

PAPER • OPEN ACCESS

Applying the Deep Learning Model on an IoT Board for Breast Cancer Detection based on Histopathological Images

To cite this article: Shahirah Zahir *et al* 2021 *J. Phys.: Conf. Ser.* **1755** 012026

View the [article online](#) for updates and enhancements.

You may also like

- [Architectures Based on Deep Learning for the Detection of Invasive Ductal Carcinoma](#)
Isha Gupta, Sheifali Gupta and Swati Singh
- [Towards label-efficient automatic diagnosis and analysis: a comprehensive survey of advanced deep learning-based weakly-supervised, semi-supervised and self-supervised techniques in histopathological image analysis](#)
Linhao Qu, Siyu Liu, Xiaoyu Liu *et al.*
- [Establishing a survival prediction model for esophageal squamous cell carcinoma based on CT and histopathological images](#)
Jinlong Wang, Lei-Lei Wu, Yunzhe Zhang *et al.*



PRIME
PACIFIC RIM MEETING
ON ELECTROCHEMICAL
AND SOLID STATE SCIENCE

HONOLULU, HI
Oct 6–11, 2024

Abstract submission deadline:
April 12, 2024

Learn more and submit!



Joint Meeting of

The Electrochemical Society
•
The Electrochemical Society of Japan
•
Korea Electrochemical Society

Applying the Deep Learning Model on an IoT Board for Breast Cancer Detection based on Histopathological Images

Shahirah Zahir¹, Amiza Amir^{1,2}, Nik Adilah Hanin Zahri^{1,2} and Wei Chern Ang³

¹ School of Computer & Communication Engineering, Universiti Malaysia Perlis (UniMAP), Malaysia

² Advanced Communication Engineering (ACE) Centre of Excellence, UniMAP, Malaysia

³ Clinical Research Centre, Ministry of Health Malaysia, Hospital Tuanku Fauziah, Perlis, Malaysia

E-mail address: amizaamir@unimap.edu.my

Abstract. In breast cancer diagnosis, pathologists evaluate microscopic images of tissue samples to determine if it is benign or malignant. The manual examination process could result in delayed diagnosis, which leads to late cancer treatment and can risk lives. In this paper, we proposed an automated, low-cost, and portable breast cancer detection based on histopathological images by using deep learning. The deep learning models were designed by using the Convolutional Neural Network (CNN). This paper compares the performance of the CNN model by using transfer learning utilizing a pre-trained model (VGG16) and the performance of a CNN model without transfer learning. The result shows that transfer learning provides a good base for classification of histopathological images. The model was successfully deployed on a Raspberry Pi, which demonstrates the model efficiency to run on a lightweight and portable processor.

1. Introduction

Breast cancer is the second most common cancer to occur towards women worldwide, and it is the most common cancer that been detected among Malaysian women. Some studies show that breast cancer developed between 20 years old and 40 years old. A woman tends to get breast cancer when their age is increasing. Nearly 3,500 breast cancer cases are detected every year. Breast cancer is a significant cancer among women in Malaysia and followed by cervical cancer [1].

Traditionally, breast cancer usually detected by pathologists manually examining in the lab by using a microscope or x-ray images to analyze whether the breast tissue has cancer or not. The disadvantage of this process is the time taken for the pathologist to examine the tissues, which will take a long time to know the results. This process is time-consuming. Fast detection is essential and needed so that action can be considered earlier to cure and prevent cancer from becoming worst.

There are numerous technologies available that can enhance breast cancer detection, and the utilization of artificial intelligence approaches in medical fields can be considered as excellent assistance in the decision-making process of medical practitioners. Most of the research that has been done in breast cancer detection is using machine learning method to detect the existence of cancer from the histopathological (microscopic) images [2][3][4][5][6].



Deep learning demonstrated a high accuracy rate [6]. Deep learning can be considered as a sophisticated version of neural networks with a higher number of layers that result in more complex computation. The accuracy and efficiency of the algorithm depend on the deep learning network design, which is specific to the application. Hence, the study to improve the deep learning structure for cancer detection based on microscopic images is desirable.

Also, the equipment for breast cancer detection is usually costly. It is expensive to buy or do the maintenance of the machine. Patients need to wait for their turn to get the diagnosis results since the resources (machine, money, and staff) are limited. Low cost and portable devices are useful to speed up the breast cancer diagnosis and to allow scalable examination.

The whole new impact can be brought to the environment of breast cancer detection by using deep learning on a low cost, small, and embedded device. Similar work has been conducted for skin cancer detection [7]. Hence, our research mainly focuses on the development of a deep learning model to detect breast cancer based on histopathological images on an IoT board so that it is low cost, portable, and easy to access.

The system examines the microscopic images of breast tissues and decides whether the breast tissue has cancer. Two deep learning design approaches have been used in this research. The first approach is the Convolutional Neural Network (CNN) model, which is developed from scratch (without a pre-trained model). The second approach is by using transfer learning that combines the convolutional layers of a pre-trained VGG16 model with the fully connected layers, which is fine-tuned for breast cancer pathological images. This paper compares the performance of breast cancer detection models from these two approaches. Then, the model was successfully applied on an IoT board (a Raspberry Pi) to demonstrate its low-cost and portability.

