July 2021

Bibliometric Review of FPGA Based Implementation of CNN

Priti Shahane
Symbiosis International University, pritis@sitpune.edu.in

Piyush Tyagi
Symbiosis International University, piyush.tyagi.btech2018@sitpune.edu.in

Purba Saha
Symbiosis International University, purba.saha.btech2018@sitpune.edu.in

Shravan Sainath
Symbiosis International University, sainath.shravan.btech2018@sitpune.edu.in

Swanand Bedekar
Symbiosis International UniversityConvolution Neural Network, Hardware accelerators, GPU, ASIC, FPGA, Parallelism and pipelining architecture.Convolution Neural Network, Hardware accelerators, GPU, ASIC, FPGA, Parallelism and pipelining architecture., swanand.bedekar.btech2018@sitpune.edu.in

Bibliometric Review of FPGA Based Implementation of CNN

Piyush Tyagi[1], Purba Saha[2], Sainath Shravan Lingala[3], Swanand Bedekar[4], Priti Shahane[5]

[1]Department of Electronics and Telecommunications, Symbiosis Institute of Technology

Affiliated to Symbiosis International (Deemed University), Pune, India

Email: piyush.tyagi.btech2018@sitpune.edu.in

[2]Department of Electronics and Telecommunications, Symbiosis Institute of Technology

Affiliated to Symbiosis International (Deemed University), Pune, India

Email: purba.saha.btech2018@sitpune.edu.in

[3]Department of Electronics and Telecommunications, Symbiosis Institute of Technology

Affiliated to Symbiosis International (Deemed University), Pune, India

Email: sainath.shravan.btech2018@sitpune.edu.in

[4]Department of Electronics and Telecommunications, Symbiosis Institute of Technology

Affiliated to Symbiosis International (Deemed University), Pune, India

Email: swanand.bedekar.btech2018@sitpune.edu.in

[5]Assistant Professor, Department of Electronics and Telecommunications, Symbiosis Institute of Technology,

Affiliated to Symbiosis International (Deemed University), Pune, India

Email: priti.shahane@sitpune.edu.in

ABSTRACT

Nowadays Convolution Neural Network (CNN) has become the state of the art for machine learning algorithms due to their high accuracy. However, implementation of CNN algorithms on hardware platforms becomes challenging due to high computation complexity, memory bandwidth and power consumption. Hardware accelerators such as Graphics Processing Unit (GPU), Field Programmable Gate Array (FPGA), Application-Specific Integrated Circuit (ASIC) are suitable platforms to model CNN algorithms. Recently FPGAs have been considered as an attractive platform for CNN implementation. Modern FPGAs have various embedded hardware and software blocks such as a soft processor, DSP slice and memory blocks. These embedded resources along with customized logic blocks, makes FPGA a perfect candidate for CNN model. Also, the major advantage of FPGA in the case of CNN is its parallelism and pipelining architecture which helps to accelerate CNN operations. The primary goal of this bibliometric review is to determine the scope of current literature in the field of implementing CNN algorithms on various hardware platforms, with a particular emphasis on the FPGA platform for CNN-based applications. Data from Scopus is mostly used in this bibliometric analysis. It reveals that researchers from China, India, and the United Kingdom make the most significant contributions in the form of conferences, journals, and book proceedings. All the documents are from subject areas of Engineering, Computer Science, Mathematics, Physics and Astronomy, Decision Sciences, and Material Science make significant contributions.

Keywords: Convolution Neural Network, Hardware accelerators, GPU, ASIC, FPGA, Parallelism and pipelining architecture.

1. INTRODUCTION

Significant amounts of unstructured data have resulted from the machines that can generate and consume information. In context to this, The area of deep learning has been resurrected as a result of the rise in reliable data in the form of audios and documents, and it now seeks to provide techniques for the automated recovery of valuable information and trends from data. In the past few years, The modern use of deep learning algorithms such as CNN, which is now a test ground in various fields of science, has grown at an incredibly rapid rate in the world of technology [1].

CNNs are algorithms with properties such as low power consumption and easy reconfigurability. They have a significant impact on society due to their various applications in problem-solving. CNNs have proved to showcase higher accuracies and good performance in real time applications compared to the traditional approaches [2-3].

With the widespread use of CNN, it has given the industries and researchers the advantage of being able to resolve issues in various domains where knowledge is not expressed explicitly, and implicit

information is stored in raw data. CNN finds its application in various domains such as Agriculture, Telecom, robotics, military services, etc, with some of the applications such as image classification, speech processing tasks, video surveillance, mobile robot vision, facial recognition, robotic vision, smart camera technologies and many more. Therefore, the high efficiency of CNN on embedded systems necessitates more stringent implementation criteria [2], [4-7][20].

1.1 Generic Convolution Neural Networks

CNNs are classified as feed forward networks which consist of one or more layers such as Convolution layer, Pooling layer, Fully connected layers. In this architecture the data passed through the input are processed by the different hidden layers and neurons and finally at the end the classification results are presented as the output [3].

These layers are explained below: [2-3],[8-9],[15-18][20]

- **Convolutional Layer**

  This layer can be used to retrieve either the input image's features or the upper layer's output attribute map data. The procedure is a two-dimensional convolution based on input data and separate convolution kernels, with the activation function creating a new two-dimensional output method. The convolution process is completed by element wise multiplication and accumulation. There are further operations that are required to be performed on the image during convolution like Padding and Striding. Padding is essentially adding an additional set of rows and columns at all sides of the image matrix. It only consists of ‘0’ hence, it does not actually supply any information. Following the traditional method, the filter is allowed to shift by just one row or column at a time. But with the utilization of Strided Convolution, the filter now has the flexibility to shift 2 or 3 rows/columns at each step. This leads to the reduced calculation which can come in handy within the case of huge images.

- **Pooling Layer**

  By reducing the function map's dimensions and network computing complexity, the pooling layer effectively avoids overfitting. The pooling layer is usually inserted between 2 successive convolutional blocks within the CNN structure because it helps in reducing the number of parameters and computation within the network. The subsampling layer is sandwiched between two convolutional layers, and its main purpose is to shrink the data size of the different feature maps while keeping the associated functions. It's normally implemented as a max-pooling algorithm, which replaces each input submatrix with its greatest value or average, respectively.

