Breast Cancer Classification with VGG16

Abstract

Breast cancer remains one of the most severe public health problems. The early diagnosis of this disease is vital to avoid the development of malignancy. In this study, a deep convolutional neural network (CNN) model based on VGG16 (VGG16) for breast cancer classification is proposed. The model architecture is utilised to extract high-level features from the input images. The experimental results showed that the proposed model outperformed various techniques in terms of accuracy, loss, sensitivity, and specificity.

Introduction

Healthy breast cells start to change and enlarge out of control to produce a mass or sheet of cells called a. A could be benign or cancerous. Malignant describes a cancer's capacity to grow and spread to various body parts. A benign growth is one that has not yet spread and is still developing. Even though breast cancer most usually travels to adjacent lymph nodes, in which case it is still recognised as a local or regional disease, it can move farther through the body through blood arteries and/or lymph nodes to places including the bones, lungs, liver, and brain. Metastatic, or stage IV, breast cancer is the most advanced stage. The presence of adjacent lymph nodes alone, however, is not usually a sign of stage IV breast cancer. Breast cancer can be either invasive or non-invasive. Invasive breast cancer is breast cancer that has spread to neighbouring tissues or distant organs. The extent of non-invasive breast cancer is limited to the breast or milk ducts. Breast cancers can take on many shapes and types depending on how they look under a microscope. Breast cancer comes in a variety of forms, such as Invasive (infiltrating) Carcinoma. This cancer starts in your breast's milk ducts, breaks through the duct wall, and then spreads to the adjacent breast tissue.

This kind of breast cancer is the most prevalent, accounting for more than 80% of cases. Diffuse carcinoma is the second type. Because the cancerous cells haven't progressed past your milk ducts, some people believe carcinoma in, also known as stage 0 breast cancer, to be precancerous. The medical condition can be easily treated. To stop the cancer from growing more aggressive and from spreading to other tissues, immediate therapy is required. The third type is invasive (infiltrating) carcinoma. The breast, where breast milk is produced, is where this cancer first appears. It has since moved to the close-by breast tissue. It is to blame for 10% to 15% of breast cancer cases.

The most effective deep CNNs for classification in the field of computer vision are VGG16. The ImageNet dataset (1.3 million natural images) with 1000 classes served as the pre-training data for VGG16. The graphics have a three-colour channel layout with a patch size of 224 by 244 pixels. The one-channel, greyscale mammography pictures in this work will be concatenated along its third dimension in order to fully utilise the deep models. In addition, rather than using 1000 classes as the final layer, a new classification layer for the two classes of benign and malignant is used. The mammography dataset must then begin to be fine-tuned after some parameters have been defined. To retrain the VGG16 model, the iteration number and primary learning should be 105 and 104, respectively.

Methods

Knowledge Acquisition

The breast histopathology images from the breast class dataset for our Breast Cancer Classification project were obtained from Universiti Kebangsaan Malaysia Hospital. The given dataset comprises three grades of invasive breast cancer which includes; **Grade 1 (Well Differentiated), Grade 2 (Moderately Differentiated) and Grade 3 (Poorly Differentiated).** Breast cancers are given a grade according to: how different the cancer cells are to normal breast cells and how quickly they are growing. A cancer’s grade is determined when a doctor looks at the cells under a microscope, using tissue from biopsy or after breast cancer surgery.

Low grade or Grade 1, the cells will look like normal breast cells and usually slow in growing.These are well differentiated cancers. They are arranged in small tubules for ductal cancer and cords for lobular cancer. These cancers tend to grow and spread slowly and have a good outlook (prognosis). Furthermore, Intermediate grade or Grade 2, will look less from normal cells and can grow much faster. They are moderately differentiated. This means the features and outlook (prognosis) are somewhere between well and poorly differentiated. Lastly, High grade or Grade 3, looks really different to normal breast cells and will grow much faster than previous grades. These are poorly differentiated cancers that have abnormal features. They tend to grow and spread more quickly and have a worse outlook (prognosis).

Figure 1 shows the number of dataset for each Breast Cancer Grades.

