Regular Article

An Efficient Hardware Implementation of Convolutional Neural Network in Detect Breast Cancer Histopathology Image

Vo Tan Phat, Hoang Trang

Department of Electronics Engineering, Ho Chi Minh City University of Technology (HCMUT), Ho Chi Minh City Vietnam National University, Ho Chi Minh City, Vietnam

Correspondence: Hoang Trang, hoangtrang@hcmut.edu.vn

Communication: received 17 August 2021, revised 20 September 2021, accepted 21 September 2021
Online publication: 23 October 2021, Digital Object Identifier: 10.21553/rev-jec.280
The associate editor coordinating the review of this article and recommending it for publication was Prof. Tran Manh Ha.

Abstract– **This paper presents our work on evaluating the effectiveness of a novel deep convolutional neural network architecture (CNN) for classifying breast histology images for cancer risk factors as negative or positive. Also, the hardware structure of the proposed model was successfully synthesized and verified. The results indicate that a CNN trained on a small dataset achieved an overall AUC (Area under the receiver operating characteristic curve - ROC Curve) value of 0.922 on a set of 55505 test images. In addition, the time it takes to classify each image is within 3.8 milliseconds instead of a task that even trained pathologists take hours to complete.**

Keywords– **Convolutional Neural Network, FPGA, Breast Cancer, Hardware Implementation.**

1 Introduction

Breast cancer is the most common form of cancer in women, and invasive ductal carcinoma (IDC) is the most common form of breast cancer, according to the American Cancer Society. The peak mortality rate due to breast cancer is due to a lack of awareness about the importance of detecting symptoms early, as well as a lack of training in how to identify breast cancer symptoms [1]. On the other hand, probability of saving a woman with breast cancer depends primarily on whether the patient receives a diagnosis of the disease at an early stage and immediately begins treatment [2–4]. Hence, the accurate identification and classification of breast cancers is an important clinical task, where automated methods can be used to save time and minimize errors.

One of automated methods to early detect breast cancer is use of machine learning techniques such as Artificial Neural Network, K-Nearest Neighbours, and Decision Tree, other studies [5–8] conducted. In [9–19], Convolutional Neural Networks (CNNs) - a subfield of machine learning has also been used in classifying histopathological images as benign or malignant.

However, in the methods of the research papers mentioned above, the choice of hardware implementation for CNN architectures has received little attention, although the benefits of implementing CNN network hardware were demonstrated [20–22] with inference phase due to its parallel architecture and high performance per unit power. Besides, hardware has limitations due to the limited amount of hardware resources. Therefore, it is important to find an efficient hardware architecture of the CNN accelerator to make it available for more computations.

This paper proposes a model with the possibility of deploying CNN network hardware and overcomes the problem of hardware resource limitation. The architecture of the model was built with the proper size to meet the ability to deploy in hardware, and it would be flexible enough for changing some layers or for retraining purposes when there is a change in data input. At the same time, the proposed model achieved an overall AUC (Area under the ROC Curve) value of 0.922 for detecting IDC on a set of 55505 test images.

The rest of this paper is organized as follows. Section 2 provides a brief description of the conventional model and databases. This section also proposes deep CNN architecture employed in this research. Section 3 presents the results obtained through experiments using the proposed methods. Section 4 is for discussion followed by conclusions in Section 5.

2 Methodology

2.1 Conventional Model, Database

For this paper, a publicly available database was studied to evaluate the effectiveness of the proposed model architectures. Originally, the dataset was provided by Janowczyk, Madabhushi [23], and Roa et al. [24], which are later widely shared by the Kaggle community. It consists of 162 sample images that were scanned at 40 times magnification. From there, 277524 images of 50×50 dimensions are extracted (198738 negative, 78786 positive). Each image after this extraction is named in the format uxXyYclassC.png. For example, 10254_idx5_x1801_y1601_class1.png defined as u: patient ID (10254_idx5); X: the x-coordinate where the extracted image is separated from the original

Table I
Distribution of the Train, Validation and Test Datasets

Dataset	Label Negative	Label Positive
Train	126 986	50 630
Validation	31 819	12 584
Test	39 933	15 572