- **Fully Connected Layer**

  The high-level two-dimensional feature map taken from the previous convolutional layer is transformed into a one-dimensional feature map provided by this layer, which is usually located at the end of the convolutional neural network. There is no weight distribution in the fully connected layer because each of the neurons is linked to every other neuron in the preceding layer. These weights come from the training process, and the vector is a set of features extracted by the CNN's function extractor. The output vector is called fully connected since each object is a weighted sum of the input vector. The conceptual design's key goal is to ensure that this layer is faster than the applications. This leads to the reduced calculation which can come in handy within the case of huge images.

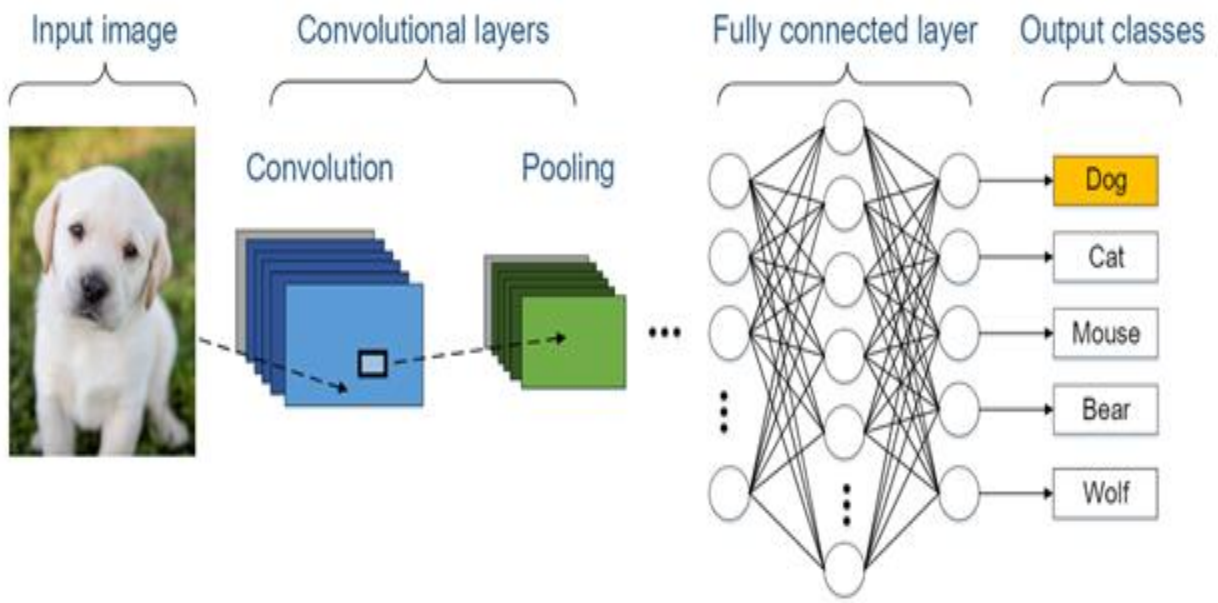

Figure1: Demonstration of a (CNN) layers using an image,[3]

1.2 Challenges of traditional CNN approaches

Despite its rising popularity, CNN approaches come with challenges and limitations such as high-cost computational complexity, high amount of computational resources, greater memory access and large amount of power consumption which pose greater challenges to the design. Also with the advancement in technologies, the difficulty of the CNN algorithms keep expanding as new applications constantly rise, which further demands for higher accuracy that is fulfilled by general purpose processors. To keep up with the demand, researchers are forced to find alternatives to the traditional approaches [1-2], [10][20].

1.3 Overcoming the challenges using FPGA, GPU and ASIC

Until now most of the implementations were through software interface but recent studies show that these limitations and implications faced by software such as its speed of execution due to software being sequential in nature can be addressed through hardware implementations. In comparison to software-based implementations, hardware-based implementations exhibit properties such as parallelism more effectively, and can produce the same performance in a smaller number of cycles for the same algorithm using Graphics Processing Unit ((GPU), Field Programmable-Gate Array FPGA) or Application-Specific Integrated Circuit ((ASIC). These hardware based accelerators are found to be not only capable, reliable, and having fast execution speed but also help in increasing the efficiency as well as effectiveness of CNN [1-2],[8].

The results of hardware-based accelerators such as FPGA, GPU and ASIC have been nifty in improving CNN based applications. However, among the three, ASIC designs on the other hand have achieved higher throughput with lower power consumptions but have large development cost and time as compared to other solutions. Compared to application-specific integrated circuit (ASIC), the design cycle of FPGA is shorter and FPGAs are reconfigurable, customizable, and efficient, due to this, many efficient networking structures have been proposed that decrease the computational complexity and researchers are concentrating on FPGA based CNN hardware accelerated implementation. By parameter tuning, an accelerator can accommodate network layers of various sizes, optimize latency, and achieve maximum pipeline by using a data stream interface, making it ideal for CNN deployment [2][8][16][21].

1.4.1 FPGA platform for CNN

FPGA is made by a combination of fixed arrays of logic blocks which are connected by programmable interconnects. This provides the properties such as high logic density, enhanced parallelism efficiency, wafer processing, reprogram ability and high efficiency. The characteristics and key techniques pertaining to parallelism of layers and pipelining technique within several of the layers, which speed up the execution tasks and processes, make FPGA well suited and favored. These properties help in mapping different layers on one single chip and thus enabling achievement of high performance and greater resource utilization [2][11][19].

With the increase in operations in CNN models it becomes a matter of time when general purpose processors are of no use to us. This is where FPGA comes into the picture. FPGA provides flexibility and a good amount of efficiency which makes it a solid candidate as the hardware accelerator [10].

1.5 CNN implementation on FPGA

Examples of FPGA implementation of CNN over the years include Bell Labs' 2D Convolution and Matrix-vector multiplication, Float Point CNN implementation, and Fixed Point CNN for real-time video processing [13]. This section describes the process of Implementation of CNN on FPGA which is first achieved by building and designing a CNN network on a Computer platform using pertinent softwares. This is done through programming a High Level Language which is used for training and testing the data sets. These layers and values are then stored over the FPGA in a memory which can be called immediately for real time application and execution. The example below shows how CNN can be implemented on an FPGA for handwritten character recognition. This flowchart explains in detail the process and working of CNN explaining the step-by-step procedure and various layers involved. The input data is provided from MNIST dataset which contains the binary images of handwritten digits. It has a set of seventy thousand images out of which sixty thousand are used as training sets while the rest of ten thousand images are used as test sets.