The second section of this paper will discuss the related works, followed by methodology. Next, we present the results, which include the performance of the breast cancer detection models and demonstrate the application of the model on a Raspberry Pi. Finally, we summarize our works in this paper in conclusion

2. Related Works

Breast cancer auto-detection and classification from breast histopathology images were developed in [2] from the DRYAD database. First, the contrast enhancement is used to remove the stain because the histological image is not clear, and the noise needs to be reduced by using the Gaussian blur filter so that the contrast can be improved, and noise can be reduced. Next, segmentation is applied by using the K-mean clustering algorithm, and then, watershed segmentation and color thresholding were performed. Shape and morphology features were extracted during feature extraction. In this study, a simple rule-based classifier and decision tree J48 were used for classification.

In [3], the histopathological images were taken from Mansoura University Hospital with 72 microscopic image samples of each benign and malignant. Then, the images are resized by using any of two sizes: (1024 x 1024) or (512 x 512). The noise in the images was removed by using a median filter. Next, the purposed of unsharp masking was applied to make the quality of the color image better. K-means, C-means Clustering and Watershed were used during the segmentation. The pre-processed images then underwent a feature extraction process that includes shape, texture, and color descriptors. Three classifiers that were used in this research are Support Vector Machine (SVM), K-Nearest Neighbours (K-NN), and Back-Propagation Neural Network (BPNNs).

In [4], developed a system for diagnosis, prognosis and prediction of breast cancer by using Artificial Neural Network (ANN) and Learning Vector Quantization (LVQ). A total of 683 microscopic images is taken from the Wisconsin breast cancer diagnosis (WBCD) database. The amount of 444 images is used for training that contains 260 benign and 184 malignant meanwhile, 239 images are used

for testing that includes 184 benign and 55 malignant for each neural network model. The LVQ model consists of 1 hidden layer and 15 neurons in the hidden layer.

In [5], proposed a technique of a parallel neural network model to detect breast cancer. The data set that been used in this research is from the University of Wisconsin hospital. The data set consists of 699 images where 458 images are benign, and 241 is malignant. The feed-forward neural network model and backpropagation learning algorithm with momentum and the variable learning rate are trained. In this paper, single and multilayer networks are trained. The output results show that a multilayer neural network is better than the single layer in terms of time taken that need to be trained.

Based on [6], new hybrid convolutional and recurrent deep neural network for breast cancer histopathological image classification is proposed. The initial pathological data set images of 249 have been taken from Bioimaging 2015 data set, and 3771 extended images have been collected by collaboration with Peking University International Hospital. The total pathological images used is 4020.

In [7], proposed an implementation of a low-cost embedded device, Raspberry Pi, that classifies the dermatologist-level accuracy skin lesion images and works in a standalone without network connectivity. This application motivates the potential of developing such a low-cost device for automated detection of breast cancer based on microscopic images.

3. Methodology

This project involved several stages, which include image pre-processing, design, and implementation of deep neural network models to select the best possible hyperparameters and deployment of the model on hardware.

3.1. Model Development

The breast cancer histopathological images dataset will be taken from the Breast Cancer Histopathological Image Classification (BreakHis) database. After the image pre-processing completed, 70% of the images were split for training, 15% for validation, and 15% for testing. All the pre-processed images will be used as data input for the training Convolutional Neural Network (CNN).

By using transfer learning, we take a pre-trained model VGG16 and re-purpose it for solving the breast cancer detection problems. The fully connected layers of the pre-trained model were removed, and the remaining convolutional layers are used as feature extractors. The new fully connected networks are then trained with the samples of microscopic images to fine-tune the model in solving breast cancer detection. After the complete CNN model (convolutional layers with fully connected networks) has been trained, the images from the testing set will be applied to the trained model to classify the output. The final output determines whether the breast cancer histopathological images are cancerous (Malignant) or non-cancerous (Benign).

3.2. Breast Cancer Histopathological Database (BreakHis)

A total of 7,909 breast cancer histopathological images is taken from the BreakHis database, as shown in figure 1.

