Available onlinewww.ejaet.com

European Journal of Advances in Engineering and Technology, 2022, 9(4s):449-454
International Conference on Tech Trends in Science & Engineering (ICTTSE) 2022
Suryodaya College of Engineering & Technology, Suryodaya Polytechnic, Nagpur, Maharashtra, India

Research Article ISSN: 2394 - 658X

Discussion on the Design of a Proposed CNN Model Based on FPGA

Amita P. Thakare[1], Sunil Kumar[2]

Research Scholar[1], Professor[2]
Electronics Engineering, Kalinga University, Naya Raipur, India
amita.thakare27@gmail.com[1], sunil.kumar@kalingauniversity.ac.in[2]

ABSTRACT

Convolutional neural networks (CNNs) have paved the way for black-box primarily based classification by augmentation in function extraction, and classification methods. The design of a CNN primarily based classification device is very complex, and requires implementation of convolutional function extraction unit, most variance pooling unit for function selection, rectilinear unit (ReLU) for characteristic activation, totally linked neural community (FCNN) unit for classification, and SoftMax for remaining category activation & chance evaluation. In order to function this task, a giant wide variety of Field Programmable Gated Array (FPGA) primarily based designs are proposed by using researchers. But these designs have inherent redundancy issues, which limits their speed and will increase their strength consumption when utilized in real-time classification systems. In order to enhance the overall performance of hardware-level CNN designs, this textual content proposes an incredibly environment-friendly parallel CNN mannequin sketch for FPGA-based total deployments.

Key words: Convolution, neural, network, FPGA, model, integration

---

INTRODUCTION

Artificial Genius (AI) has turn out to be an integral phase of our everyday activities. These things to do vary from the usage of AI enabled clever telephones to controlling industrial & domestic equipment. These AI based totally units are powered with the aid of convolutional neural networks (CNNs), or comparable deep mastering models.

Thus, environment friendly diagram of these fashions with minimal latency, and most throughput is needed. Thus, FPGA based totally CNN fashions are designed by means of researchers, which aid in direct hardware-level interfacing, thereby improving usual speed, with the aid of discount in software program degree redundancies [1]. An ordinary CNN model, with more than one kernels & convolutional layers is depicted in parent 1, whereby buffers & shared registers are used to classify given enter into a couple of output categories.

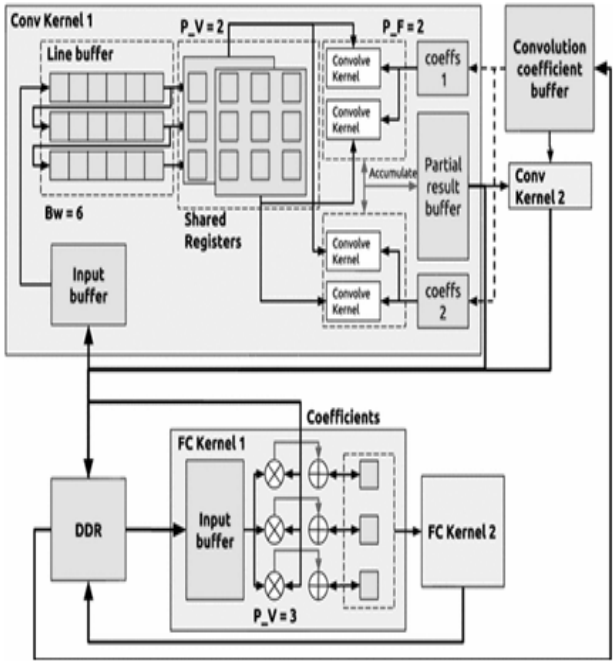

Fig. 1 Design of a typical CNN model based on FPGA

The model graph makes use of line buffers to store incoming enter data, which is given to shared registers for performing convolutional operations. These operations are improved with neighborhood coefficients, which assists in discovering a preliminary feature set. The factors are given to a double fact cost (DDR) module, the region attribute willpower and variance maximization processes are applied. The chosen elements are given to a evaluation layer, whereby neurons are used for performing remaining classification [2]. Thus, the plan of memory unit, arithmetic & logical processing unit, neuron unit, and evaluation unit are majorly wished to structure a notably surroundings pleasant CNN model. A massive vary of machine fashions are proposed through capability of researchers, which useful resource in designing CNNs on FPGAs & different VLSI (very large-scale integration) circuits. These fashions vary in phrases of power, area, and prolong requirements. To summarize their performance, a consider of these fashions alongside with their nuances, advantages, drawbacks, and future look up scopes is noted in the subsequent section. This will assist readers to find out first-rate workable techniques of CNN structure for their inner use cases. This will assist readers to identify best possible methods of CNN design for their internal use cases.