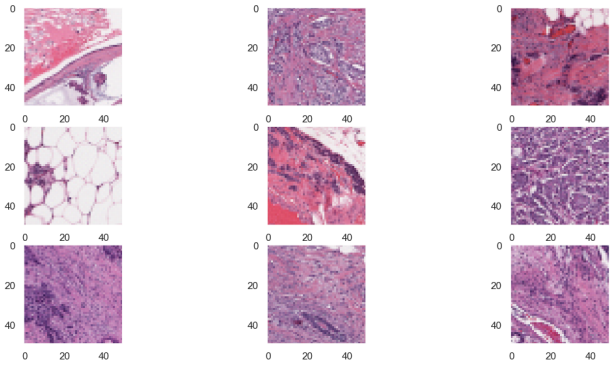

Figure 1. Histopathology images marked as Positive.

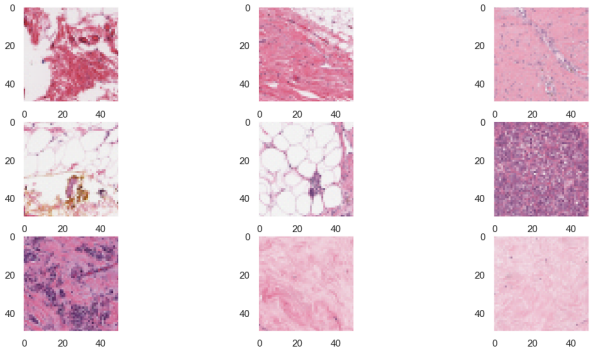

Figure 2. Histopathology images marked as Negative.

image; Y: the y-coordinate where the extracted image is cropped from the original image; C: label class with a value of 0 indicates a negative case, 1 indicates a positive case.

For this research, the dataset was split into groups with the percentages of 64%, 16% and 20% respectively for training, validation and testing purposes. Table I presents the distribution of the number of samples included in each of the training, validation and testing datasets utilized in this research.

Figures 1 and 2 show the symbolic patches of positive and negative images present in the dataset, the x and y axes representing the pixel coordinates of the images. In these two figures, it can be observed that some of them are very small, whereas some have very large differences between the cancer image and the non-cancer image, thus making it challenging for its classification.

In an attempt to address this challenge, there were 2 different network architectures tested, both of these were built in Tensorflow [25]. The main difference between the two architectures was the size of the kernel

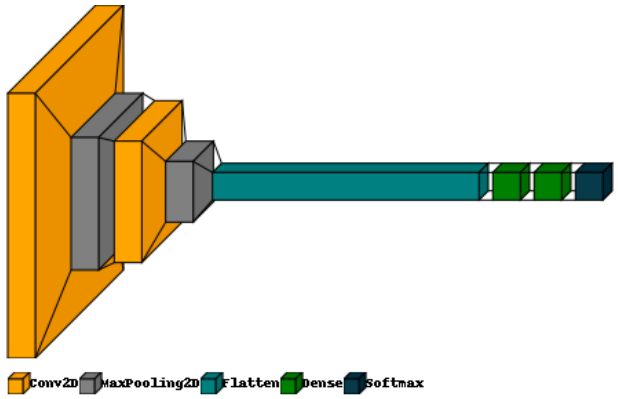

Figure 3. Proposed CNN architecture for predicting breast cancer.

filter. While the filter with a small size (3×3) will have a smaller receptive field which means it will look at very few pixels at once whereas a large kernel (5×5) will look at a larger field view. This in turn mean the features extracted by a small kernel would be highly local whereas the features extracted from the large kernel would be generic and spread across the image.

Both of these network architectures were designed with the effort of minimizing parameters in order to meet the memory strict to be capable of implementing hardware but still retaining the same level of accuracy that existed [13, 26]. All hyperparameters, such as pooling window size, padding, stride, or the size of the convolutional layer, were found experimentally.

2.2 Our Proposed Model

The proposed CNN architecture has the following components: convolutional layer, activation using Rectified Linear Unit (ReLU) layer, local response normalization layer, pooling layer, fully connected layer, and a softmax layer. The arrangement of the architecture is shown visually in Figure 3. From there, it can be seen that the proposed architecture has 2 convolutional layers which were then followed by a fully connected and a softmax layer for classification.