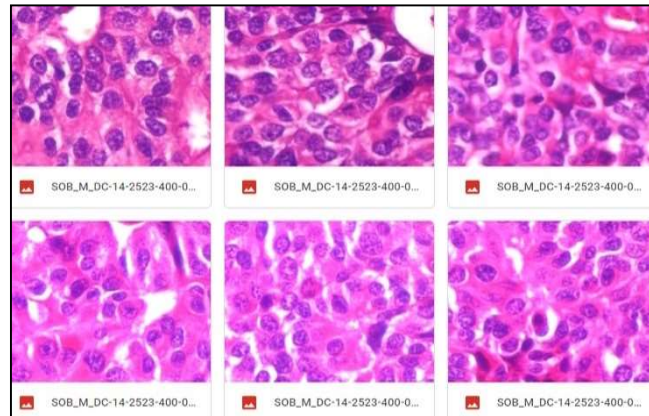


Figure 1. Example of breast cancer histopathological images from the BreakHis database

The BreakHis database [8] was built in collaboration with the P&D Laboratory – Pathological Anatomy and Cytopathology, Parana, Brazil. It consists of four different magnifying factors, which are 40X, 100X, 200X, and 400X. The total samples of 5429 cancerous (Malignant) and 2480 samples of non-cancerous (Benign) are provided, as described in Table-I. All the images will be split into three sets of data: training, validation, and testing.

Table 1. Dataset Distribution

Magnification Factors	Benign	Malignant	Total
40X	652	1370	1995
100X	644	1437	2081
200X	623	1390	2013
400X	588	1232	1820
Total of Images	2480	5429	7909

The images are in PNG format, RGB, 700X460 pixels, and 8-bit depth in each channel. Benign and Malignant tumors can be divided into several different types. There are four types of benign tumor (non-cancerous): adenosis (A), fibroadenoma (F), phyllodes tumor (PT), and tubular adenoma (TA). While, there are four types of a malignant tumor (cancerous) which are carcinoma (DC), lobular carcinoma (LC), mucinous carcinoma (MC) and papillary carcinoma (PC).

3.3. Image Pre-processing

Image pre-processing is a process to improve the input image by removing unwanted distortions or enhances the image features for further process. Since the microscopic images have very detailed and complicated pattern structured, image pre-processing is needed to overcome this problem. The first technique that will be used is resizing the images. The purpose of this technique is to reduce the size of pixels and to make the process become faster. The image pre-processing was done by using the OpenCV (Open Source Computer Vision) library, which is one of the open-source machine learning software libraries where it specifically for computer vision problems.

The images consist of different magnifying factors (40X, 100X, 200X, and 400X). All of the images are initially in size of 700X460 pixels. Then, the images will be resized to 200X200 pixels to make sure that all the images have the same dimensions of square and in the same pixels. The second techniques that will be used is the conversion of RGB to grayscale images. This technique is applied because, in the colour images, it makes more difficult to identify the images features such as edges

compared to grayscale images. The format of images is originally kept as PNG (Portable Network Graphic). After all the pre-processing steps are done, the images will be saved at the Google Drive. Then, the images will be classified into training, validation, and testing sets for the CNN model used.

3.4. Convolutional neural network (CNN) Development

The model is developed by using Python. Python is useful for developing machine learning and deep learning because it has extensive libraries such as Keras, TensorFlow, NumPy, and many more. Convolutional Neural Network (CNN) is a deep artificial neural network that specially used to recognized and classify images meanwhile, the other neural network is typically used in other applications such as voice recognition, pattern recognition, and many more. This breast cancer detection will detect whether it is cancer or non-cancer. Therefore, there will be only two classes for the output, namely Benign and Malignant. This means that binary classification is used, "0" or "1". 1 stand for Malignant, and 0 is for Benign.

The four basic operations that are needed to build breast cancer detection, neural network models:

- Convolutional Process
- ReLU Operation
- Max Pooling Process
- Classification (Fully Connected Layer)

The convolutional process involves three elements: input matrix, feature descriptor, and feature maps. An input matrix in this research is a standard image consisting of three channels where the channels are RGB (red, green, and blue). In this preliminary research, grayscale images are used. Grayscale images consist of only just one channel. So, a single 2D matrix will be representing an image. The value of each pixel for the matrix has a range from 0 to 255, where 0 indicating black and 255 pointing white. The feature descriptor (also called a kernel) is an element at the middle process that takes an input matrix, filters it, and compute the dot product. Finally, the feature map is produced. The convolution effects are depending on the operations such as edge detection or sharpen. The filter matrix will change based on the process that is needed. The objective of this step is to decrease the size of the image and make processing faster and easier.

Furthermore, the network will quickly identify the patterns in the new image. Many feature maps will prevent the loss of image information because every feature map identifies the location of features in the image. This operation of convolution will maintain the originality of the image.

ReLU stands for the Rectified Linear Activation Unit function, where it usually most commonly used after every convolution operation, and it is a non-linear operation. ReLU aims to increase non-linearity on CNN. Images are made of different objects that are not linear to each other. In this project, max pooling will be used. Max pooling is a method of down-sampling that is used to detect the most important features of the image. It is needed as it enables the convolutional neural network to recognize the object when presented with the image in any way. It allows the system to learn and detect the images with a different kind of position, pattern, angle, or textures. This will be useful as it can detect the small objects inside the image irrespective of where they are located, such as breast cancer tissues. There are many types of pooling, which is mean pooling, sum pooling, and max pooling.