LITERATURE REVIEW

A huge form of CNN fashions for FPGA layout are proposed via way of means of researchers, and every of them varies in phrases of area, strength and postpone requirements. For instance, the paintings in [3, 4, 5, 6] proposes excessive throughput CNN accelerator-primarily based totally layout, speedy multiplier-primarily based totally CNN layout, reconfigurable CNN (RCNN), and hyperspectral imaging CNN designs. These designs are found to have higher computational performance whilst carried out to precise packages, and hence have restricted scalability. To enhance this scalability, paintings in [7] proposes binary weights-primarily based totally electricity green CNN version architecture. This version assists in lowering computational complexity thru use of binary weights for convolutional operations. Due to which the proposed version is able to excessive-velocity, and occasional electricity computations. Similar fashions are proposed in [8, 9, 10, 11], in which photograph super-resolution, coprocessor layout for constant factor CNNs, residue variety gadget (RNS) primarily based totally CNN, and softcore primarily based totally CNN fashions are mentioned. These fashions help in lowering redundancies throughout convolutional computations, thereby enhancing typical velocity & decreasing electricity wanted throughout function selection & class operations. Extensions to those fashions are mentioned in [12, 13, 14, 15], wherein paintings & weight load balancing, area exploration fashions in CNN, utility precise mixture of various CNN fashions, and complicated arithmetic-primarily based totally Angel Eye layout for CNN (AE-CNN) are mentioned. These fashions permit community designers to estimate portability problems with current CNN designs, and comprise them for lowering redundancies, and improving utility

precise deployment functionality for those fashions. Modern designs for CNN may be found the use of intensity smart separable convolution [16], Graphical processing unit (GPU) primarily based totally convolutional version layout [17], facts optimization the use of CNN prototyping [18], and pre-educated CNN fashions [19] with their overall performance assessment are mentioned. These fashions help in high-quality tuning inner computations thru parallel processing & pipelining methods, thereby enhancing overall performance of mixed CNN version. Further extensions to those fashions are mentioned in [20, 21, 23, 23], in which Differentiable Neural Architecture Search (DNAS), configurable CNN, PYNQ board precise CNN layout, and speedy Fourier transform (FFT) primarily based totally green convolutions are proposed. These fashions intention at exploring optimization alternatives for enhancing current version overall performance thru garage reduction, and growth in variety of fanouts for the given CNN version. These fashions are carried out to a huge form of packages including, handwriting recognition [24], asynchronous & synchronous layout [25], and embedded CNN accelerators for low strength gadget getting to know deployments [26], because of which applicability of the proposed version is increased. Thus, it could be found that a huge form of CNN modelling processes are proposed via way of means of researchers, and every of them have their personal nuances, advantages, limitations, and destiny scopes. Based on those observations, a parallel CNN version is proposed withinside the subsequent section, which assists in lowering computational complexity thru use of excessive-velocity and occasional strength microunits, that help in green CNN version layout for real-time deployments.

DESIGN OF A PROPOSED CNN MODEL BASED ON FPGA

From the evaluation it could be found that a huge form of gadget fashions are to be had for CNN layout, and every of those fashions range in phrases of postpone, area, and strength requirements. It is in addition found that each CNN version calls for layout of convolutional unit, most pooling unit, activation unit, and absolutely related neural community unit. In this section, parallel layout of those units, in conjunction with their inner running is mentioned in detail. Overall waft of the proposed version is defined in determine 2, in which connection of various convolutional units, and the very last absolutely related neural community (FCNN) unit may be found. Here, 2 convolutional layers are depicted for simplicity, and may be prolonged to 'N' extraordinary layers relying upon community's requirements. Each layer estimates a huge set of features, which may be evaluated the use of the subsequent equation 1, wherein enter facts is improved with a rectilinear unit (ReLU), for augmentation of function vectors. This augmentation assists in comparing a big variety of function units for any given enter facts, thereby comparing more than one function units for a given enter facts frame.

$$Conv_{out_{i,j}} = \sum_{a=-\frac{m}{2}}^{\frac{m}{2}} \sum_{b=-\frac{n}{2}}^{\frac{n}{2}} IN(i-a, j-b) * ReLU\left(\frac{m}{2}+a, \frac{n}{2}+b\right) \ldots (1)$$

Where, IN represents input data, a, b represents size for strides in the given convolutional layer, while m, n indicates current window size for given convolutional layer.