The sixty thousand images used in the training set are trained by writing and implementing the code in python. This training is achieved by providing information in the code such as number of layers, number of neurons, type of activation code etc. After this process is completed, generated weights and biases are stored in memory which is a one-time process. The images in MNIST dataset are of the size 28x28. This results in the count of pixels in these images as 784, thus the initial layer contains 784 neurons. The output of the data processed in the first layer is then sent to the next layer where similar operations are performed. Inside the neuron the input data is multiplied with the weights values which are pre calculated and stored in the memory which enables in real time implementation. The result value obtained is stored in a variable “mul” whose value is added with the existing or previous cycle’s value in the neuron. The obtained value is again sent back in the loop which is added with the next ‘mul’ value (value obtained by the product of the next new data value and weight). Finally at the end of summation, the final obtained result value is added with the bias value. This value now moves towards the activation function which can be linear or non- linear in nature. In this project we will prefer a non-linear activation function since most of the applications in the practical world are nonlinear in nature. Hence, these activation functions are either sigmoid or ReLu. The value obtained from the activation function is the output of the neuron which is now the input of the next neuron. This process keeps on continuing until the final layer has processed the data and provides us with the required output value by recognizing the digit/ character.

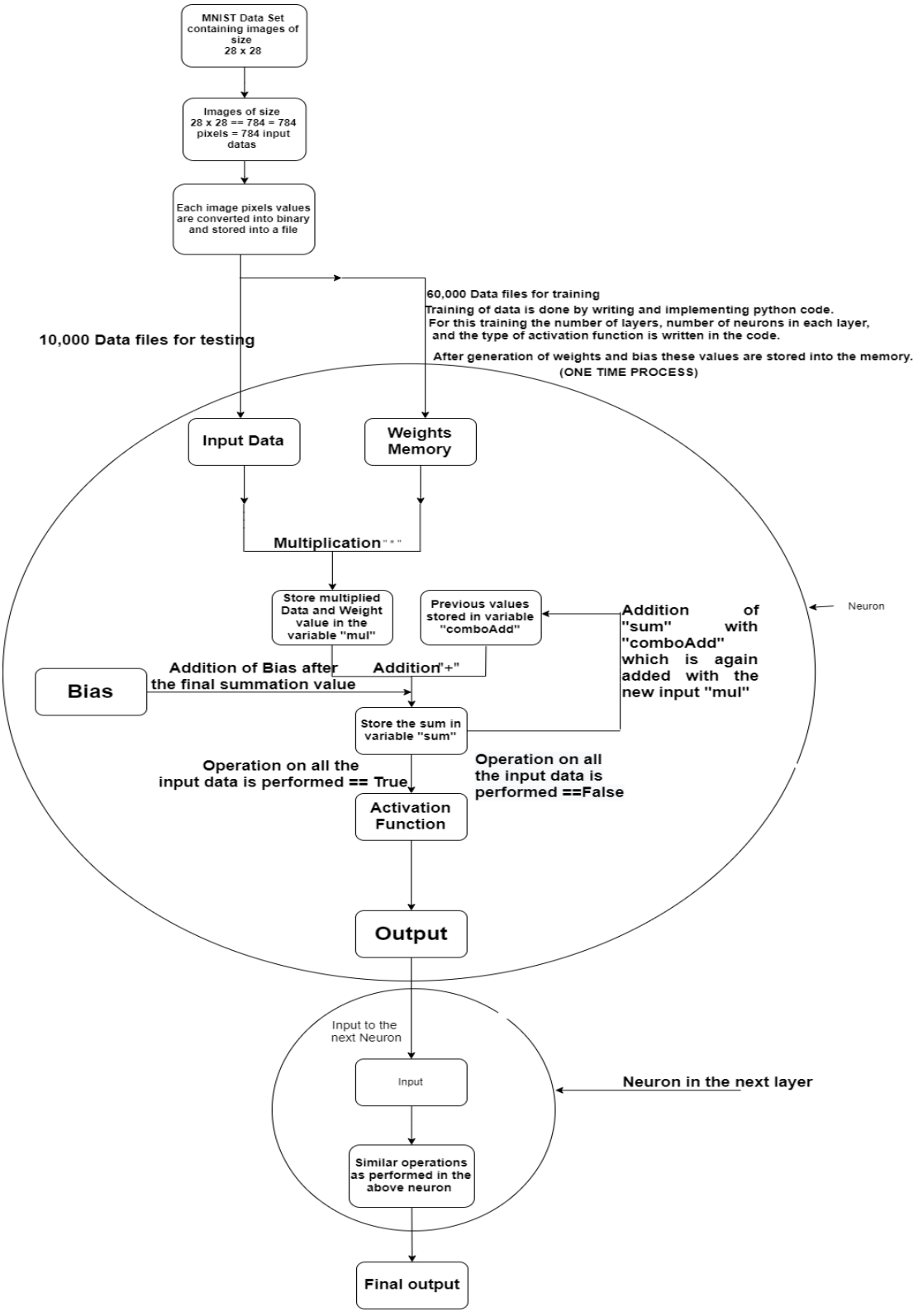

Figure 2: Working of neural networks

2. DATA COLLECTION IN A PREPARATORY STAGE

The data for this paper was gathered from Scopus. For this, various databases such as Scopus, Clarivate, Web of Science, etc. were used. This preliminary data was collected on 11th of May 2021 using Scopus database. We chose this database because it is one of the most popular and comprehensive peer-reviewed databases available. The following section contains a list of the terminologies that were included.

2.1 Keyphrases that were utilised

Top twenty key phrases associated with the analysis of our study are present in table number 1. Important keyphrases used among those were “Convolutional Neural Networks (CNN)” and “ Field Programmable Gate Arrays (FPGA)” for conducting the search. Table 1 lists additional related keywords as well as the publication number.