The classification process is then performed, involving the fully connected layer where every neuron in the next layer is connected to every individual neuron in the previous layer. The objective of this process is to get the train, and test data combined the features into a wider variety of attributes that make the convolutional neural network more capable of classifying images as in the project, it develops its classes as cancerous and non-cancerous. Pooling and convolutional layer outputs will give high-resolution features of the breast cancer image. The fully connected layer based on the training dataset develops these features and categorized into cancerous and non-cancerous.

3.5. Types of CNN Layers

In our design of the CNN model, there are five types of layers: flatten, dense, LeakyRelu activation, dropout, and softmax.

3.5.1. Flatten Layer

The flatten layer is where the two-dimensional matrix of features is flattened into a vector (see figure 2). The flatten layer is placed in between the convolutional layer and the fully connected layer. After transforming into a vector, it fed into a fully connected CNN classifier.

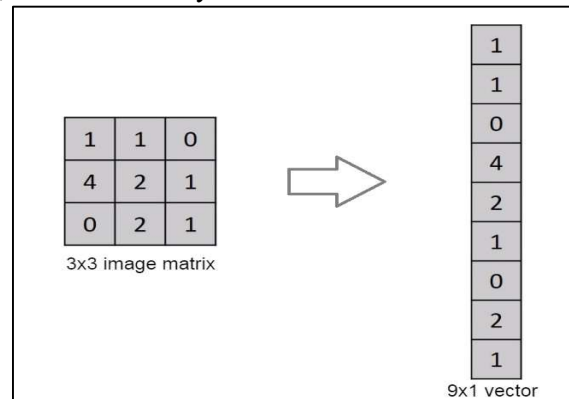


Figure 2. Process of 3x3 Matrix Flatten into 9x1 Vector [9]

3.5.2. Dense Layer

Generally, dense layers and fully connected layers are the same functions where each neuron in the next layer is connected to every single neuron in the previous layer. Dense formula is shown in equation (1):

$$\text{output} = \text{activation}(\text{dot}(\text{input}, \text{kernel}) + \text{bias}) \quad (1)$$

Where activation is the element-wise activation passed as the activation argument, the kernel is a weight matrix formed in the layer, and bias is a vector generated by the layer.

3.5.3. LeakyRelu Activation Layer

LeakyRelu, as illustrated in figure 3, is a type of activation layer which always use in the Keras compared to other activation function such as ReLU or Sigmoid. Leaky ReLU has a small slope for negative values, instead of altogether zero. For example, leaky ReLU may have $y = 0.01x$ when $x < 0$. The big advantage of LeakyRelu compared to the normal ReLU is that it fixed the “dying ReLU” problem because it does not have a zero-slope part, and also this activation layer can speed up the training process.

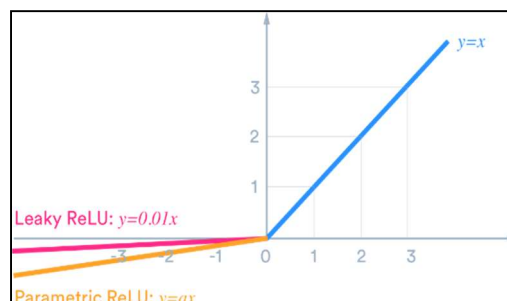


Figure 3. LeakyRelu graph display [10]

3.5.4. Dropout Layer

Deep learning neural networks are likely to quickly overfit a training dataset with few examples. The function of the dropout layer is used to overcome the overfitting when training the dataset. It is a very

efficient ways in term of performance in accuracy and loss to process a big dataset. Figure 4. shows that the process of the dropout layer handles the overfit data.

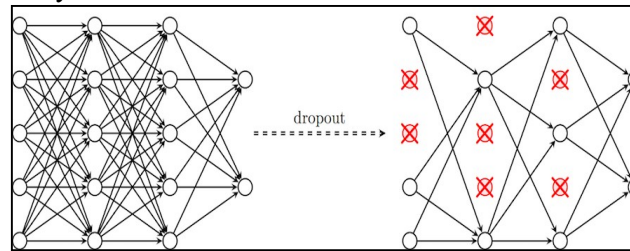


Figure 4. Process of dropout layer handles overfit data [11]

3.5.5. Softmax Layer

Softmax is implemented through a neural network layer just before the output layer. The Softmax layer must have the same number of nodes as the output layer. That is, Softmax assigns decimal probabilities to each class in a multi-class problem. Those decimal probabilities must add up to 1.0. This research has two types of classes that are Malignant and also Benign.

3.6. Keras and Transfer Learning

Keras is a high-level neural networks API (Application Programmable Interface), written in Python, and capable of running on top of TensorFlow. It is user-friendly and easy to implement because there are fewer variables to use for the model to run. When using Keras, there are a few steps that need to be considered. Firstly, training data is defined. Then, the neural network model needs to be defined, which is Keras sequential model. In Keras, there are two way to build the model, sequential and functional model. Keras sequential model is a model constructed of layers in a linear stack. Thirdly, the learning process are configured where in the sequential model, and there are three arguments used, which are Adam (Adaptive moment estimation) optimizer, categorical cross-entropy loss function, and accuracy metric. Lastly, all the data is passed to the training model.