The convolutional layer is a set of filters (also called convolutional kernels) that are convolved with the input data to create an output feature map [27]. In the first proposed architecture, each convolutional layer contains 5×5 convolution filters along with a ReLU, a local response normalization layer, and a maximum pooling layer. The use of 5×5 filters in the first architecture was aimed at extracting generic and spread features across the images of this data. In addition, the convolutional layers have the same padding instead of valid padding meaning the convoluted output retains the same size as the input. The number of convolution filters in each layer is shown in Table II. The initializer values of convolution filters followed a normal distribution with specified mean = 0 and standard deviation = 0.05, except that values whose magnitude was more than 2 standard deviations from the mean were dropped and re-picked. Outputs of the convolution operations are fed into a ReLU layer.

Table II
Number of Convolution Filters in Each Convolutional Layer

Convolutional Layer	Number of Filters
Layer 1	8
Layer 2	16

Table III
Summary Table for the Parameters of the First Architecture in Our Work

Layer Name	Tensor Size	Weights	Biases	Parameters
Input image	$50 \times 50 \times 3$	0	0	0
Conv-1	$46 \times 46 \times 8$	600	8	608
MaxPool-1	$23 \times 23 \times 8$	0	0	0
Conv-2	$19 \times 19 \times 6$	3200	16	3216
MaxPool-2	$9 \times 9 \times 16$	0	0	0
Flatten	1296	0	0	0
FC-1	20	25 920	20	25 940
FC-2	2	40	2	42

Table IV
Summary Table for the Parameters of the Second Architecture in Our Work

Layer Name	Tensor Size	Weights	Biases	Parameters
Input image	$50 \times 50 \times 3$	0	0	0
Conv-1	$48 \times 48 \times 8$	216	8	224
MaxPool-1	$24 \times 24 \times 8$	0	0	0
Conv-2	$22 \times 22 \times 16$	1152	16	1168
MaxPool-2	$9 \times 9 \times 16$	0	0	0
Flatten	1936	0	0	0
FC-1	20	38 720	20	38 740
FC-2	2	40	2	42

The activated values were normalized by the local response normalization layer. This form of normalization makes the neurons that most strongly activate inhibit the activity of its neighbors. This lateral inhibition encourages local contrast enhancement, pushing them apart and forcing them to explore a wider range of features, ultimately improving generalization [28].

The input feature map will then be effectively sampled by the pooling operation. Such a downsampling procedure is useful to obtain a compact, invariant feature representation for moderate changes in object scale, pose, and translation in an image [29]. Thus, the number of parameters and the computational complexity of the model are reduced. In this paper, a max-pooling layer of size 2×2 with a stride of 2 was used.

Fully connected layers connect all neurons in one layer to all neurons in another layer. It works similarly to a traditional multi-layer perceptron (MLP) neural network. To classify the images, the flattened matrix from the output of the convolutional layers passes through a fully connected layer. The fully connected layer was used in this paper has 20 neuron units.

Then a softmax layer with 2 neuron units was connected to it. Essentially, the softmax layer is a densely connected layer with the output values added to a softmax function. The softmax function normalizes the output values into a probability distribution.

The total number of parameters in the first structure is the sum of all parameters in the 2 Convolutional Layers and 2 Fully Connected Layers, achieved 29806 parameters. Table III provides a summary.

The second architecture kept the same number of convolution filters in each convolutional layer, used exclusively 3×3 convolutional filters aimed at extracting highly local features, multiple 3×3 convolution filters were stacked on top of each other before fed into ReLU, normalized, and performing max-pooling. It was achieved 40174 parameters. Table IV provides a summary.

