The Utilization of ResNet Model in White Blood Cell Classification

Chenxu Li
Glasgow College
University of Electronic Science and Technology of China
Chengdu, China
Email: 2066523411@qq.com

Haotian Wang
Glasgow College
University of Electronic Science and Technology of China
Chengdu, China
Email: 2932592997@qq.com

Zhihan Xiao
Glasgow College
University of Electronic Science and Technology of China
Chengdu, China
Email: 2278362285@qq.com

Shuyi Jin
Glasgow College
University of Electronic Science and Technology of China
Chengdu, China
Email: 2458144741@qq.com

Teoh Teik Toe
College of Business (Nanyang Business School)
Nanyang Technological University
Singapore
Email: ttteoh@ntu.etu.sg

With the rapid development and innovation in the field of computer science, artificial intelligence and machine learning have become more and more common in the daily life. In medical field, these technologies are used to support image processing and classification, like helping with blood test. Neutral networks will be chosen as the method. In this paper, a classification system founded on ResNet-50 model, which takes CNN as the main structure, is introduced. The system extracts important features of images and analyzes images through layers. A trained model will be built using a dataset and then it can classify other strange samples. The reliability and possible optimization of this classification system is also evaluated, including accuracy and model loss in order to increase its practical value.

Keywords: Blood Cell Classifier; CNN; Python; ResNet-50; Machine Learning

I. INTRODUCTION

A. White blood cell

White blood cell (or leukocyte, WBC) is one of the most significant categories of blood cells in human body, as its normal amount of an adult is calculated by nine orders of magnitude. The exact content will fluctuate within a certain range according to different time of the day and changeable organism functional status. External features of this kind of blood cell include colorless, spherical and nucleated [1]. Unlike other forms of blood cell such as erythrocyte and thrombocyte, shape and character of leukocytes are unstable.

According to appearance, function and originated area, white blood cells can be divided into mainly three categories: granulocyte, monocyte, and lymphocyte. Among these, granulocytes can be further divided into neutrophil, eosinophil and basophil according to the staining properties of granules in the cytoplasm.

The main function of white blood cells is defense.

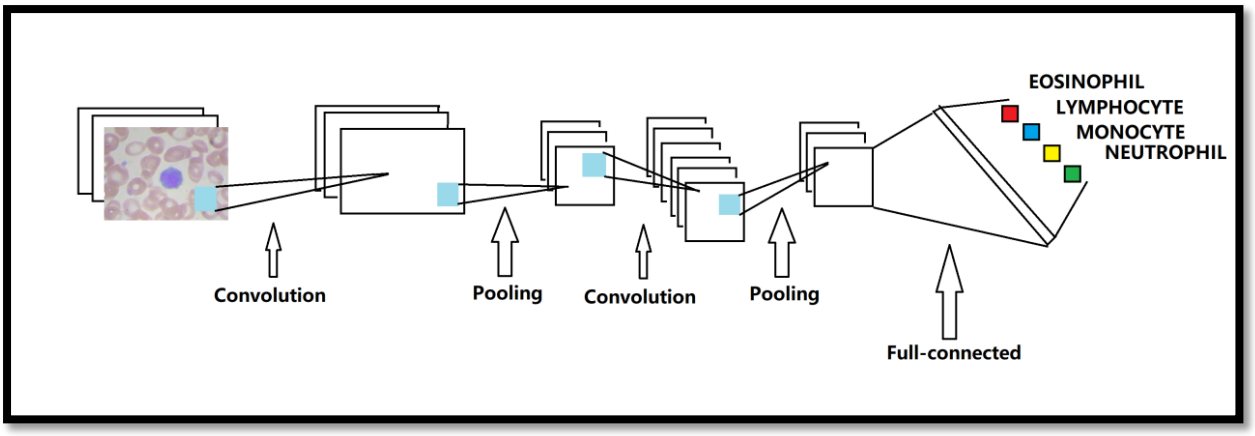

Figure 1: Structure of CNN model

Various types of them participate in the body's defense response in their own ways. In fact, the pathogen purge activities taken by white blood cells often cause external symptoms like fever or sore throat representing upper respiratory tract infection. For medical diagnosis, after observing the symptoms, the doctor may analyze the blood routine test of the patient to distinguish whether the main infection is bacterial or viral, which provides guidance on choosing antibiotics or antiviral agents. If bacterial infection occurs, a statistical increase in neutrophils may be seen. If a viral infection occurs, an obvious increase in lymphocytes (or lymphopenia) is usually seen.