Transfer learning is usually stated through the use of pre-trained models. A pre-trained model is a process where it that has been trained with a large amount of dataset to solve a problem similar to the one that currently want to be solved. When using Keras API, there are many pre-trained architectures that available such as Xception, MobileNet, DenseNet, InceptionV3, VGG16, VGG19, and many more. These pre-trained models that available can be used instead of start doing the model itself from the zero [12]. The pre-trained model needs to be modified by fine-tuning the model.

3.7. Pre-trained Model VGG16

VGG16 is used as a pre-trained model for transfer learning in this project. The model was trained with 14 million images from the ImageNet database of 1000 classes. The training of VGG16 was performed on NVIDIA Titan Black GPU, and training time was for weeks. As shown in figure 5, VGG16 architecture consists of 13 convolutional layers, five max-pooling layers, and three dense layers, which is in total 21 layers. However, only 16 are weight layers. The network has an image input size of 224 x 224 pixels. Transfer learning has an advantage in the training process of deep learning models in terms of reducing the size of the training dataset that required and time-consuming for the model to be trained. Nowadays, transfer learning has been suggested in the research field as one of the solutions to the insufficiency of training data in healthcare.

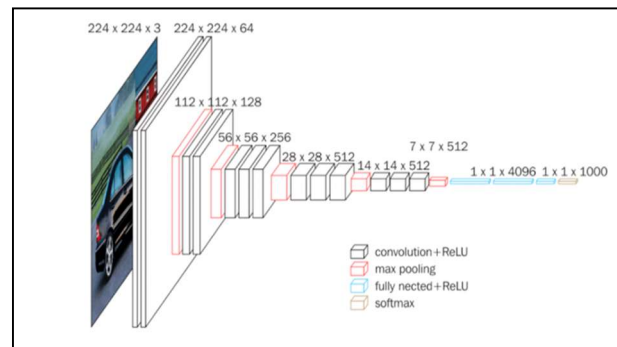


Figure 5. VGG16 Architecture [13]

3.8. Hardware

After the model has been completed and successfully trained and tested at the Google Colab, the completed model and testing images will be transferred and implemented on Raspberry Pi. The Raspberry Pi 3 B+ is used to show the efficiency of the model to run on the lightweight processor, as shown in figure 6.

The only limitation by using Raspberry Pi is that it cannot train a deep neural network because it does not have enough memory of CPU power to train deep neural networks from scratch [14]. Table 2. shows the specification of Raspberry Pi 3 B+.

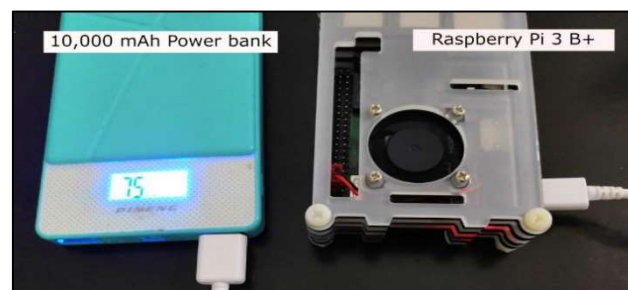


Figure 6. Raspberry Pi 3 B+ Connected to Power Supply

Table 2. Raspberry Pi 3 B+ Specifications

CPU type/speed	ARM Cortex-A53 1.4GHz
RAM size	1GB SRAM
Integrated Wi-Fi	2.4GHz and 5GHz
Ethernet speed	300 Mbps
PoE	Yes
Bluetooth	4.2

4. Accuracy Results

The performance of breast cancer detection by using transfer learning and without transfer learning is presented in this section. The model compared on both training and validation datasets. The training set is used to train the model. Meanwhile, the validation set is used to evaluate model performance. Ten epochs were tested to assess either the model is overfitting or underfitting. An epoch is to define how many times the algorithm sees the entire data set [15]. Every time the algorithm has seen all samples in the dataset, an epoch has finished. When training with a smaller epoch, it will tend to become underfitting, and a bigger epoch will lead to overfitting the data. The number of epochs used to train is

set to 10. The model can learn more by increasing the number of epochs, adding more layers, and training the dataset to improve accuracy.

4.1. Pre-Trained Model (with VGG16)

Table 3. shows the accuracy and loss based on each epoch by using transfer learning approach. Figure 7 shows the graph of the training and validation accuracy of the model.