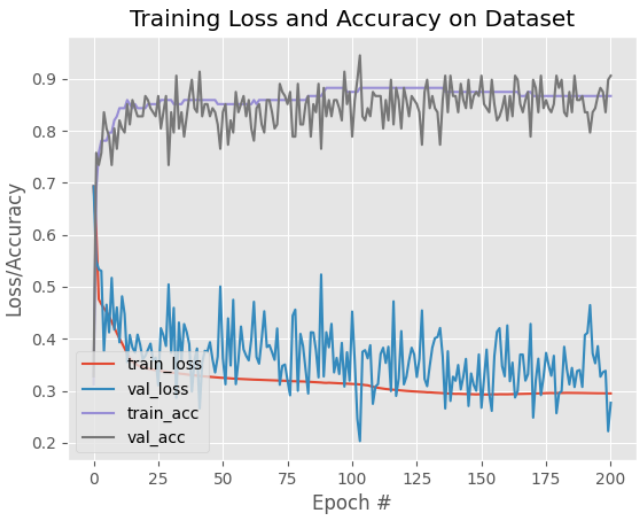

Figure 4. Our first proposed classification model traning plot using Adam.

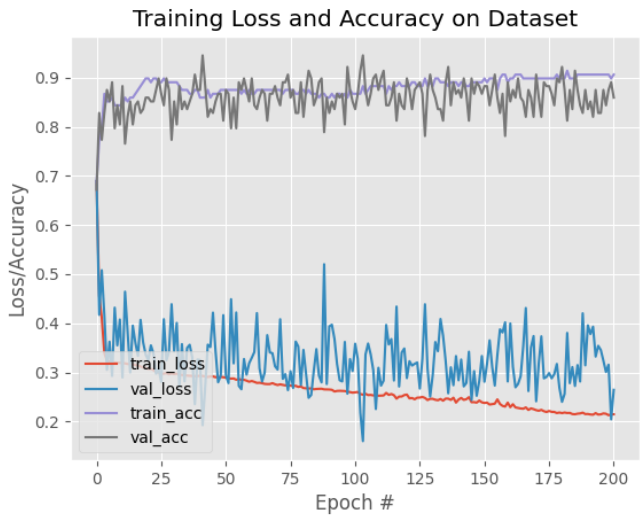

Figure 5. Our first proposed classification model traning plot using Adadelta.

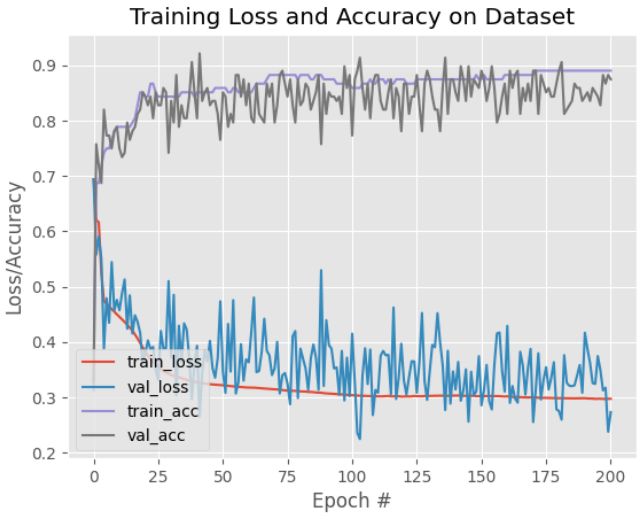

Figure 6. Our second proposed classification model traning plot using Adam.

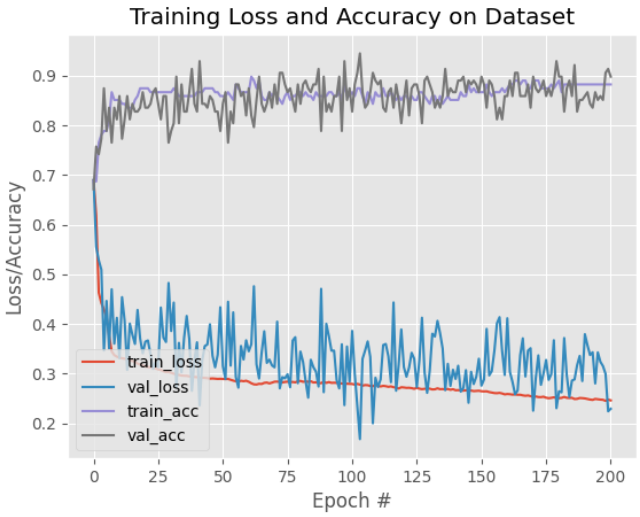

Figure 7. Our second proposed classification model traning plot using Adadelta.