Table 1: Publication Number & Keyword

Keyword	Publications Number
Field Programmable Gate Arrays (FPGA)	369
Convolutional Neural Network	219
Integrated Circuit Design	106
Energy Efficiency	98
Deep Learning	87
FPGA Implementations	78
Image Processing	39
System on Chip	39
Hardware Accelerators	38

Memory Architecture	35
Graphics Processing Unit	35
Character Recognition	25
Pipeline Processing Systems	25
Image Recognition	23
Low Power Consumption	20
Energy Efficient	19
HLS	9
HandWritten Digit Recognition	8
Parallel Processing	7
Reconfigurability	7

Source: http://www.scopus.com (accessed on 11th May 2021)

The study done in this paper is limited to the language of English and based on the search results, there are 470 English publications available on this subject as depicted in table 2.

Table 2: Publishing language trends

Language of Publications	Publication Number
English	470
Chinese	4
Turkish	2

Source: http://www.scopus.com (accessed on 11th May 2021)

According to the table below, 65.126 percent of researchers in this field have published their papers in conference proceedings, 26.050 percent in journals, 8.613 percent in book series, and 0.210 percent in books.

Table 3: Publication percentage and type

Type	Publication Number	Percentage of 476
Conference Proceeding	310	65.126 %
Journal	124	26.050 %
Book Series	41	8.613 %
Book	1	0.210 %

Source: http://www.scopus.com (accessed on 11th May 2021)

2.2 Highlights of previous analysis

476 publications were retrieved here using the keywords mentioned in the above section. The preliminary research is solely based on the Scopus Database. For the world of Convolution Neural Networks, various kinds of publications from the conference proceedings, journal articles, book series, and books are spanned between years 2010 to 2021. The below table 4 represents per year publication count in the last twelve years and the analysis of publication per year as depicted in figure 3.

Table 4: Annual number of publications

Year	Number of Publications
2021	30
2020	122
2019	137
2018	92
2017	45
2016	19

2015	7
2014	2
2013	6
2012	6
2011	4
2010	6

Source: http://www.scopus.com (accessed on 11th May 2021)

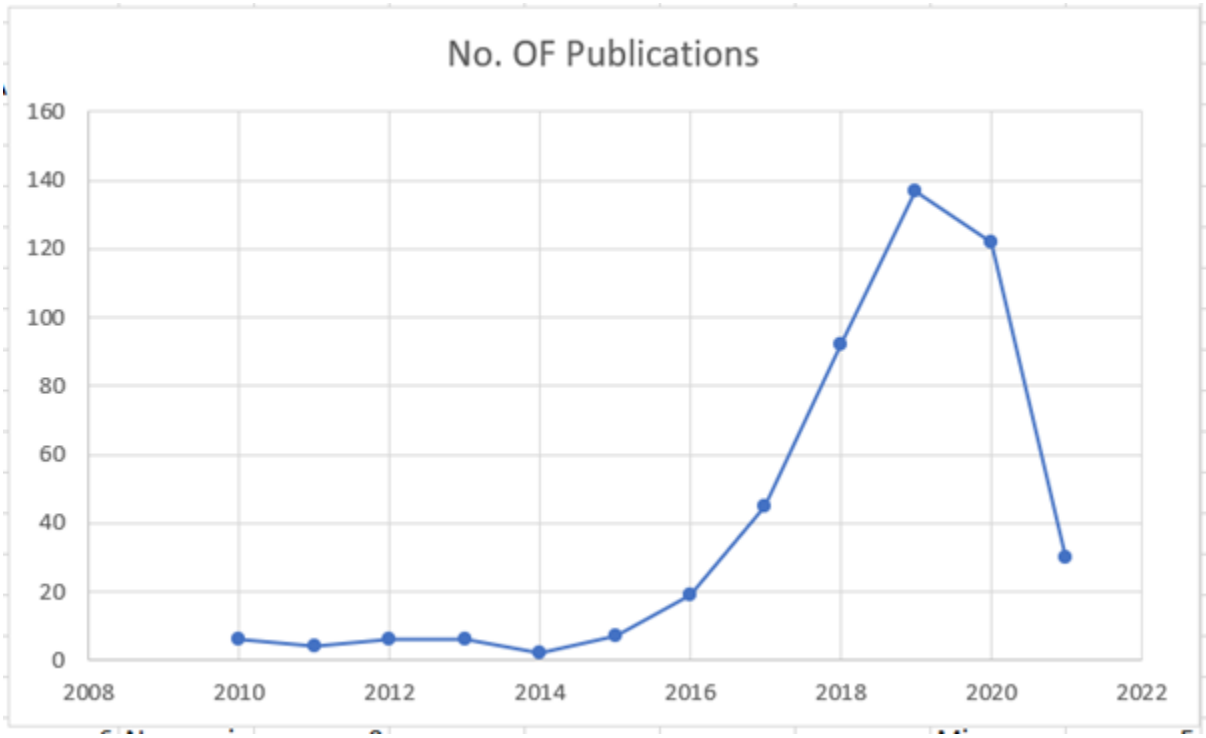

Figure 3: Trends in publication per annum

Source: http://www.scopus.com (accessed on 11th May 2021)

3. BIBLIOMETRIC ANALYSIS

The bibliometric survey uses geographical distribution to display the types of literature available in this research field, as well as how different writers from various geographical locations have contributed through publication and association statistics.

This bibliometric analysis is carried out via two different ways

- Analysis via Geographical position
- Statistical analysis of the documents using affiliations, type of document, statistics about the authors, and citation analysis.

3.1 Geographical location-based assessment

This study of research work in the field of CNN and FPGA based on geographical locations is depicted in Figure 4 and Figure 5. This is done using the Tableau Software to create these figures. The countries with the most publications are China, India and the United Kingdom, followed by Germany, Australia, Singapore and Iran.