However, the analysis of blood cell morphology largely depends on manual work. Doctors have to exam the blood samples through microscope [2,3], so there exist inevitable errors and labor-intensive status. Therefore, it is necessary to develop an accurate and fast digital image analysis system to classify and count WBCs.

B. CNN

CNN, meaning convolutional neural networks, has almost excellent performance in image analysis nowadays, so this paper will classify leukocytes based on CNN and optimize the classification system. The basic theories about CNN are as follows.

It is a kind of feedforward neural networks with deep structure and convolution computation. It is one of the representative items of deep learning. There are five layers in the structure of CNN, which in detail include input layer, output layer, convolution layer, pooling layer, and full-connected layer [4,5]. In order to improve the performance of the whole network, the input layer needs to preprocess and normalize the input image. The main function of convolution layer is to extract the features of input data. It uses convolution kernel as a tool to complete the function of data extraction [6], which is actually a scanner with a specified window size. It extracts the features in the data by scanning the input data again and again. Then the important features in the image can be recognized after convolution kernel processing. Pooling layer is similar to convolution kernel, and can be regarded as a core feature extraction method of input data in coiler neural network. It not only realizes the compression of the original data, but also greatly reduces the parameters involved in the calculation of the model. The most commonly used pooling layer methods are average pooling layer and maximum pooling layer.

It is worth noting that the pooling layer also needs to define a convolution kernel sliding window similar to

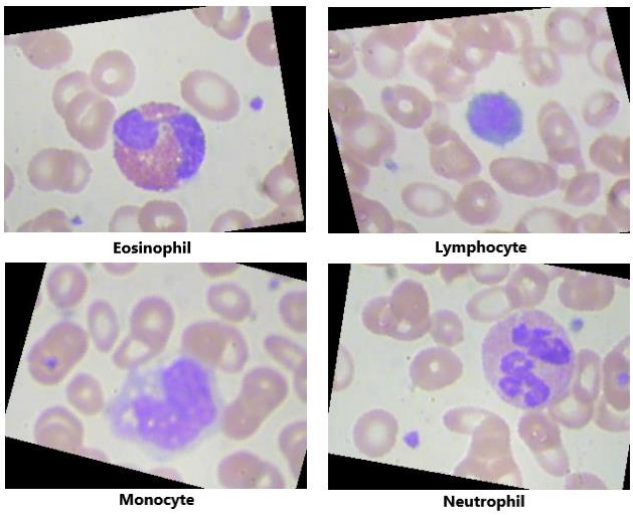

Figure 2: Blood cell figures in the dataset

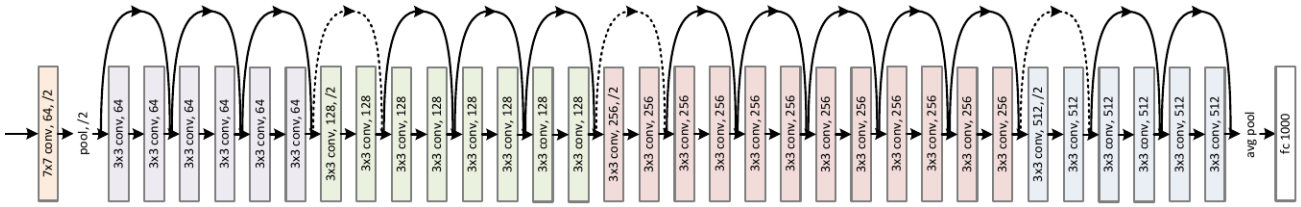

Figure 3a: Structure of ResNet-50 model

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
		3×3 max pool, stride 2				
conv2_x	56×56	[3×3, 64; 3×3, 64] ×2	[3×3, 64; 3×3, 64] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3	[1×1, 64; 3×3, 64; 1×1, 256] ×3
conv3_x	28×28	[3×3, 128; 3×3, 128] ×2	[3×3, 128; 3×3, 128] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×4	[1×1, 128; 3×3, 128; 1×1, 512] ×8
conv4_x	14×14	[3×3, 256; 3×3, 256] ×2	[3×3, 256; 3×3, 256] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×6	[1×1, 256; 3×3, 256; 1×1, 1024] ×23	[1×1, 256; 3×3, 256; 1×1, 1024] ×36
conv5_x	7×7	[3×3, 512; 3×3, 512] ×2	[3×3, 512; 3×3, 512] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3	[1×1, 512; 3×3, 512; 1×1, 2048] ×3
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Figure 3b: Structure of ResNet-50 layer