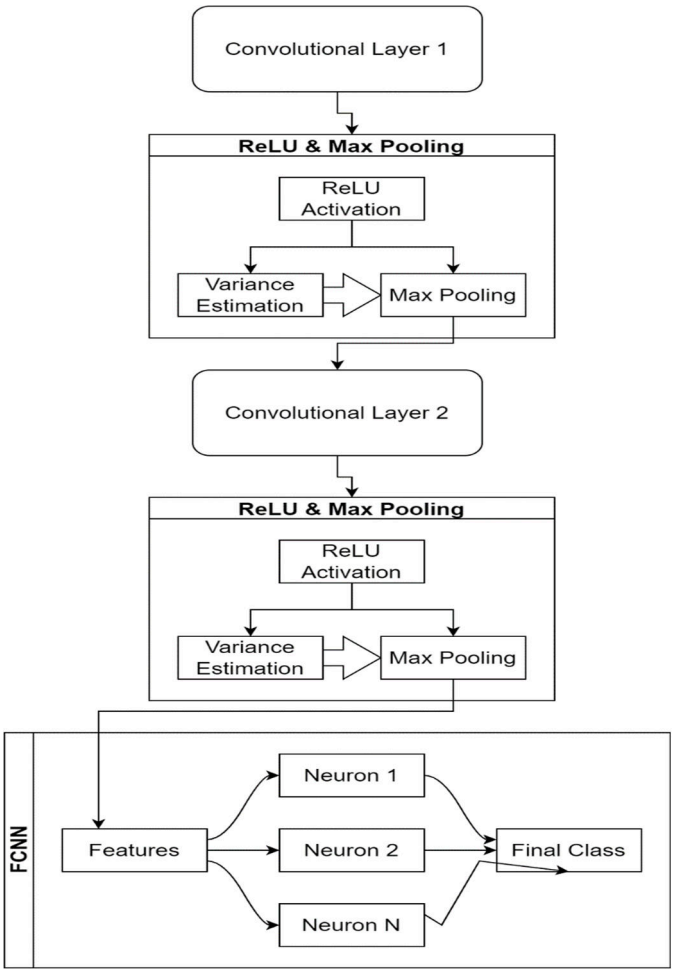

Fig. 2 Overall flow of the proposed model

These input values are complex with a Direct Unit Enable (ReLU) function. This is done in order to improve the efficiency of feature extraction and selection, through feature fill and stride operations as observed by equation 2,

$$ReLU = \frac{C_{in} + 2 * p - k}{s} + 1 \ldots (2)$$

Where, C_in represents the convolution features input to the system, s represents the convolution step size, p represents the padding added to the convolution, and k represents the convolution kernel size. The extracted features are given at a maximum level of clustering, which makes it possible to find highly variable feature sets from the extracted convolutional features. To perform this activity, this model evaluates the between-class variance between the different convolutional features and uses this comparison for the evaluation. of the final variance. This evaluation can be done using Equation 3, which finds a feature variance vector with extracted feature sets,

$$V_{TH} = \frac{\sum_{i=1}^{m} Conv_i - \frac{\sum_{a=1}^{n} Conv_a}{n}}{m-1} \ldots (3).$$

Where m represents the total of the current characteristics, n represents the total of the characteristics in the other classes and Conv_i represents the convolutional features extracted using equations 1 and 2 based on the feature pool. These features are re-evaluated for each convolution level and assigned to a SoftMax level for final classification. The design of this level can be observed from FIG. 2, wherein the implementation of Sum of Products (SOP) is performed to obtain the final output class.