3 Experimental Results

3.1 Scenario

In this paper, the network was trained for 200 epochs and the frequency of validation per 1388 iterations, with two different optimization algorithms, Adam [30] - Adaptive learning rate optimization algorithm and Adadelta [31] - An adaptive learning rate method. Also, both of these optimizations combined with a technique called schedule learning rate with an initial 0.0001 for Adam and 1 for Adadelta, decay every 1000 steps with a 0.9 rate. This training process of two proposed architectures is shown in the following Figure 4, Figure 5, Figure 6, and Figure 7 respectively.

3.2 Results

In these Figures, it can be indicated that the network consistently increases the training and validation accuracy to their optimum values.

The consistency of the validation is often more unpredictable than the smooth progression of the consistency of the training. Such behavior can be attributed to the challenging as well as the practicality level of the validation dataset used, leading to higher variances in training. However, over 200 epochs, the overall fluctuation of the validation process still tends to follow the training of the model, which proves that the model was not overfitting or underfitting.

4 Discussion

While accuracy is an important issue, what is as, or sometimes more important is the number of false negatives and false positives, or in this instance, breast cancer image patches classified as benign and vice versa. Tables V and VI show the confusion matrices of the first and second proposed network architecture in this paper, respectively. Also, AUC-based performance for breast cancer detection was also presented. By using the

Table V
Confusion Matrix of the First Model in Our Work

Optimizations	Testing set	Predicted: benign	Predicted: malignant
Adam	Actual: benign	True Neg.: 36553	False Pos.: 3380
	Actual: malignant	False Neg.: 4481	True Pos.: 11091
Ađaelta	Actual: benign	True Neg.: 36820	False Pos.: 3113
	Actual: malignant	False Neg.: 4438	True Pos.: 11134

Table VI
Confusion Matrix of the Second Model in Our Work

Optimizations	Testing set	Predicted: benign	Predicted: malignant
Adam	Actual: benign	True Neg.: 36415	False Pos.: 3518
	Actual: malignant	False Neg.: 4289	True Pos.: 11283
Ađaelta	Actual: benign	True Neg.: 36737	False Pos.: 3196
	Actual: malignant	False Neg.: 4143	True Pos.: 11429

Table VII
Summary Table for AUC

Methodology		AUC
Existing Benchmark [13]		0.935
Existing Benchmark [26]		0.902
First architecture	Adam	0.911
	Adadelta	0.920
Second architecture	Adam	0.914
	Adadelta	0.922

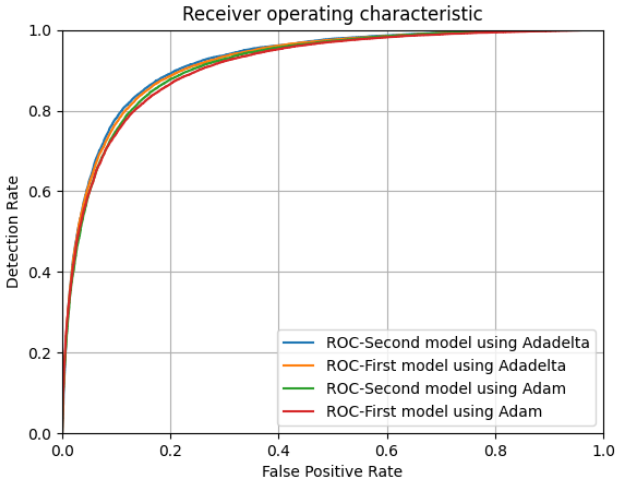

Figure 8. ROC Curves for all approaches.

Test Data Set as mentioned in Table I, the results were determined. Figure 8 presents the Receiver Operating Characteristic (ROC) curve for all approaches. Table VII summarizes the AUC values obtained using these approaches.

In the task of determining whether a patient has breast cancer or not, breast cancer image patches classified as benign (false negative) should be far more concerned than benign image patches classified as malignant (false positive). A false negative would mean not warning about breast cancer when in fact it is the case of cancer which leads to health problems due to no precaution. A false positive means cases would take precautions even if they don't really need to. Therefore, through the comparison of the confusion matrices, the second proposed model and the Adadelta optimization algorithm emerged as the most suitable choice.