Table 3. Accuracy and Loss (Using VGG16)

Number of epochs	Accuracy	Loss	Validation accuracy	Validation Loss
1	58.81%	11.36%	63.41%	61.61%
2	64.20%	69.30%	51.92%	74.00%
3	67.10%	65.28%	73.81%	52.11%
4	70.37%	58.82%	73.91%	54.10%
5	71.11%	56.27%	74.00%	50.28%
6	73.82%	53.05%	75.46%	49.56%
7	75.40%	51.30%	75.36%	49.79%
8	76.48%	49.27%	75.55%	51.77%
9	78.18%	45.39%	77.28%	50.33%
10	79.12%	42.54%	76.00%	54.30%

In this result, the highest training accuracy is 79% meanwhile for the validation accuracy is 76%. This means that this training model is expected to perform with 76% accuracy on the new data. For epoch from 8 to 10, when the training accuracy increases, the validation accuracy has slightly decreased. This means that the training model is fitting the training set better but slightly losing its ability to predict new data, which is the beginning of the data overfitting.

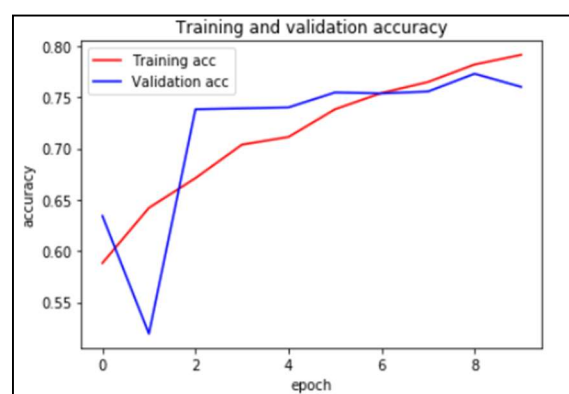


Figure 7. Training and Validation Accuracy Graph with VGG16

Figure 8. represents the graph of training and validation loss of the model. A loss function is used to improve the efficiency of the deep learning algorithm. At epoch 0, the training loss keeps decreasing, which means that the model is learning to recognize all the data set in the training set. Based on the graph, the minimum reading of training loss is 42% meanwhile for the validation loss is 49%. For the

epoch of 10, the training loss is 42%, and the validation loss is 54%. This means that the data has started to overfitting because the validation loss is higher than the training loss.

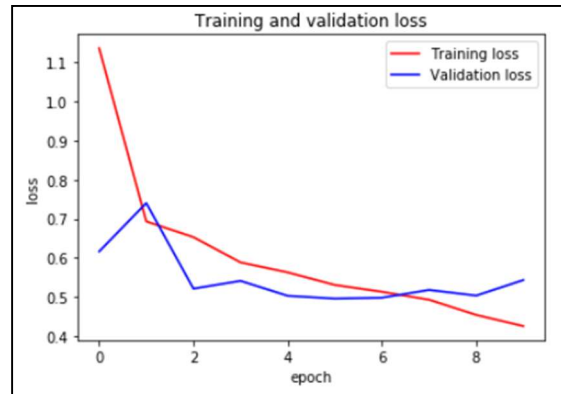


Figure 8. Training and Validation Loss Graph with VGG16

4.2. Without Pre-Trained Model (VGG16)

Figure 9. shows the graph of training and validation accuracy of the model and figure 9. represents the graph of training and validation loss of the model. The highest training accuracy is 69% meanwhile for the validation accuracy is 67%. This means that this training model expected to perform with 67% accuracy on the new data. For epoch from 4 to 10, when the training accuracy increases, the validation accuracy is inconsistent and when it comes to epoch 6, the graph trend becomes drastically decreased and suddenly sharply increased after it. This means that the training model is not fitting the training set and losing its ability to predict the new data, which is the beginning of the data overfitting.

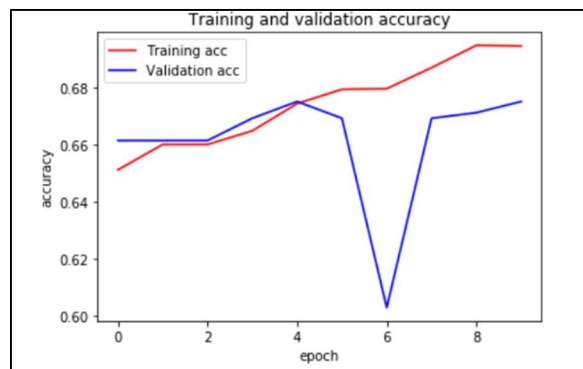


Figure 9. Training and Validation Accuracy Graph without VGG16

Figure 10. shows that the training loss keeps decreasing, which means that the model is learning to recognize all the data set in the training set. Based on the graph, the minimum reading of training loss is 55% meanwhile for the validation loss is 60%. For the epoch of 10, the training loss is 55%, and the validation loss is 63%. This shows that the data has started to overfit because the validation loss is higher than the training loss. Table-IV shows the accuracy and loss based on each epoch without using VGG16.