that in the convolution layer, but this sliding window is only used to extract important features in the feature map. The main function of the full-connected layer is to compress the features extracted from the input image after rolling and pooling operations, and complete the classification function of the model according to the compressed features. Finally, the output layer shows the classification and its probability.

II. METHODOLOGY

A. Dataset

The dataset about white blood cells we use to train the model is from Kaggle. It contains 12515 images, with the images divided into 4 different kinds, eosinophil, neutrophil, lymphocyte and monocyte. Samples of these 4 kinds are respectively shown in Figure 2.

B. Network structure

The main structure used as mentioned before is ResNet-50. ResNet is the abbreviation of residual network, which has two blocks named convolution and identity respectively [5].

1) Model structure

ResNet is a convolutional neural network proposed by 4 scholars from Microsoft Research. The whole structure of ResNet-50 is shown in Figure 3a and 3b.

The ResNet network is referenced to the Visual Geometry Group 19 (VGG19) network, which is modified, with a residual element added through the

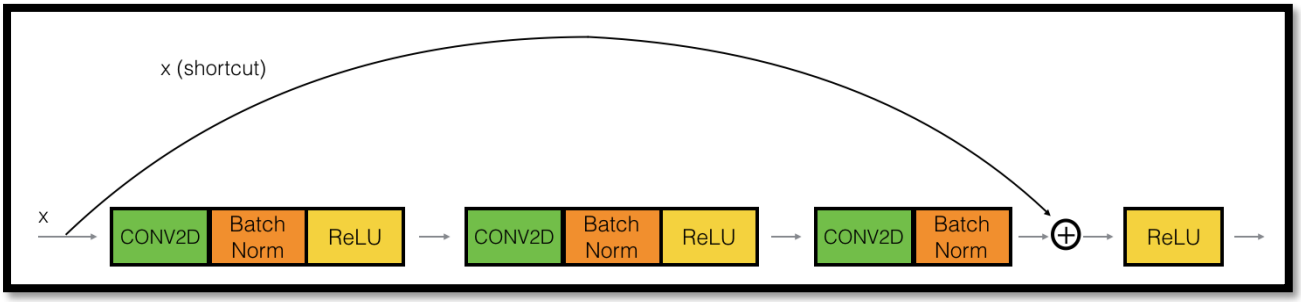

Figure 4a: Structure of conv block

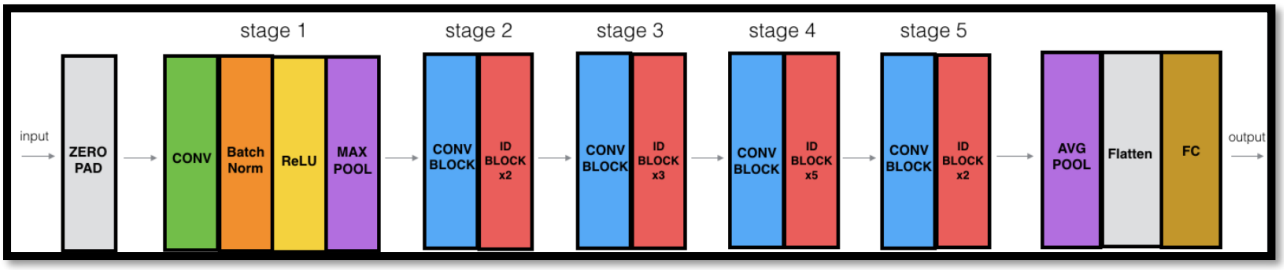

Figure 4b: Structure of identity block

short-circuit mechanism. Instead of learning some function without reference (X), it makes a reference (X) of the input at each level and learns to form a residual function. This residual function is easier to optimize and can greatly deepen the number of network layers. In the residual block in Figure 4a and 4b it has two levels, the following expression,

$$F = W_2\sigma(W_1x) \quad (1)$$

where σ stands for the nonlinear function ReLU. Then take a shortcut and a second ReLU to get the output y.