As can be seen in Figure 8 and Table VII, there is a noticeable difference in performance when Adadelta optimization was used instead of Adam optimization technique. The effect was outperformed roughly 1% based on AUC results. Also, the AUC value obtained from the chosen model was 0.922 which significantly outperforms the existing benchmark in [26] and almost identical to the result obtained in [13].

However, the model in [13] has up to 458498 parameters, which means 11.4 times larger than the 40174 parameters of the model proposed in this paper. It is the huge number of parameters that is one of the main reasons leading to the hardware implementation barrier for deep CNN network models.

As mention earlier, the proposed model in this paper was built with the proper size to meet the ability to deploy in hardware, and it would be flexible enough for changing some layers or for retraining purposes when there is a change in data input. The proposed convolutional neuron network hardware architecture was created using a high-level synthesis (HLS) tool named LeFlow [32]. This tool uses Google's XLA compiler which emits LLVM [33] code directly from a Tensorflow specification. This LLVM code can then be used with a high-level synthesis tool such as LegUp [34] to automatically generate Verilog code, then subsequently mapped using Quartus to an Altera Cyclone II FPGA to estimate the resource usage. The number of resources used was statistically and summarized in Table VIII.

Table VIII
Proposed Convolutional Neural Network Performance

Resource	Altera - Cyclone II
Combination Logic (elements)	34 562
Register	22 677
Other resource	None

The hardware structure was also verified in real-time condition using Modelsim to simulate in 50 MHz frequency maximum to strongly demonstrate the design potential. From this point, if the proposed model in this paper is developed to its fullest ability, it may potentially detect cancer risk factors - detection hardware or process that can operate parallel to the CPU.

5 Conclusions and Future Works

In this work, a novel model of convolutional neural network was used to classify breast histology images for cancer risk factors as negative or positive. An overall AUC (Area under the ROC Curve) value of 0.922 was achieved on a set of 55505 images (of which 15572 images belong to IDC). According to the results, CNN was proven to detect breast cancer effectively. Also, the hardware structure of the proposed model was successfully synthesized and verified.

A lot of future research would be available. The highest priority would be obtaining a much larger data set as well as more consistency and working with the network parameters to maximize precision and eliminate false negatives. For some other enhancements, there are a variety of embedding strategies that can be attempted and it is worth looking deeper into the max-pooling process.

Acknowledgment

We acknowledge the support of time and facilities from Ho Chi Minh City University of Technology (HCMUT), VNU-HCM for this study.

References

[1] T. Swathi, S. Krishna, and M. V. Ramesh, "A survey on breast cancer diagnosis methods and modalities," in *Proceedings of the International Conference on Wireless Communications Signal Processing and Networking (WiSPNET)*. IEEE, 2019, pp. 287–292.

[2] C. E. DeSantis, F. Bray, J. Ferlay, J. Lortet-Tieulent, B. O. Anderson, and A. Jemal, "International variation in female breast cancer incidence and mortality rates," *Cancer Epidemiology and Prevention Biomarkers*, vol. 24, no. 10, pp. 1495–1506, 2015.

[3] O. Ginsburg, C.-H. Yip, A. Brooks, A. Cabanes, M. Caleffi, J. A. Dunstan Yataco, B. Gyawali, V. McCormack, M. McLaughlin de Anderson, R. Mehrotra *et al.*, "Breast cancer early detection: A phased approach to implementation," *Cancer*, vol. 126, pp. 2379–2393, 2020.

[4] A. F. Rositch, K. Unger-Saldaña, R. J. DeBoer, A. Ng'ang'a, and B. J. Weiner, "The role of dissemination and implementation science in global breast cancer control programs: frameworks, methods, and examples," *Cancer*, vol. 126, pp. 2394–2404, 2020.

[5] A. Kajala and V. Jain, "Diagnosis of breast cancer using machine learning algorithms-a review," in *Proceedings of the International Conference on Emerging Trends in Communication, Control and Computing (ICONC3)*. IEEE, 2020, pp. 1–5.