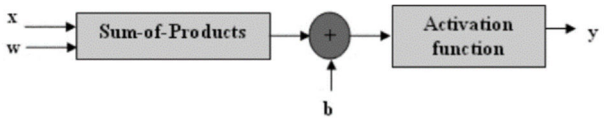

Fig. 3 SoP implementation for final classification

Results of this model are controlled using equation 4, wherein output is controlled via different bias fields.

$$y_{out} = SoftMax\left(\sum_{i=1}^{N_f} x_i * w_i + b\right) \ldots (4)$$

Where, x_i represents input feature vector, w_i represents weight value for given feature vector, b represents feature bias value, and N_f represents total number of convolutional features extracted by the given model. Class with maximum value of y_{out} is selected as the final category for input features. From these operations, it is observed that the proposed model requires design of the following components,

- Arithmetic unit with addition, multiplication, and division operation design
- Memory unit for storage of components
- Neuron design for SoP operation

Design of each of these units is described in different sub-sections of this text, which can be referred by readers in part(s) or as a whole for deploying the models for their designs.

CONCLUSION

In this research paper, decorate ordinary overall performance of hardware-level CNN designs, this textual content material proposes a distinctly surroundings pleasant parallel CNN model layout for FPGA based totally absolutely deployments.

REFERENCES

[1]. C. Huang, S. Ni and G. Chen, "A layer-based structured design of CNN on FPGA," 2017 IEEE 12th International Conference on ASIC (ASICON), 2017, pp. 1037-1040, doi: 10.1109/ASICON.2017.8252656.

[2]. J. Mu, W. Zhang, H. Liang and S. Sinha, "A Collaborative Framework for FPGA-based CNN Design Modeling and Optimization," 2018 28th International Conference on Field Programmable Logic and Applications (FPL), 2018, pp. 139-1397, doi: 10.1109/FPL.2018.00032.

[3]. L. Xie, X. Fan, W. Cao and L. Wang, "High Throughput CNN Accelerator Design Based on FPGA," 2018 International Conference on Field-Programmable Technology (FPT), 2018, pp. 274-277, doi: 10.1109/FPT.2018.00052.

[4]. B. -S. Yu, Y. Tsao, S. W. Yang, Y. K. Chen and S. -Y. Chien, "Architecture Design of Convolutional Neural Networks for Face Detection on an FPGA Platform," 2018 IEEE International Workshop on Signal Processing Systems (SiPS), 2018, pp. 88-93, doi: 10.1109/SiPS.2018.8598428.

[5]. S. Zeng et al., "An Efficient Reconfigurable Framework for General Purpose CNN-RNN Models on FPGAs," 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), 2018, pp. 1-5, doi: 10.1109/ICDSP.2018.8631880.

[6]. X. Chen, J. Ji, S. Mei, Y. Zhang, M. Han and Q. Du, "FPGA Based Implementation of Convolutional Neural Network for Hyperspectral Classification," IGARSS 2018 - 2018 IEEE International Geoscience and Remote Sensing Symposium, 2018, pp. 2451-2454, doi: 10.1109/IGARSS.2018.8517973.

[7]. Y. Duan, S. Li, R. Zhang, Q. Wang, J. Chen and G. E. Sobelman, "Energy-Efficient Architecture for FPGA-based Deep Convolutional Neural Networks with Binary Weights," 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), 2018, pp. 1-5, doi: 10.1109/ICDSP.2018.8631596.

[8]. J. -W. Chang and S. -J. Kang, "Optimizing FPGA-based convolutional neural networks accelerator for image super-resolution," 2018 23rd Asia and South Pacific Design Automation Conference (ASP-DAC), 2018, pp. 343-348, doi: 10.1109/ASPDAC.2018.8297347.

[9]. F. Liang, Y. Yang, G. Zhang, X. Zhang and B. Wu, "Design of 16-bit fixed-point CNN coprocessor based on FPGA," 2018 IEEE 23rd International Conference on Digital Signal Processing (DSP), 2018, pp. 1-5, doi: 10.1109/ICDSP.2018.8631564.

[10]. H. Nakahara and T. Sasao, "A High-speed Low-power Deep Neural Network on an FPGA based on the Nested RNS: Applied to an Object Detector," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018, pp. 1-5, doi: 10.1109/ISCAS.2018.8351850.

[11]. W. Xie, C. Zhang, Y. Zhang, C. Hu, H. Jiang and Z. Wang, "An Energy-Efficient FPGA-Based Embedded System for CNN Application," 2018 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC), 2018, pp. 1-2, doi: 10.1109/EDSSC.2018.8487057.