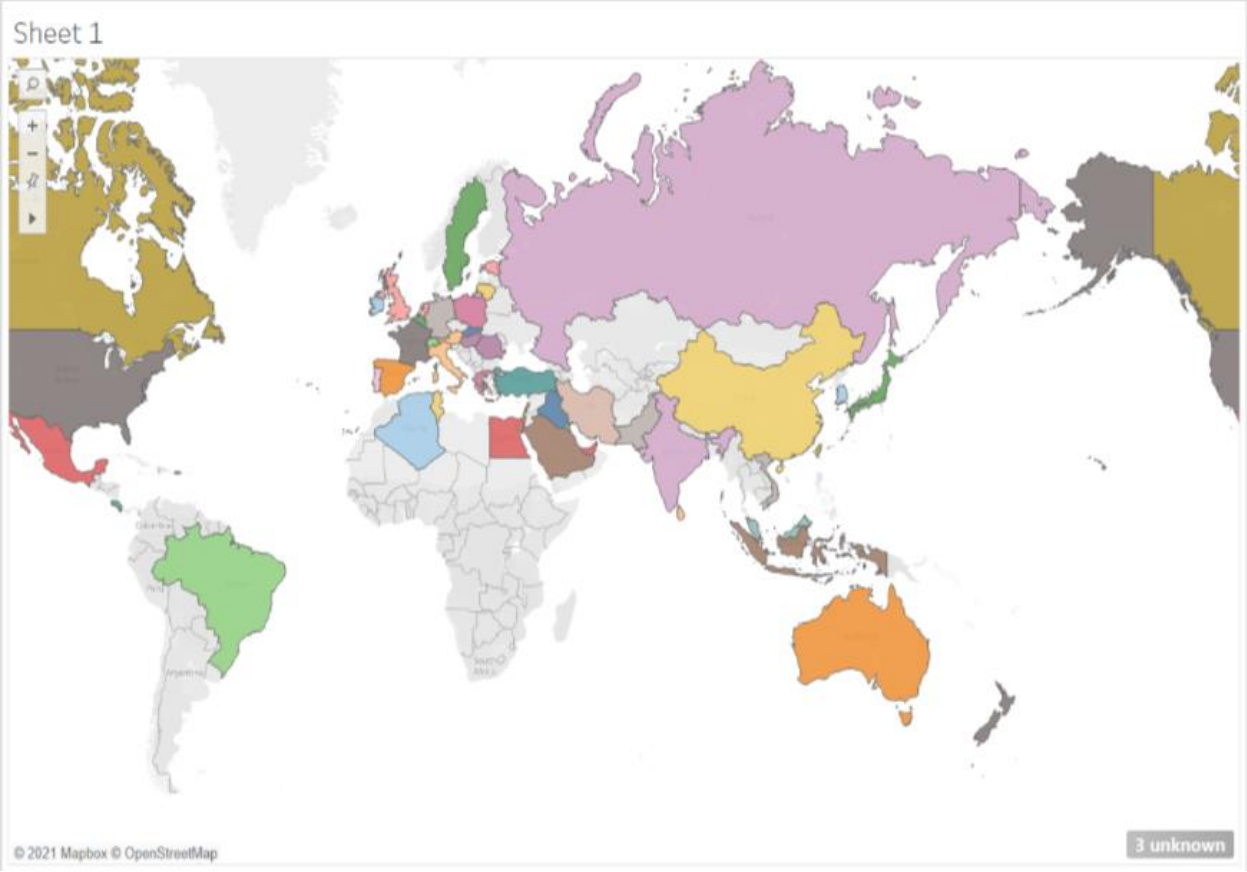

Figure 4: Geographical locations based analysis for study of CNN and FPGA

Source: http://www.scopus.com (accessed on 11th May 2021)

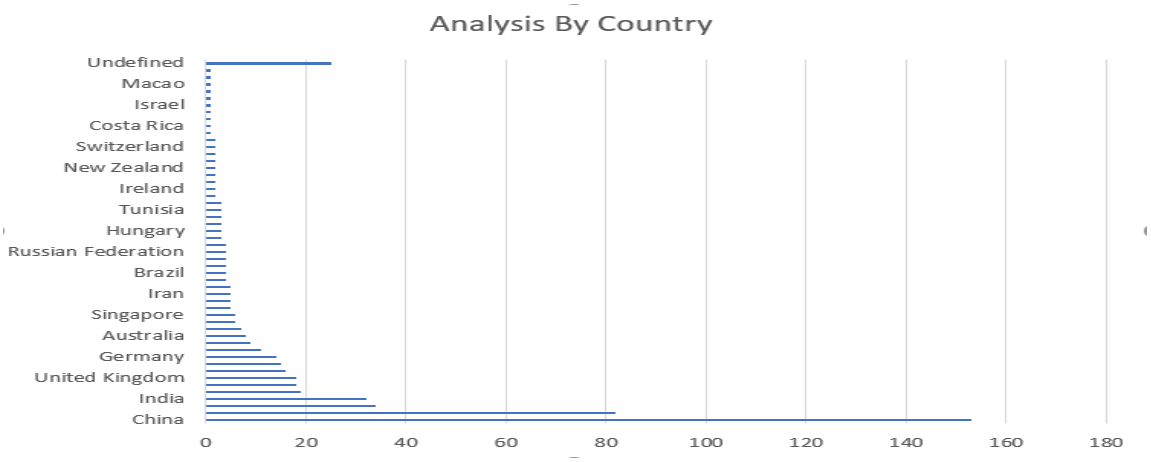

Figure 5: Country Specific Analysis

Source: http://www.scopus.com (accessed on 11th May 2021)

3.2 Analyses based upon subject matter

Figure 6 depicts the study depending on the subject field. The majority of the research is focused on Computer Science, Engineering, Mathematics, Physics, and Astronomy, as well as Decision Sciences and Material Science. Whereas the publications from the fields of Energy, NeuroScience, Medicine is relatively low.