[6] S. Laghmati, A. Tmiri, and B. Cherradi, "Machine learning based system for prediction of breast cancer severity," in *Proceedings of the International Conference on Wireless Networks and Mobile Communications (WINCOM)*. IEEE, 2019, pp. 1–5.

[7] A. Gupta, D. Kaushik, M. Garg, and A. Verma, "Machine learning model for breast cancer prediction," in *Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. IEEE, 2020, pp. 472–477.

[8] S. A. El Rahman, A. Al-montasheri, B. Al-hazmi, H. Aldkaan, and M. Al-shehri, "Machine learning model for breast cancer prediction," in *Proceedings of the International Conference on Fourth Industrial Revolution (ICFIR)*. IEEE, 2019, pp. 1–8.

[9] O. V. Singh and P. Choudhary, "A study on convolution neural network for breast cancer detection," in *Proceedings of the Second International Conference on Advanced Computational and Communication Paradigms (ICACCP)*. IEEE, 2019, pp. 1–7.

[10] E. L. Omonigho, M. David, A. Adejo, and S. Aliyu, "Breast cancer: tumor detection in mammogram images using modified alexnet deep convolution neural network," in *Proceedings of the International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*. IEEE, 2020, pp. 1–6.

[11] P. Yamlome, A. D. Akwaboah, A. Marz, and M. Deo, "Convolutional neural network based breast cancer histopathology image classification," in *Proceedings of the 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, 2020, pp. 1144–1147.

[12] A. Roy, "Deep convolutional neural networks for breast cancer detection," in *Proceedings of the IEEE 10th Annual Ubiquitous Computing, Electronics & Mobile Communication Conference (UEMCON)*. IEEE, 2019, pp. 0169–0171.

[13] B. N. Narayanan, V. Krishnaraja, and R. Ali, "Convolutional neural network for classification of histopathology images for breast cancer detection," in *Proceedings of the IEEE National Aerospace and Electronics Conference (NAECON)*. IEEE, 2019, pp. 291–295.

[14] F. Shahidi, S. M. Daud, H. Abas, N. A. Ahmad, and N. Maarop, "Breast cancer classification using deep learning approaches and histopathology image: A comparison study," *IEEE Access*, vol. 8, pp. 187 531–187 552, 2020.

[15] B. Kieffer, M. Babaie, S. Kalra, and H. R. Tizhoosh, "Convolutional neural networks for histopathology image classification: Training vs. using pre-trained networks," in *Proceedings of the Seventh International Conference on Image Processing Theory, Tools and Applications (IPTA)*. IEEE, 2017, pp. 1–6.

[16] S. A. Adeshina, A. P. Adedigba, A. A. Adeniyi, and A. M. Aibinu, "Breast cancer histopathology image classification with deep convolutional neural networks," in *Proceedings of the 14th international conference on electronics computer and computation (ICECCO)*. IEEE, 2018, pp. 206–212.

[17] A. Patil, D. Tamboli, S. Meena, D. Anand, and A. Sethi, "Breast cancer histopathology image classification and localization using multiple instance learning," in *Proceedings of the IEEE International WIE Conference on Electrical and Computer Engineering (WIECON-ECE)*. IEEE, 2019, pp. 1–4.

[18] S. Angara, M. Robinson, and P. Guillén-Rondon, "Convolutional neural networks for breast cancer histopathological image classification," in *Proceedings of the 4th International Conference on Big Data and Information Analytics (BigDIA)*. IEEE, 2018, pp. 1–6.

[19] Z. Xiang, Z. Ting, F. Weiyan, and L. Cong, "Breast cancer diagnosis from histopathological image based on deep learning," in *Proceedings of the Chinese Control And Decision Conference (CCDC)*. IEEE, 2019, pp. 4616–4619.