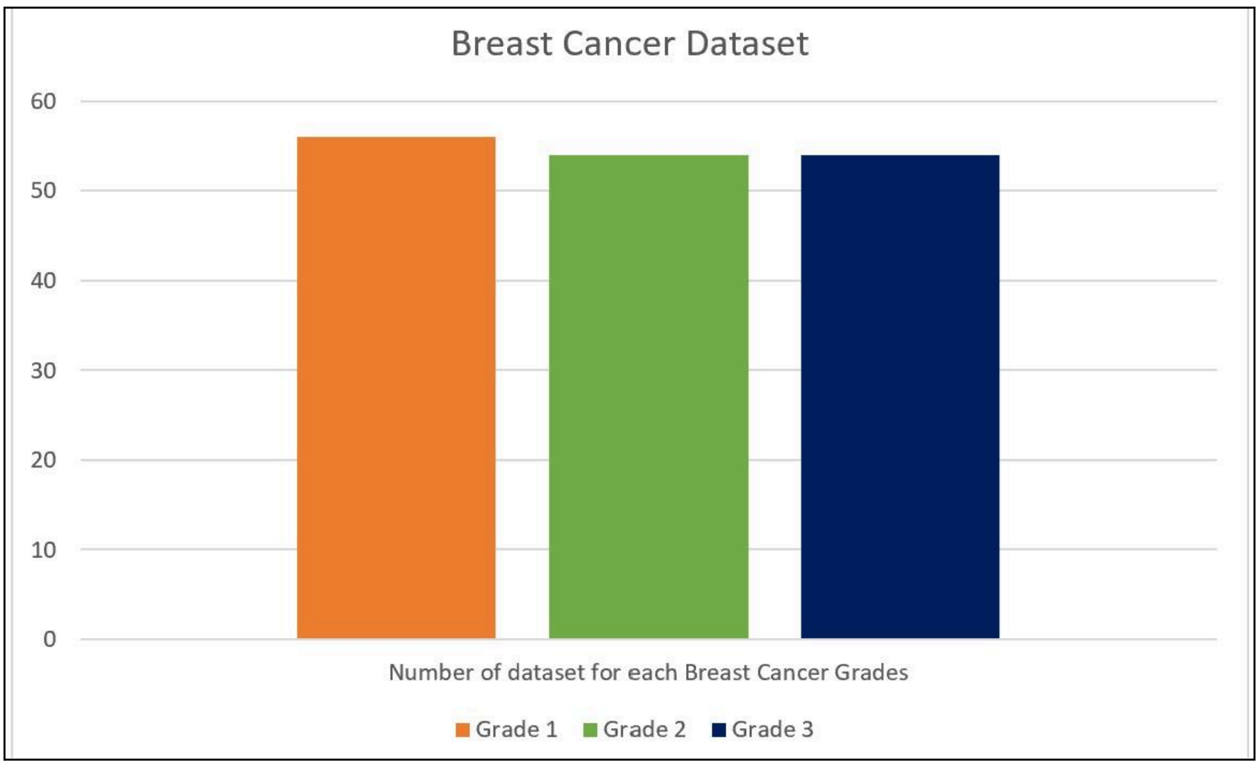


Figure 1 Breast Cancer Dataset

Figure 2 shows a few breast histopathology images with different grades of cancer. The top row shows the Grade 1 tumour. The appearance of the Grade 1 tumour is well-differentiated. The cells appear to be normal, growing slowly and are not aggressive. Next, the second row shows the Grade 2 tumour where its appearance is moderately-differentiated. This shows that the cells appear to be semi-normal and growing moderately fast. Lastly, the third row shows the sample images of Grade 3 tumours which are poorly-differentiated. It shows that the cells appear to be abnormal, growing quickly and aggressive.