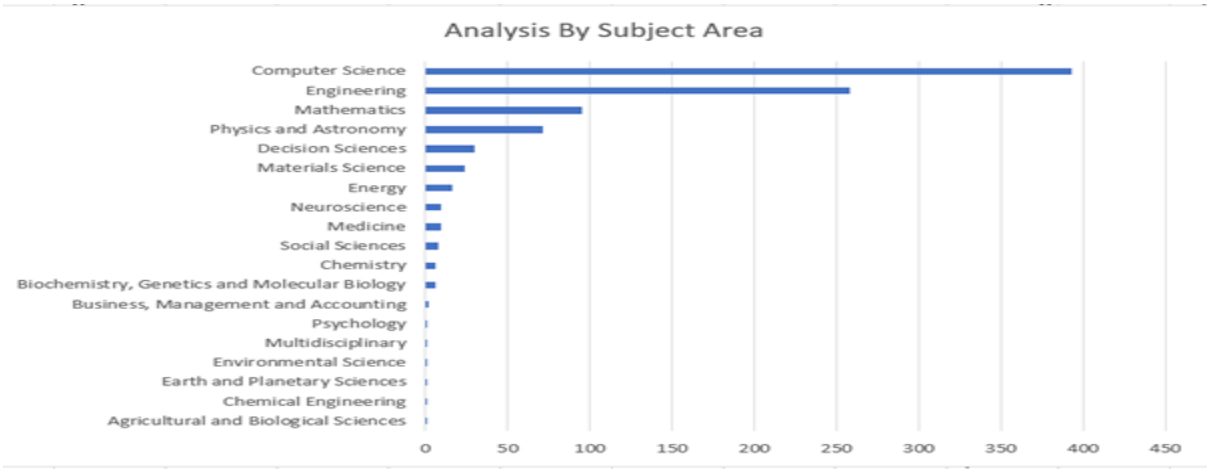

Figure 6: Analysis-based upon subject area

Source: http://www.scopus.com (accessed on 11th May 2021)

3.3 Affiliation-based analysis:

As shown in figure 7, many universities from around the world have made major contributions to research in the field of CNN and FPGA. Universities in London and Los Angeles dominate this field of research. Figure 7 depicts the top twenty three affiliations.

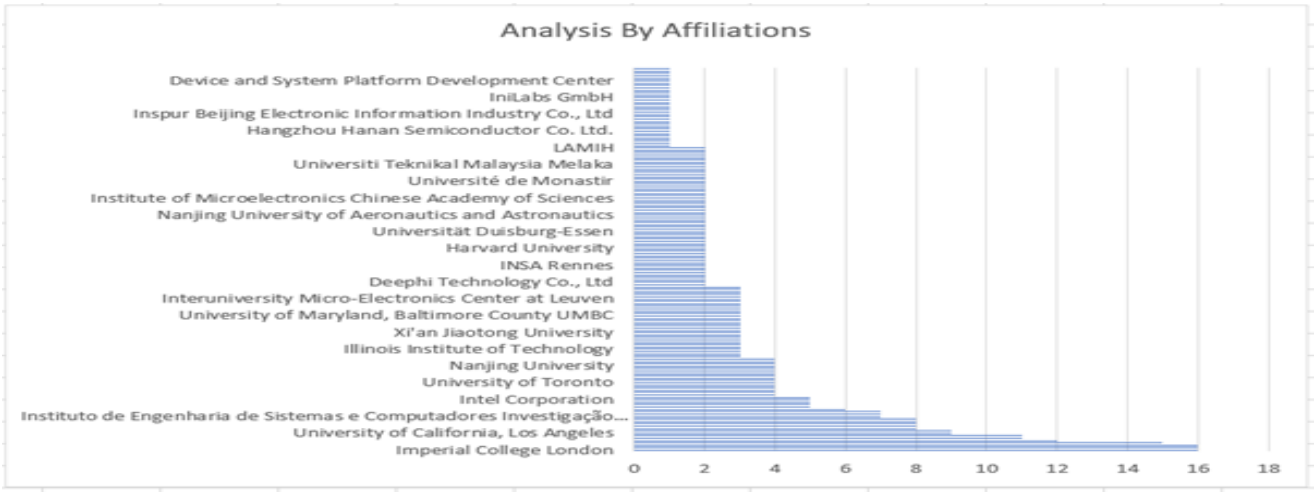

Figure 7: Affiliation-based analysis:

Source: http://www.scopus.com (accessed on 11th May 2021)

3.4 Analysis upon number of publications and their authors

Figure 8 depicts researchers in the field of CNN and FPGA. The main scholars in this field are highlighted below.

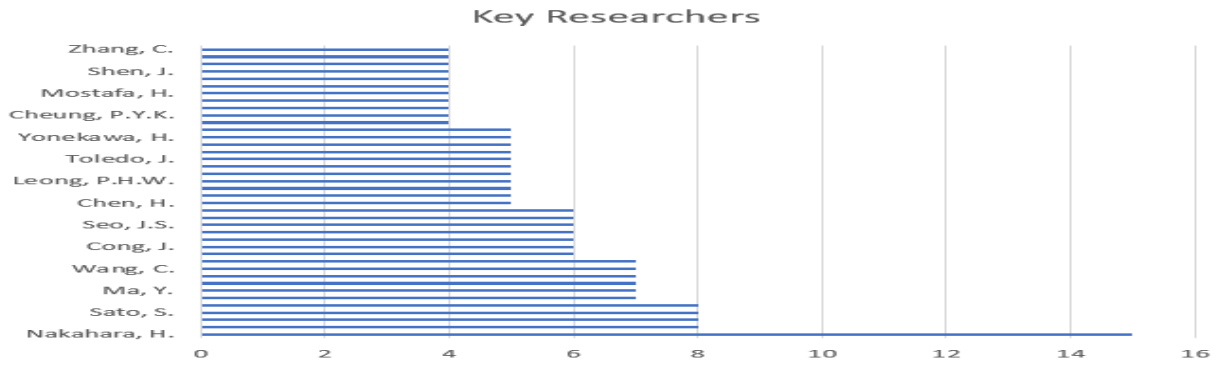

Figure 8: Key Researchers

Source: http://www.scopus.com (accessed on 11th May 2021)

Citation Analysis

The below figure 9 shows the citation analysis on the basis of research done in the field discipline of FPGA based implementation of CNN.