[20] R. Ding, X. Tian, G. Bai, G. Su, and X. Wu, "Hardware implementation of convolutional neural network for face feature extraction," in *Proceedings of the IEEE 13th Inter-*

national Conference on ASIC (ASICON). IEEE, 2019, pp. 1–4.
[21] F. U. D. Farrukh, T. Xie, C. Zhang, and Z. Wang, "Optimization for efficient hardware implementation of CNN on FPGA," in *Proceedings of the IEEE International Conference on Integrated Circuits, Technologies and Applications (ICTA)*. IEEE, 2018, pp. 88–89.
[22] W. Chen, Y. Wang, C. Yang, and Y. Li, "Hardware acceleration implementation of three-dimensional convolutional neural network on vector digital signal processors," in *Proceedings of the 4th International Conference on Robotics and Automation Sciences (ICRAS)*. IEEE, 2020, pp. 122–129.
[23] A. Janowczyk and A. Madabhushi, "Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases," *Journal of pathology informatics*, vol. 7, 2016.
[24] A. Cruz-Roa, A. Basavanhally, F. González, H. Gilmore, M. Feldman, S. Ganesan, N. Shih, J. Tomaszewski, and A. Madabhushi, "Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks," in *Medical Imaging 2014: Digital Pathology*, vol. 9041. International Society for Optics and Photonics, 2014, p. 904103.
[25] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard *et al.*, "Tensorflow: A system for large-scale machine learning," in *12th {USENIX} symposium on operating systems design and implementation ({OSDI} 16)*, 2016, pp. 265–283.
[26] A. Cruz-Roa, H. Gilmore, A. Basavanhally, M. Feldman, S. Ganesan, N. N. Shih, J. Tomaszewski, F. A. González, and A. Madabhushi, "Accurate and reproducible invasive breast cancer detection in whole-slide images: A deep learning approach for quantifying tumor extent," *Scientific reports*, vol. 7, no. 1, pp. 1–14, 2017.
[27] S. Khan, H. Rahmani, S. A. A. Shah, and M. Bennamoun, "A guide to convolutional neural networks for computer vision," *Synthesis Lectures on Computer Vision*, vol. 8, no. 1, pp. 1–207, 2018.
[28] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019.
[29] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio, "Deep learning (vol. 1, no. 2)," *Cambridge: MIT Press. Retrieved December*, vol. 21, p. 2020, 2016.
[30] K. DP and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of the 3rd International Conference for Learning Representations (ICLR)*, 2015.
[31] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.
[32] D. H. Noronha, B. Salehpour, and S. J. Wilton, "Leflow: Enabling flexible fpga high-level synthesis of tensorflow deep neural networks," in *Proceedings of the FSP Workshop 2018; Fifth International Workshop on FPGAs for Software Programmers*. VDE, 2018, pp. 1–8.
[33] C. Lattner and V. Adve, "LLVM: A compilation framework for lifelong program analysis & transformation," in *Proceedings of the International Symposium on Code Generation and Optimization (CGO 2004)*. IEEE, 2004, pp. 75–86.
[34] A. Canis, J. Choi, M. Aldham, V. Zhang, A. Kammoona, T. Czajkowski, S. D. Brown, and J. H. Anderson, "Legup: An open-source high-level synthesis tool for fpga-based processor/accelerator systems," *ACM Transactions on Embedded Computing Systems (TECS)*, vol. 13, no. 2, pp. 1–27, 2013.

Vo Tan Phat is currently a programming engineer at the Research and Development Center of the Vietnam Posts and Telecommunications Group (VNPT RnD). He received his B.Sc. degree from Ho Chi Minh University of Technology (HCMUT), Vietnam in 2018. His research interests include Artificial intelligence, System-on-Chips/Network-on-Chips, Multimedia applications.

Hoang Trang was born in Nha Trang city, Vietnam. He received the Bachelor of Engineering, and Master of Science degree in Electronics-Telecommunication Engineering from Ho Chi Minh City University of Technology in 2002 and 2004, respectively. He received the Ph.D. degree in Microelectronics-MEMS from CEA-LETI and University Joseph Fourier, France, in 2009. From 2009-2010, he did the post-doctorate research in Orange Lab-France Telecom. Since 2010, he has been a lecturer, promoted to Associate Professor in 2014, at Faculty of Electricals-Electronics Engineering, Ho Chi Minh City University of Technology. His field of research interest is in the domain of ASIC/FPGA implementation-architecture, Speech Recognizer, MEMS, virtual fabrication, security router.