Figure 10. Training and Validation Loss Graph without VGG16

Table 4. Accuracy and Loss (Without VGG16)

Number of epochs	Accuracy	Loss	Validation accuracy	Validation Loss
1	65.11%	67.00%	66.14%	64.11%
2	66.01%	64.94%	66.14%	65.38%
3	66.01%	64.81%	66.14%	63.17%
4	66.49%	62.87%	66.93%	65.13%
5	67.44%	61.71%	67.51%	61.11%
6	67.94%	61.43%	66.93%	60.92%
7	67.97%	60.00%	60.27%	61.93%
8	68.71%	58.86%	66.93%	60.64%
9	69.49%	57.32%	67.12%	63.18%
10	69.47%	55.73%	67.51%	63.45%

4.3. Without Pre-Trained Model (VGG16)

After the model has been trained, the images been tested. Figure 11. shows the testing output for the test images of the 40X magnification factor on Google Colab with 79.07% accuracy.

```
Found 301 images belonging to 2 classes.
Time: 0:00:03.784870
Found 301 images belonging to 2 classes.
301/301 [=====] - 0s 334us/sample - loss: 0.4868 - acc: 0.7907
Test image, accuracy: 79.07%
```

Figure 11. Testing Accuracy

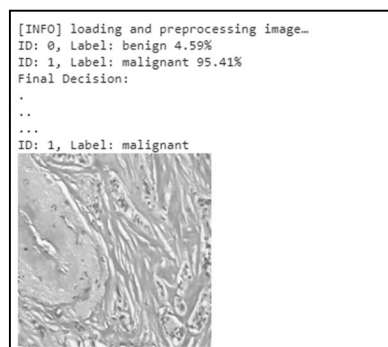
Table 5 shows the testing accuracy and loss that were achieved for the tested histopathological images based on every magnification factor of 40X,100X,200X, and 400X.

Table 5. Testing Accuracy and Loss

Magnification Factor	Accuracy	Loss
40X	79.07%	48.68%
100X	70.00%	59.12%
200X	80.20%	44.80%
400X	74.09%	58.95%

As shown in figure 12, a single histopathological image with a 40X magnifying factor has been taken from Google Drive to be tested in Google Colab. The testing image can be chosen based on four different magnify factors (40X, 100X, 200X, 400X).

After the image has processed in the model, it will display the output of the ID 0 and the label of Benign and ID1 with the label of Malignant. It will also display the percentage of accuracy based on the detected image. Based on the output, the image was detected as Malignant with an accuracy of 95.41%, and it also recognizes the image as Benign with a low accuracy percentage of 4.59%. It displays that the final decision of that output is ID:1 with the label: Malignant based on the highest rate of accuracy.

**Figure 12.** Sample Image Detection Result for Cancer (malignant) on Colab

4.4. Hardware Implementation

Figure 13. and Figure 14. show the output every time the single image was tested. The model has been deployed on Raspberry Pi to show that it can run on a lightweight processor. The images can be chosen to be prove based on different magnifying factors.

Figure 13. shows the output of breast cancer detection, where it detects benign with an accuracy percentage of 67.34% for benign and 32.66% for Malignant. The highest accuracy shows the final output, which it recognized as Benign.

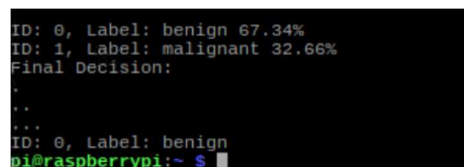
**Figure 13.** Result of Sample Cancer (Benign) Image on Raspberry Pi

Figure 14. illustrates how the model detects the image with a percentage of 8.13% for Benign and 91.87% for Malignant. The highest accuracy shows the final output, which is recognized as Malignant with a high rate of accuracy.

```

ID: 0, Label: benign 8.13%
ID: 1, Label: malignant 91.87%
Final Decision:
.
.
.
ID: 1, Label: malignant
pi@raspberrypi:~$

```

Figure 14. Result of Sample Cancer (Malignant) Image on Raspberry Pi

The graphical user interface (GUI) is used to display the result on the Raspberry Pi. This GUI is created by using AppJar platform and developed by using Python language. It combined the backend process with the CNN model that have been tested. AppJar can be run on Linux and Windows. Figure 15. shows that the verified image has detected as Benign.

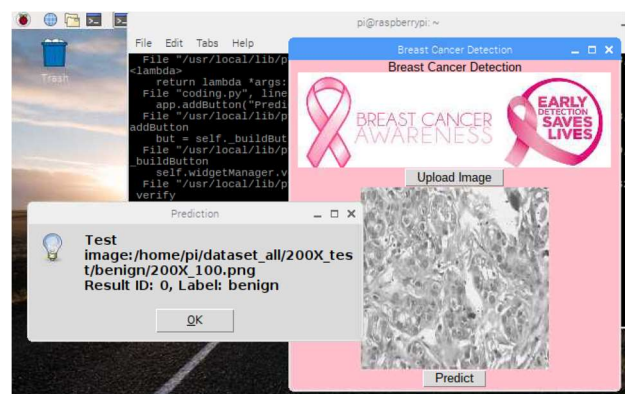


Figure 15. GUI of Breast Cancer Detection System

5. Conclusion

Breast cancer detection by using histopathological images on an IoT board is proposed by using the deep learning method. It detects whether the histopathological images of the patients are Malignant or Benign. The histopathological images are taken from the BreakHis database with different criteria, such as magnifying factors (40X, 100X, 200X, 400X). Two images pre-processing method has been done, RGB to grayscale and image resizing. The type of neural network model used is Convolutional Neural Network (CNN), that was trained by using a GPU.