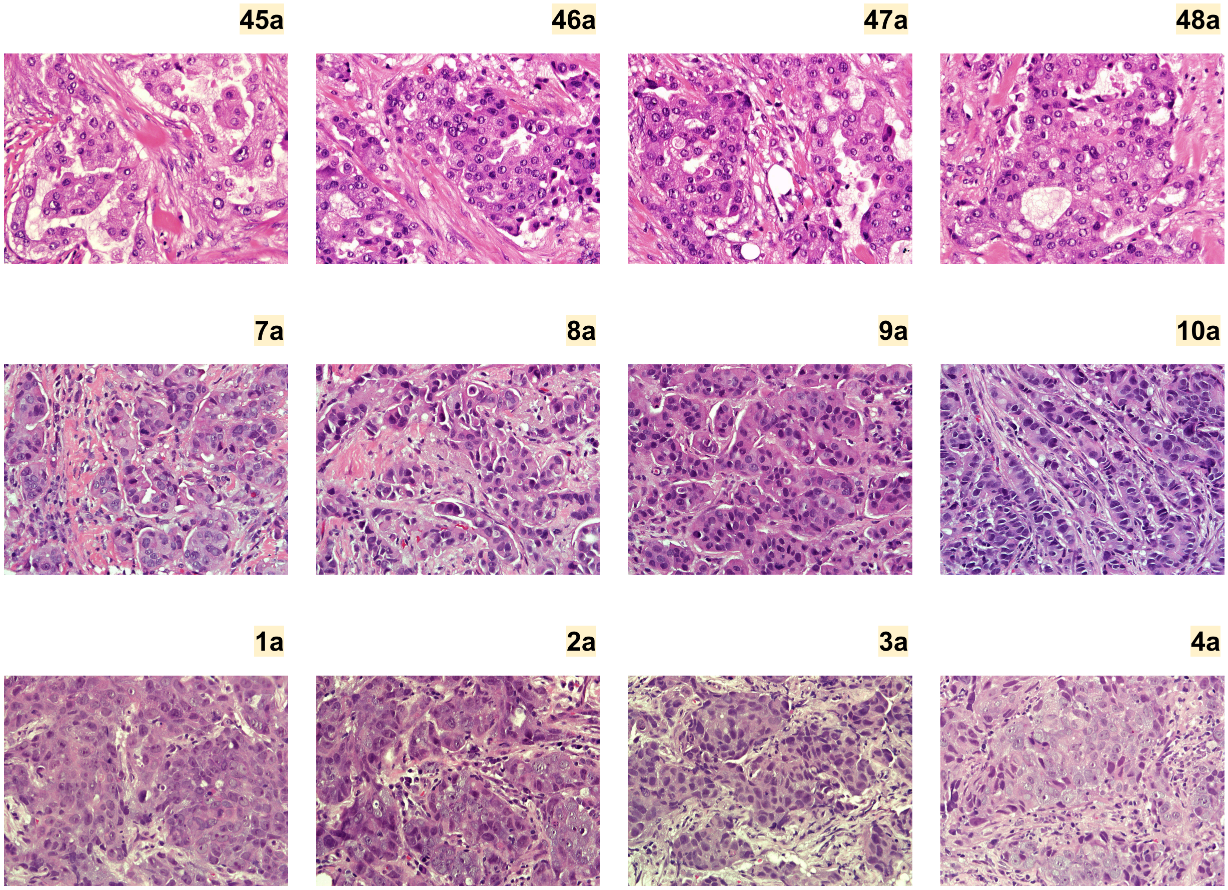


Figure 2 Breast histopathology images

Knowledge Representation

Knowledge representation is an important chain for any artificial intelligence system, including Automated breast cancer diagnosis.In essence, it is a study of the best ways to describe an intelligent agent's beliefs, intents, and judgements for automated reasoning. Modelling intelligent behaviour for an agent is one of the main goals of knowledge representation. Knowledge representation is not just storing data in some database. Still, it also enables an intelligent device to learn from that knowledge and experience to behave intelligently like a human.

Breast Cancer (BC) is measured by 5 attributes in our system that determines whether users have the said disease. The 5 attributes include growth of lump on breast, mammography, malignancy, prognosis and grade. The diagnosis is carried out by obtaining the condition of the user based on the data gathered. From there, a decision tree is made out to represent the chain of knowledge. Decision trees provide an effective method of decision making because they: Clearly lay out the problem so that all options can be challenged. Not only that, it allows us to analyse fully the possible consequences of a decision. Next, decision trees are able to provide a framework to quantify the values of outcomes and the probabilities of achieving them.

Figure 3 shows the decision tree of Breast Cancer (BC) grade diagnosis.