$$y = F(x,\{W_i\}) + x \quad (2)$$

When you need to change the input and output dimensions, it is easy to do a linear transformation Ws on x at

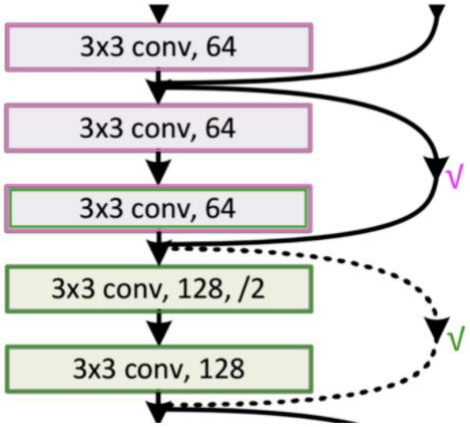

Figure 5: The shortcut connection

shortcut as follows.

$$y = F(x,\{W_i\}) + W_sx \quad (3)$$

The experiment proved that x was enough. Therefore, for layers with the same output feature map size, there are the same number of filters, that is, the same number of channels. When the feature map size is halved (pooling), the number of filters is doubled.

For residual networks, a shortcut connection with a matching dimension is a solid line, and vice versa is a dashed line, according to Figure 5.

The connection part of the solid line (the first pink rectangle and the third pink rectangle) both perform 3x3x64 convolution, and they have the same number of channels, so the calculation method is used:

$$Y = F(x) + x \quad (4)$$

The connection part of the dashed line (the first green rectangle and the third green rectangle) is the convolution operation of 3x3x64 and 3x3x128, respectively. They have different number of channels (64 and 128), so the calculation method is used:

$$Y = F(x) + Wx \quad (5)$$

2) Evaluation

Deep convolutional network naturally integrates features at different levels of low, middle and high, and the levels of features can be enriched by deepening the network levels. Thus, when constructing convolutional networks, the higher the depth of the network, the richer the feature levels that can be extracted. Therefore, it tends to use a deeper network structure in order to obtain higher level features. However, when using deep network structures, it meets two problems, the network degradation problem and the gradient vanishing or explosion problem. ResNet-34 is one of the solutions to these two problems.

Initially, the network degradation problem is explained. ResNet wants to avoid learning the parameters of the identity mapping of this layer, and uses the structure as shown in the above figure, so that

$$h(x) = f(x) + X \quad (6)$$

Here, f (x) is called residual term. It is found that if the

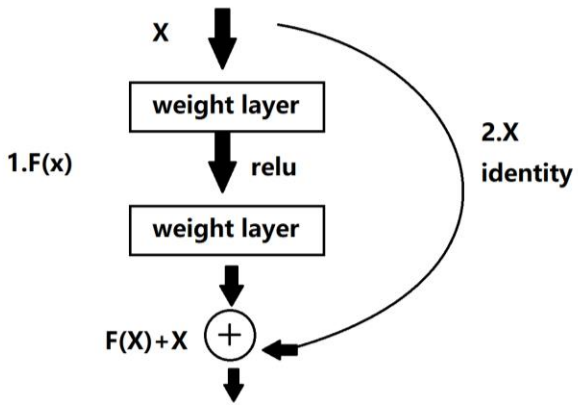

Figure 6: Theory of the equation

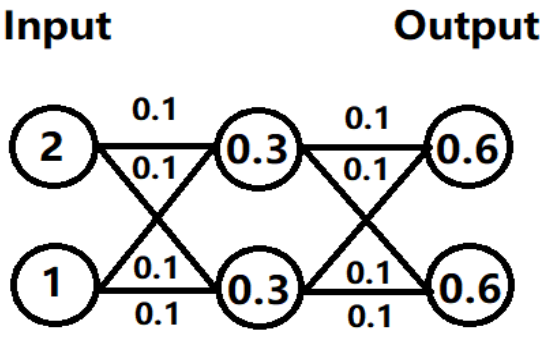

Figure 7: Calculation principle

redundant layer to be identically mapped is wanted, it will need to learn

$$f(x) = 0 \quad (7)$$

Learning Equation (7) is easier than learning

$$H(x) = x \quad (8)$$

because the parameter initialization in each layer of the network is generally biased towards 0, so that when learning Equation (8) compared to updating the parameters of the network layer, the updated parameters of the redundant layer learning Equation (7) can converge faster, as shown in Figure 7.