[12]. T. Geng, T. Wang, A. Sanaullah, C. Yang, R. Patel and M. Herbordt, "A Framework for Acceleration of CNN Training on Deeply-Pipelined FPGA Clusters with Work and Weight Load Balancing," 2018 28th International Conference on Field Programmable Logic and Applications (FPL), 2018, pp. 394-3944, doi: 10.1109/FPL.2018.00074.

[13]. K. S., D. Paul, B. R. Jose and N. S., "Design Space Exploration of Convolution Algorithms to Accelerate CNNs on FPGA," 2018 8th International Symposium on Embedded Computing and System Design (ISED), 2018, pp. 21-25, doi: 10.1109/ISED.2018.8704043.

[14]. R. Zhao, H. Ng, W. Luk and X. Niu, "Towards Efficient Convolutional Neural Network for Domain-Specific Applications on FPGA," 2018 28th International Conference on Field Programmable Logic and Applications (FPL), 2018, pp. 147-1477, doi: 10.1109/FPL.2018.00033.

[15]. K. Guo et al., "Angel-Eye: A Complete Design Flow for Mapping CNN onto Customized Hardware," 2016 IEEE Computer Society Annual Symposium on VLSI (ISVLSI), 2016, pp. 24-29, doi: 10.1109/ISVLSI.2016.129.

[16]. L. Bai, Y. Zhao and X. Huang, "A CNN Accelerator on FPGA Using Depthwise Separable Convolution," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 65, no. 10, pp. 1415-1419, Oct. 2018, doi: 10.1109/TCSII.2018.2865896.

[17]. L. Kang, H. Li, X. Li and H. Zheng, "Design of Convolution Operation Accelerator based on FPGA," 2020 2nd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI), 2020, pp. 80-84, doi: 10.1109/MLBDBI51377.2020.00021.

[18]. W. Hu, S. Chen, Z. Li, T. Liu and Y. Li, "Data Optimization CNN Accelerator Design on FPGA," 2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom), 2019, pp. 294-299, doi: 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00051.

[19]. D. T. Kwadjo, J. M. Mbongue and C. Bobda, "Performance Exploration on Pre-implemented CNN Hardware Accelerator on FPGA," 2020 International Conference on Field-Programmable Technology (ICFPT), 2020, pp. 298-299, doi: 10.1109/ICFPT51103.2020.00055.

[20]. H. Fan, M. Ferianc, S. Liu, Z. Que, X. Niu and W. Luk, "Optimizing FPGA-Based CNN Accelerator Using Differentiable Neural Architecture Search," 2020 IEEE 38th International Conference on Computer Design (ICCD), 2020, pp. 465-468, doi: 10.1109/ICCD50377.2020.00085.

[21]. H. V. Phu, T. Minh Tan, P. Van Men, N. Van Hieu and T. Van Cuong, "Design and Implementation of Configurable Convolutional Neural Network on FPGA," 2019 6th NAFOSTED Conference on Information and Computer Science (NICS),2019, pp. 298-302, doi: 10.1109/NICS48868.2019.9023810.

[22]. M. Dhouibi, A. K. Ben Salem and S. B. Saoud, "CNN for object recognition implementation on FPGA using PYNQ framework," 2020 IEEE Eighth International Conference on Communications and Networking (ComNet), 2020, pp. 1-6, doi: 10.1109/ComNet47917.2020.9306094.

[23]. L. He, X. Xie, J. Lin and Z. Wang, "Efficient FPGA design for Convolutions in CNN based on FFT-pruning," 2020 IEEE Asia Pacific Conference on Circuits and Systems (APCCAS), 2020, pp. 27-30, doi: 10.1109/APCCAS50809.2020.9301653.

[24]. R. Xiao, J. Shi and C. Zhang, "FPGA Implementation of CNN for Handwritten Digit Recognition," 2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2020, pp. 1128-1133, doi: 10.1109/ITNEC48623.2020.9085002.

[25]. H. Kato and H. Saito, "Design of Asynchronous CNN Circuits on Commercial FPGA from Synchronous CNN Circuits," 2019 IEEE 13th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC), 2019, pp. 61-67, doi: 10.1109/MCSoC.2019.00016.

[26]. B. Khabbazan and S. Mirzakuchaki, "Design and Implementation of a Low-Power, Embedded CNN Accelerator on a Low-end FPGA," 2019 22nd Euromicro Conference on Digital System Design (DSD), 2019, pp. 647-650, doi: 10.1109/DSD.2019.00102.