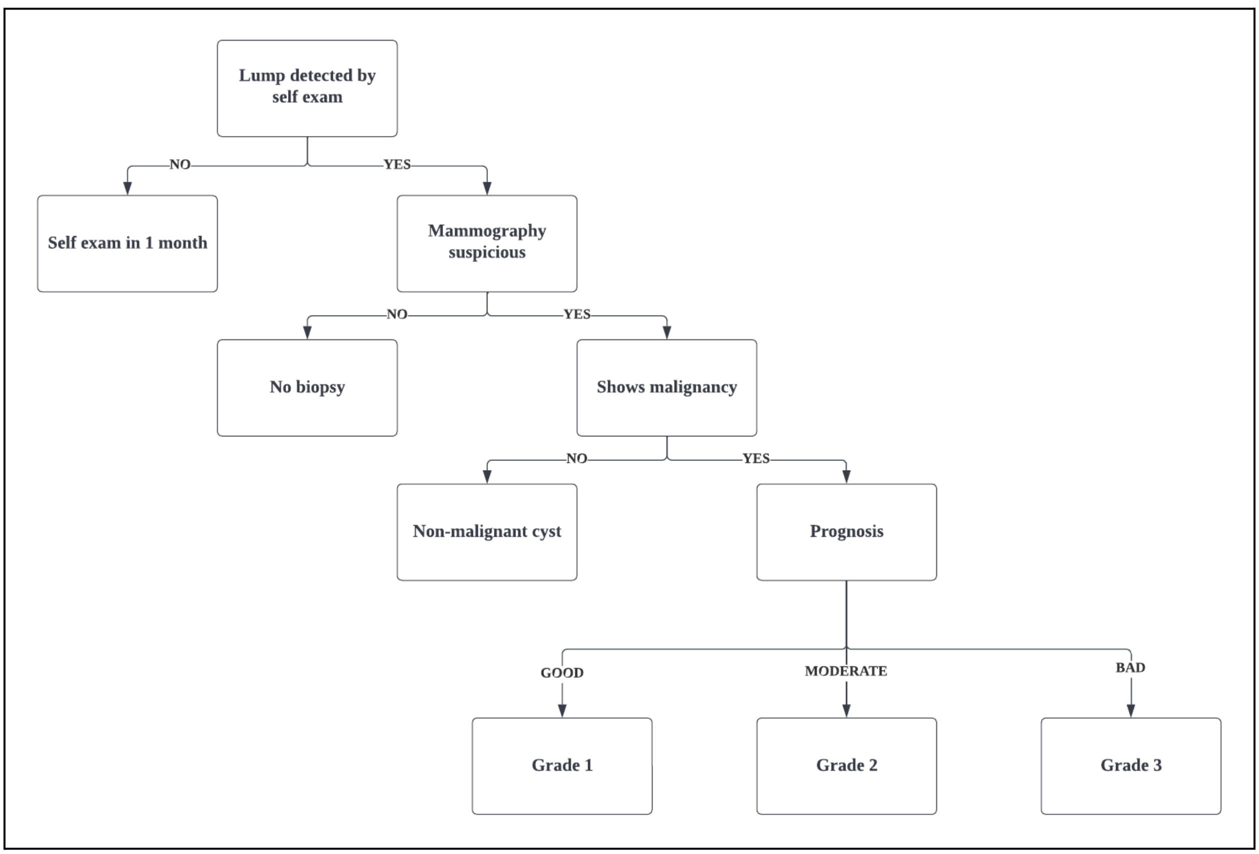


Figure 3 Decision Tree

The machine learning techniques

Oxford University introduced VGG16 as one of the most common deep learning architectures. It has 41 layers that have been disrupted as follows: 16 weight layers, 13 convolutional layers, and 3 FC layers. On all Conv Layers with stride one, VGG16 uses a small 3x3 kernel. Conv Layers are always followed by Max Pooling Layers. The VGG16 input is a fixed 224x224 three-channel image. The three FC layers in VGG16 have varying depths. The first two FC have the same channel size (4096), while the final FC has a channel size of 1000, which represents the number of class labels in the ImageNet dataset. The output layer is the soft-max layer, which is in charge of the probability assigned to the input image.

An overview of the VGG architecture is provided below:

- *INPUT* : An image of size 224x224 is fed into VGGNet. By removing a 224224 square from the centre of each image submitted for the ImageNet competition, the model's developers were able to maintain a constant image input size.

