compression system by carrying out the prediction of residual error values of LOCO-I-based pixel value prediction in a Pynq-Z1 board (FPGA) while performing the rest of the tasks in a Python application. TVM/VTA stack stands as a bridge between the Python application and the FPGA. The proposed method yields a better compression performance with respect to state-of-the-art codecs according to the experimental results. The hardware implementation improves the time efficiency significantly enabling utilising the predictive power of deep CNNs for image compression systems. This is the first time a DLA is used effectively in a lossless image compression system, to the best of our knowledge.

Keywords: Lossless Image Compression, FPGA, DLA, VTA, TVM, Real-time, Pynq-Z1

Table of Contents

1. Introduction 1

1.1. Background 1

1.1.1. Overview of image compression 1

1.1.2. Image compression algorithms 1

1.1.3. Evaluation of image compression algorithms 1

1.1.4. Deep Neural Networks (DNNs) 2

1.1.5. Time efficiency of DNNs 2

1.1.6. Deep Learning Accelerator (DLA) 3

1.2. Problem definition 3

1.3. Research objectives 4

2. Literature review 6

2.1. Image Compression 6

2.2. Non-Learning Codecs 7

2.2.1. Lossy Image Compression 7

2.2.2. Lossless Image Compression 8

2.3. Learning Codecs 9

2.3.1. Lossy Image Compression 9

2.3.2. Lossless Image Compression 10

2.4. Hardware Implementations 11

2.5. Deep Learning Accelerators 12

3. Methodology 13

3.1. Overview of the proposed method 13

3.2. Deep Learning based Prediction 15

3.2.1. Selection of Causal Neighbourhood 16

3.2.2. Pixel predictor in approach II 18

3.2.3. The Convolutional Neural Network (CNN) 19
3.2.4. Training CNN 23
3.3. Error Coding 25
3.4. End-to-end lossless image compression system 26
3.5. Deep Learning Accelerator for lossless image compression 26
3.5.1. Hardware accelerator using TVM/VTA stack 27
3.6. End-to-end lossless image compression system with DLA 30
4. Implementation 31
4.1. Implementation of CNN 31
4.1.1. General CNN model definition 31
4.1.2. CNN model compilation 32
4.1.3. DNN model 36
4.2. Entropy coding 38
4.3. End-to-end image compression flow 38
4.4. End-to-end image decompression flow 39
4.5. Obtaining DLA 40
4.5.1. Setting up TVM/VTA with FPGA 40
4.5.2. Prepare DNNs for compilation into hardware implementation 42
4.5.3. Compile DNN into the hardware implementation 43
5. Evaluations 45
5.1. Evaluation of compression performance 46
5.2. Evaluation of elapsed time 52
5.3. Evaluation of compression quality 54
5.4. Evaluation of FPGA resource utilisation 56
6. Discussion and conclusion 58
6.1. Discussion 58

6.2. Limitations and Future Work 60

6.3. Conclusion 61

7. References 63

List of Figures

Figure 3.1: Main blocks of a traditional prediction-based lossless image compression algorithm 13
Figure 3.2: Lossless image compression system with residual error prediction........ 14
Figure 3.3.3: General usage of the proposed CNN 16
Figure 3.4: The Causal Neighbourhood for d=5 17
Figure 3.5: Sample image of size M x N 17
Figure 3.6: Causal neighbourhood for pixel P_{xy} 18
Figure 3.7: Residual error for image of size M x N 18
Figure 3.8: Causal neighbourhood for residual error at (x, y) pixel 18
Figure 3.9: Convolutional Neural Network Architecture 22
Figure 3.10: training data preparation in approach I 24
Figure 3.11: training data preparation in approach II 25
Figure 3.12: Proposed lossless image compression system approach I 26
Figure 3.13: Proposed lossless compression system approach II 26
Figure 3.14: Basic use model of TVM 28
Figure 3.15: DLA for pixel value prediction using TVM/VTA stack and FPGA 29
Figure 4.1: Implementation of the CNN 31
Figure 4.2: Proximity-based loss algorithm 34
Figure 5.1: Mandril qualitative analysis 55
Figure 5.2: Cameraman qualitative analysis 56
Figure 5.3: Pirate qualitative analysis 56

List of Tables

Table 3.1: Input, output shapes and parameters at each layer of the proposed CNN architecture 23
Table 4.1: Different CNN compile options 32
Table 4.2: Results of experiments to find CNN model compile options 35
Table 4.3: Experiment find out best unavailable causal neighbourhood pixel value. 37
Table 4.4: Different DNN models in the proposed approaches 38
Table 5.1: Compression performance comparison for test image set in BPP 47
Table 5.2: Shannon Entropy values for test images 47
Table 5.3: Average BPP values for test images 48
Table 5.4: Compression performance for test images considering compression ratio 49
Table 5.5: Average compression ratios for test images 49
Table 5.6: compression performance for known images in BPP 50
Table 5.7: Shannon Entropy values for known images 51
Table 5.8: Average BPP for each compression algorithm/standard for known images 51
Table 5.9: Compression performance for known images considering compression ratio 51
Table 5.10: Average compression ratios for known images 52
Table 5.11: Elapsed time of each compression algorithm and DNNs in seconds 53
Table 5.12: Pixel error in different implementations of Approach II full model-based method 55
Table 5.13: Resource utilisation comparison of the proposed approach 57