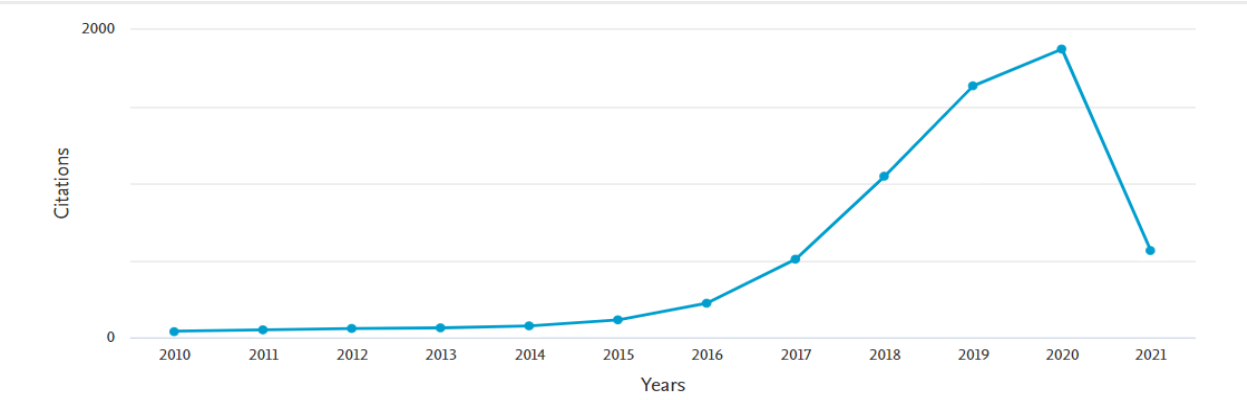

Figure 9: Citation Analysis

Source: http://www.scopus.com (accessed on 11th May 2021)

3.5 Analysis based upon the type of document

Figure 10 depicts the analysis by document form. The figure shows that conference proceedings make up 65% of the publications, journal papers make up 26%, book series make up 8%, and books make up the remaining 1%.

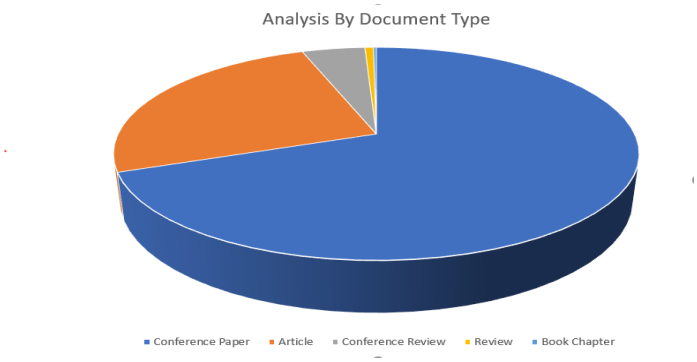

Figure 10: Analysis based by the type of document

Source: http://www.scopus.com (accessed on 11th May 2021)

3.6 Analysis using Clustering

In this research, Clusters are used in this analysis to depict the relationship between different statistical parameters. VOSViewer is a framework for creating clusters that was used for this purpose. In Figure 11 the cluster depicts the relationship between the authors while figure 12 depicts the relation between different keywords.

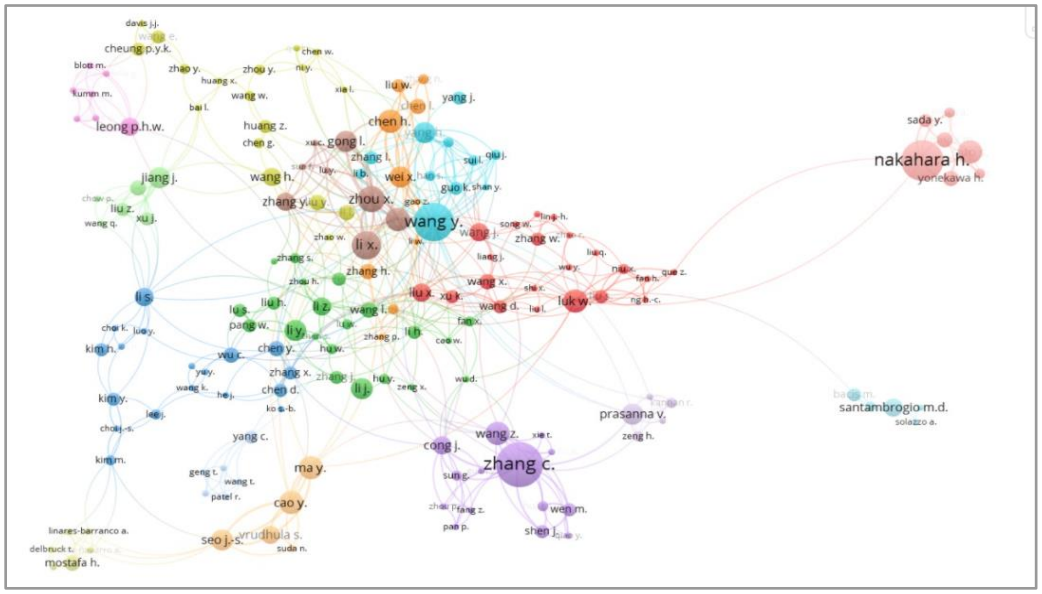

Figure 11: Depicting relation between Authors using cluster

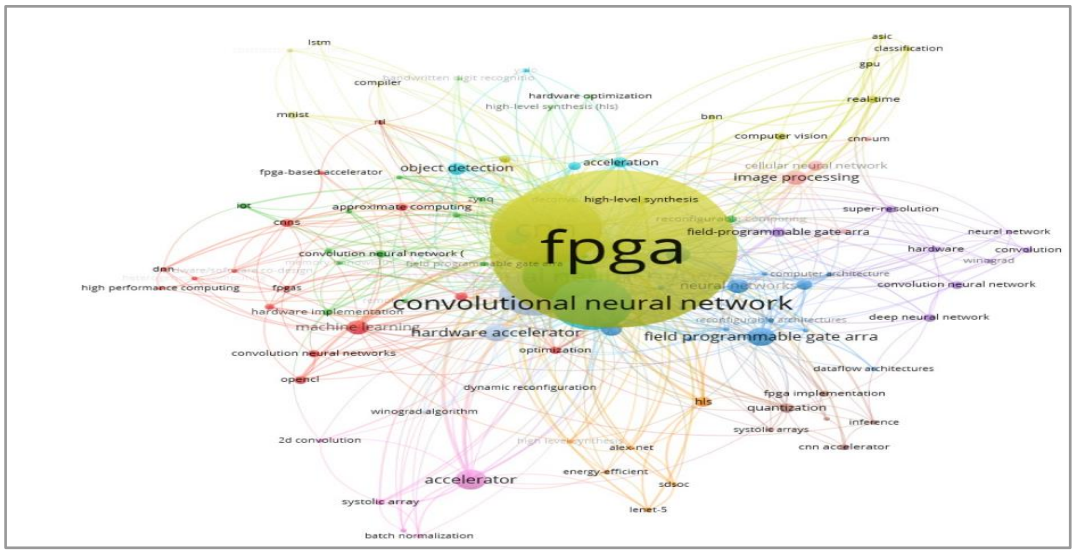

Figure 12: Depicting relation between keywords using cluster

Limitations of the study

For the bibliometric analysis, various keyword combinations were used to search the Scopus database. But this research was unable to include a few relevant papers and articles that were not available in the Scopus database.