The model that developed by using transfer learning (VGG16) shows 76% accuracy compared to the model developed without using VGG16, which is the accuracy is 67.51%. We also found out that 200X of the magnifying factor of microscopic images perform better in term of testing accuracy and loss compared to 40X, 100X, and 400X. The model is successfully deployed on Raspberry Pi to show that it can run on the lightweight processor, and it is a portable device. This breast cancer detection system is recommended to be implemented at the hospital or clinic since it considered as a low-cost system and easy to access.

In conclusion, this research outcome will serve as a preliminary step for our research in using deep learning for breast cancer detection based on histopathology images and then apply it on IoT boards and devices. In future works, we plan to improve the model through better network design and ensemble of deep learning algorithms by incorporating transfer learning approaches.

6. Reference

- [1] N. Syahida, "What Is Breast Cancer and It's Statistic in Malaysia — THREAD by ZALORA Malaysia", THREAD by ZALORA Malaysia, 2018. [Online]. Available: <https://thread.zalora.com.my/home/2018/10/what-is-breast-cancer-and-its-statistic-in-malaysia>. [Accessed: 21- Jun- 2019].
- [2] Q. Bani baker, T. abu zaitoun, S. Banat, E. Eaydat, and M. Alsmirat, "Automated Detection of Benign and Malignant in Breast Histopathology Images," in *2018 IEEE/ACS 15th International*

- Conference on Computer Systems and Applications (AICCSA)*, 2018, pp. 1–5, doi: 10.1109/AICCSA.2018.8612799.
- [3] L. M. Al-Hayali, M. Morsy, and M. M. Abdul-Razak, "Microscopic Image Processing of Automated Detection and Classification for Human Cancer Cell," *Int. J. Sci. Technol. Res.*, vol. 4, no. 8, 2015.
 - [4] R. Janghel, A. Shukla, R. Tiwari, and R. Kala, "Intelligent Decision Support System for Breast Cancer," in *Advances in Swarm Intelligence, First International Conference, ICSI 2010, Beijing, China, June 12-15, 2010, Proceedings*, 2010, pp. 351–358, doi: 10.1007/978-3-642-13498-2_46.
 - [5] K. Rani, "Parallel Approach for Diagnosis of Breast Cancer using Neural Network Technique," *Int. J. Comput. Appl.*, vol. 10, Nov. 2010, doi: 10.5120/1465-1980.
 - [6] R. Yan *et al.*, "Breast cancer histopathological image classification using a hybrid deep neural network," *Methods*, vol. 173, pp. 52–60, 2020, doi:
 - [7] P. Sahu, D. Yu, and H. Qin, "Apply lightweight deep learning on the internet of things for low-cost and easy-to-access skin cancer detection," in *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, 2018, p. 37, doi: 10.1117/12.2293350.
 - [8] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, "A Dataset for Breast Cancer Histopathological Image Classification," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 7, pp. 1455–1462, Jul. 2016, doi: 10.1109/TBME.2015.2496264.
 - [9] S. Saha, "A Comprehensive Guide to Convolutional Neural Networks—the ELI5 way", *Medium*, 2018. [Online]. Available: <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>. [Accessed: 10- Jul-2019].
 - [10] M. Art, "Generative Adversarial Networks - Part III", *Makeyourownalgorithmicart.blogspot.com*, 2019. [Online]. Available: <http://makeyourownalgorithmicart.blogspot.com/2019/05/generative-adversarial-networks-part-iii.html>.
 - [11] S. Serengil, "Handling Overfitting with Dropout in Neural Networks | sefiks.com", *Sefik Ilkin Serengil*, 2020. [Online]. Available: <https://sefiks.com/2018/03/19/handling-overfitting-with-dropout-in-neural-networks>.
 - [12] Shallu and R. Mehra, "Breast cancer histology images classification: Training from scratch or transfer learning?", *ICT Express*, vol. 4, no. 4, pp. 247-254, 2018. Available: 10.1016/j.icte.2018.10.007.
 - [13] M. Hassan, "VGG16 - Convolutional Network for Classification and Detection", *Neurohive.io*, 2020. [Online]. Available: <https://neurohive.io/en/popular-networks/vgg16/>.
 - [14] "Raspberry Pi and machine learning: How to get started", *TechRepublic*, 2018. [Online]. Available: <https://www.techrepublic.com/article/raspberry-pi-and-machine-learning-how-to-get-started/>.
 - [15] J. Brownlee, "Difference Between a Batch and an Epoch in a Neural Network", *Machine Learning Mastery*, 2018. [Online]. Available: <https://machinelearningmastery.com/difference-between-a-batch-and-an-epoch/>.