It is assumed that the Zeng network only undergoes linear transformation and has no bias or activation function. It is found that because the random initialization weight is generally biased toward 0, the output value of the network is [0.6 0.6], which is obviously closer to [0 0], rather than [2 1]. Compared with learning Equation (8), the model is faster to learn Equation (7), and ReLU can activate negative numbers to 0, filter the linear change of negative numbers, and make "f (x) = 0" faster. In this way, after the network determines which layers are redundant layers, the network of this layer is mapped to the input of the previous layer by learning the residual Equation (7), so that the network effect with these redundant layers is the same as that without these redundant layers, which largely solves the problem of network degradation.

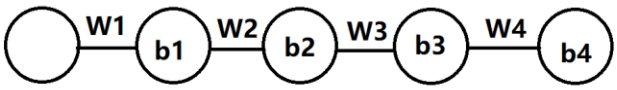

Figure 8: Calculation process of the equation

Then it goes to the problem of vanishing gradient or exploding gradient. It is found that in the deep network layer, the parameters are generally initialized closer to 0, so when updating the parameters of the shallow network in the training process, the gradient will easily disappear with the depth of the network, and the shallow parameters cannot be updated.

$$\frac{\partial C}{\partial b_1} = \sigma(z_1)W_2\sigma(z_2)W_3\sigma(z_3)W_4\sigma(z_4)\frac{\partial C}{\partial a_4} \quad (9)$$

$$z_i = w_i a_{i-1} + b_i \quad (10)$$

$$a_i = \sigma(z_i) \quad (11)$$

Suppose that the parameters b1, W2, W3 and W4 need to be updated now because the random initialization is biased towards 0. Through chain derivation, it found that the multiplication of W1, W2 and W3 will yield a number closer to 0, so the gradient of the obtained b1 will be close to 0, and the gradient will disappear. When ResNet finally updates the parameters of a node, since Equation (6), and since the result of chain derivative is shown, no matter how small the derivative parameter in the right part of the parentheses is, because of the existence of 1 on the left, and the original continuous product in chain derivative becomes a continuous addition state, it can ensure that the node parameter update will not occur gradient disappearance or gradient explosion phenomenon.

III. EXPERIMENT

A. Programming

To begin with, we imported the packages of panda, traverse and list all the images in the dataset for data processing. The number of total images was 12515, indicating that the samples are enough for our research.

Next, we imported the packages which were needed. The network architecture of Bottleneck was predefined by us and was defined its forward propagation. In the block, the convolution kernel of the main branch was changed. Then we initialized and inherited some features of nn. Module. After that, we performed three convolutions and normalized each time. What has to mention is that the output dimension of the third convolution is 4 times that of the previous convolution Following that, we let it spread forward and solved the problem that the number of heights, width and channel number in Equation (6) do not match.

In terms of the classification of Resnet, the initial number of input channels was set to 64. In this category, we utilized the method of make_layer to construct 4 blocks in the ResNet network. The first input was the block which belonged to Bottleneck. The second input were the channels that the block outputted. The third input was the residual vice structure that each block contained. (In this step block is just the same as layer, for example layer1.) Added the two parts and added the nonlinear factors through 'relu', and then spliced

according to the X dimension Downsample was also mainly aimed at handling the dimension mismatch problem, in other words, raising the dimension of the input of the residual structure. In the end, propagate it as the former step. Later we define resnet34, resnet50, resnet101, resnext50_32x4d and resnext101_32x8d for later use.

Following that, we utilized Pytorch to conduct the deep-learning. As usual, we imported the required packages. Next, we set the number of ergodic datasets, epochs to 50 times. The batch_size was set to 16. Then we prepared data sets and preprocessed. The image was randomly cropped all around 0 and resized to 256 * 256, half of the probability of the image was flipped while half of the probability was not flipped, and then the dimension transformation was from 32x32x3 to 3x3x32, normalized the mean and variance used for each layer of R, G and B. We run the training and after training each epoch, test the accuracy and the loss. We optimize the more accurate one into consideration and selection. Finally, we obtained the images of Test_accuracy and Train_loss versus epoch in the figure below. Torch summary was demonstrated in the end.