Conclusion

Machine learning methods based on convolutional neural networks (CNNs) have been widely employed in artificial intelligence applications for many years. Despite the fact that the computation-intensive, memory-intensive, and resource-intensive characteristics of large-scale CNNs have brought several obstacles to CNN implementations. FPGA based CNN implementation is a significant solution to these challenges. This analysis explains that the field of computer science engineering (CSE) have made a significant contribution in terms of scientific publications in this area. The majority of publications come from Chinese seminars, magazines, and papers, according to this bibliometric review, followed by the United States, Japan, and India. Study researchers who are main research contributors in this field of research will create FPGA-related CNN-implemented applications based on this bibliometric analysis.

REFERENCES:

1 Shawahna, A., Sait, S. M., & El-Maleh, A. (2018). FPGA-based accelerators of deep learning networks for learning and classification: A review. *IEEE Access*, *7*, 7823-7859.

2 Hassan, R. O., & Mostafa, H. (2020). Implementation of deep neural networks on FPGA-CPU platform using Xilinx SDSOC. *Analog Integrated Circuits and Signal Processing*, 1-10.

3 Ghaffari, A., & Savaria, Y. (2020). CNN2Gate: An Implementation of Convolutional Neural Networks Inference on FPGAs with Automated Design Space Exploration. *Electronics*, *9*(12), 2200.

4 Baptista, D., Morgado-Dias, F., & Sousa, L. (2019). A Platform based on HLS to Implement a Generic CNN on an FPGA. In the proceedings of 2019 International Conference in Engineering Applications (ICEA), pp. 1-7.

5 Natale, G., Bacis, M., & Santambrogio, M. D. (2017). On how to design dataflow FPGA-based accelerators for convolutional neural networks. In the proceedings of *2017 IEEE Computer Society Annual Symposium on VLSI (ISVLSI)*, pp. 639-644.

6 Lian, X., Liu, Z., Song, Z., Dai, J., Zhou, W., & Ji, X. (2019). High-performance FPGA-based CNN accelerator with block-floating-point arithmetic. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, *27*(8), 1874-1885.

7 Wang, J., Lin, J., & Wang, Z. (2017). Efficient hardware architectures for deep convolutional neural network. *IEEE Transactions on Circuits and Systems I: Regular Papers*, *65*(6), 1941-1953.

8 Liu, B., Zou, D., Feng, L., Feng, S., Fu, P., & Li, J. (2019). An fpga-based cnn accelerator integrating depthwise separable convolution. *Electronics*, *8*(3), 281.

9 Ma, Y., Suda, N., Cao, Y., Seo, J. S., & Vrudhula, S. (2016). Scalable and modularized RTL compilation of convolutional neural networks onto FPGA. In the proceedings of *2016 26th International Conference on Field Programmable Logic and Applications (FPL)* pp. 1-8.

10 Li, H., Fan, X., Jiao, L., Cao, W., Zhou, X., & Wang, L. (2016). A high performance FPGA-based accelerator for large-scale convolutional neural networks. In the proceedings of *2016 26th International Conference on Field Programmable Logic and Applications (FPL)*, pp. 1-9.

11 Kansal, S., Sikri, M., Gupta, A., & Sharma, M. (2018). A Prospect of Achieving Artificial Neural Networks through FPGA. In the proceedings of *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, pp. 358-363.

12 Xiao, R., Shi, J., & Zhang, C. (2020). FPGA Implementation of CNN for Handwritten Digit Recognition. In the proceedings of *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)* (Vol. 1), pp. 1128-1133.

13 Solovyev, R., Kustov, A., Telpukhov, D., Rukhlov, V., & Kalinin, A. (2019). Fixed-point convolutional neural network for real-time video processing in FPGA. In the proceedings of *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)* pp. 1605-1611.

14 Venieris, S. I., & Bouganis, C. S. (2016). fpgaConvNet: A framework for mapping convolutional neural networks on FPGAs. In the proceedings of *2016 IEEE 24th Annual International Symposium on Field-Programmable Custom Computing Machines (FCCM)*, pp. 40-47.

15 Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. *Neural computation*, *29*(9), 2352-2449.

16 Coşkun, M., Uçar, A., Yildirim, Ö., & Demir, Y. (2017). Face recognition based on convolutional neural network. In the proceedings of 2017 International Conference on Modern Electrical and Energy Systems (MEES), pp. 376-379.

17 Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*, 1097-1105.

18 Qiu, J., Wang, J., Yao, S., Guo, K., Li, B., Zhou, E., ... & Yang, H. (2016). Going deeper with embedded fpga platform for convolutional neural network. In *Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, pp. 26-35.

19 Bettoni, M., Urgese, G., Kobayashi, Y., Macii, E., & Acquaviva, A. (2017). A convolutional neural network fully implemented on fpga for embedded platforms. In *Proceedings of 2017 New Generation of CAS (NGCAS)*, pp. 49-52.

20 Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., ... & Farhan, L. (2021). Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of big Data*, *8*(1), 1-74.

21 Nurvitadhi, E., Sheffield, D., Sim, J., Mishra, A., Venkatesh, G., & Marr, D. (2016). Accelerating binarized neural networks: Comparison of FPGA, CPU, GPU, and ASIC. In *Proceedings of 2016 International Conference on Field-Programmable Technology (FPT)*, pp. 77-84.