B. Results and analysis

It takes approximately 1 minute and 40 seconds for each epoch to train the model. The highest training accuracy reaches 86.81% while the average training accuracy is about 82.31%. For machine learning area, this is a satisfying enough accuracy value, but for medical examination, which needs the least errors, obviously the model has better to be modified.

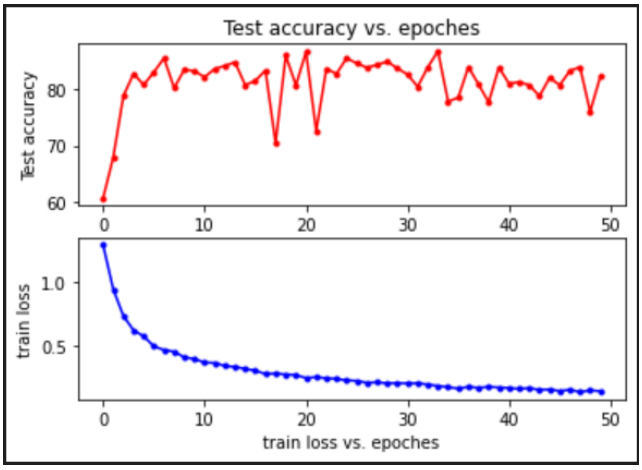

Figure 9: Results versus epoches

Through Figure 9, it can be intuitively seen that after nearly 5 epochs can the test accuracy converge to the average value. Combined with the time used by each epoch, the total training speed might be accepted well. The problem is that the test accuracy changes in a wide range to some extent, which is also the main factor dragging down the average accuracy value. However, those valley values only take extremely low percentage, meaning that the trained model has reliability more than expected.

As for the train loss, its value sharply drops to less than 0.5 after 5 epochs, while reaching a stable status after about 20 epochs. This situation points that the whole process completes learning at an early time, with a time cost similar to that of test accuracy.

IV. CONCLUSION

According to the whole experiment, it might be admitted that the ResNet-50 model has enough ability to complete the classification task. It finally reaches an accuracy up to 86.8% with an average of 82.3%. However, for practical medical blood test, the dataset lacks of a kind of white blood cell which is basophil. This leads to decrease in value of the model.

In conclusion, the ResNet-50 model has satisfying potential and application value, also is worth to be optimized for future medical examination.

ACKNOWLEDGMENT

Chenxu Li, Haotian Wang, Zhihan Xiao and Shuyi Jin thank Professor Teoh Teik Toe, the director of the AI lab in Nanyang Technological University, for his patient assistance in our experiment. Lectures and directions from our professor really help a lot and bring us to further area of science. He plays a vital role in our learning process and article writing.

REFERENCES

[1] K. T. Navya, K. Prasad and B. M. K. Singh, “Classification of blood cells into white blood cells and red blood cells from blood smear images using machine learning techniques,” in *2021 2nd Global Conference for Advancement in Technology (GCAT)*, 2021, pp. 1-4, doi: 10.1109/GCAT52182.2021.9587524.

[2] F. I. Sholeh, “White blood cell segmentation for fresh blood smear images,” in *2013 International Conference on Advanced Computer Science and Information Systems (ICACSIS)*, 2013, pp. 425-429, doi: 10.1109/ICACSIS.2013.6761613.

[3] S. Khobragade, D. D. Mor and C. Y. Patil, “Detection of leukemia in microscopic white blood cell images,” in *2015 International Conference on Information Processing (ICIP)*, 2015, pp. 435-440, doi: 10.1109/INFOP.2015.7489422.

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

[5] N. Zakaria, F. Mohamed, R. Abdelghani and K. Sundaraj, “Three ResNet Deep Learning Architectures Applied in Pulmonary Pathologies Classification,” in *2021 International Conference on Artificial Intelligence for Cyber Security Systems and Privacy (AI-CSP)*, 2021, pp. 1-8, doi: 10.1109/AI-CSP52968.2021.9671211.

[6] M. -Y. Lee, J. -H. Lee, J. -K. Kim, B. -J. Kim and J. -Y. Kim, “The Sparsity and Activation Analysis of Compressed CNN Networks in a HW CNN Accelerator Model,” in *2019 International SoC Design Conference (ISOCC)*, 2019, pp. 255-256, doi: 10.1109/ISOCC47750.2019.9027643.