- *CONVOLUTIONAL LAYERS* : The smallest 33 receptive field is used by the convolutional filters of the VGG algorithm. A 11 convolution filter is also used by VGG to linearly transform the input.

- *ReLu ACTIVATION* : The Rectified Linear Unit Activation Function (ReLU) component, a key advancement made by AlexNet to shorten training times, comes next. ReLU is a linear function that outputs zero for negative inputs and a matching output for positive inputs. To maintain spatial resolution after convolution, VGG has a predetermined convolution stride of 1 pixel (the stride value represents how many pixels the filter "moves" to cover the complete space of the picture).

- *HIDDEN LAYERS* : ReLU is used throughout the whole VGG network's hidden layers as opposed to Local Response Normalisation, like in AlexNet. The latter lengthens training sessions and uses more memory, with very minor gains in accuracy.

- *POOLING LAYERS* : The number of parameters and dimensionality of the feature maps produced by each convolution phase are decreased by adding a pooling layer after a series of convolutional layers. Given the quick increase in the number of possible filters from 64 to 128, 256, and finally 512 in the last layers, pooling is essential.

- *FULLY CONNECTED LAYERS* : Three completely interconnected layers make up VGGNet. Each of the first two layers contains 4096 channels, while the third layer contains 1000 channels—one for each class.

The breast dataset includes a total of 164 breast histopathology images which were split into 2 folders that each contain 3 classes . In the class Grade 1, there are 41 training images and 15 validation images. Next, in the class Grade 2, there are 39 training images and 15 validation images. Lastly, in the class Grade 3, there are 39 training images and 15 validation images.

Table 1 shows the number of training, testing and validation images.

Class	Count of images	Training Images	Testing / Validation Images
Grade 1	56	41	15
Grade 2	54	39	15
Grade 3	54	39	15

Table 1 Number of training, testing and validation images

Image augmentation is a technique of applying different transformations to original images which results in multiple transformed copies of the same image. In this expert system, we utilised the Keras ImageDataGenerator class which can provide real-time data augmentation. We augment the data by using an augmentation technique that is preprocessing input. Then, flow_from_directory() method is used to read the images directly from the directory and augment them while the neural network model is learning on the training data. The class_mode is set to categorical as it has more than two classes to predict.

We then created a Sequential Model and added the base of the model that is VGG16 into it. We also created a few more layers which also will be added into the model. The layer that was created is first the layer Flatten which is used to flatten the output layer into 1 dimension. Second is the layer Dense, which adds a fully connected layer with 512 hidden units with ReLu activation. Third is layer Dropout with a rate of 0.5, which helps to prevent overfitting. Lastly, a softmax layer with 3 nodes for classification output.

Optimizers are algorithms or methods that are used to change or tune the attributes of a neural network such as layer weights, learning rate, etc. in order to reduce the loss and in turn improve the model. In this expert system, we used the Adam Optimizer with a learning rate of 0.0005. Adam (Adaptive Moment Estimation) is an adaptive optimization algorithm that was created specifically for deep neural network training. It scales the learning rate using squared gradients and leverages momentum by using the gradient's moving average rather than the gradient itself.

Evaluating a model is an important step in developing an effective machine learning model. We used accuracy and loss to assess the model's performance in this expert system. One of the most common evaluation metrics in classification problems is accuracy, which is defined as the total number of correct predictions divided by the total number of predictions made for a dataset and loss which indicate how bad the model prediction. After obtaining the accuracy and loss, a graph is then plotted to represent the data.

Results (Diagram and Table)

Interface

About 56 of Grade 1, 54 of Grade 2 and 54 of Grade 3 of breast cancer cell image have been given and used to demonstrate the effectiveness of the proposed technique. The database covers various cases of breast cancer abnormality and density of the breast tissues.

Input

```
Found 119 images belonging to 3 classes.
Found 45 images belonging to 3 classes.
```

Figure 4

Figure 4 shows the data have been split into 2 variables which are for Training and Validation. For Training, it contains 81 images belonging to 3 classes while Validation contains 83 images also belonging to 3 classes. The 3 classes inside the two folders are the folder that contain data images for Grade 1, Grade 2 and Grade 3 for breast cancer that have been mentioned above which the data have split equally into both the Training and Validation.

Output

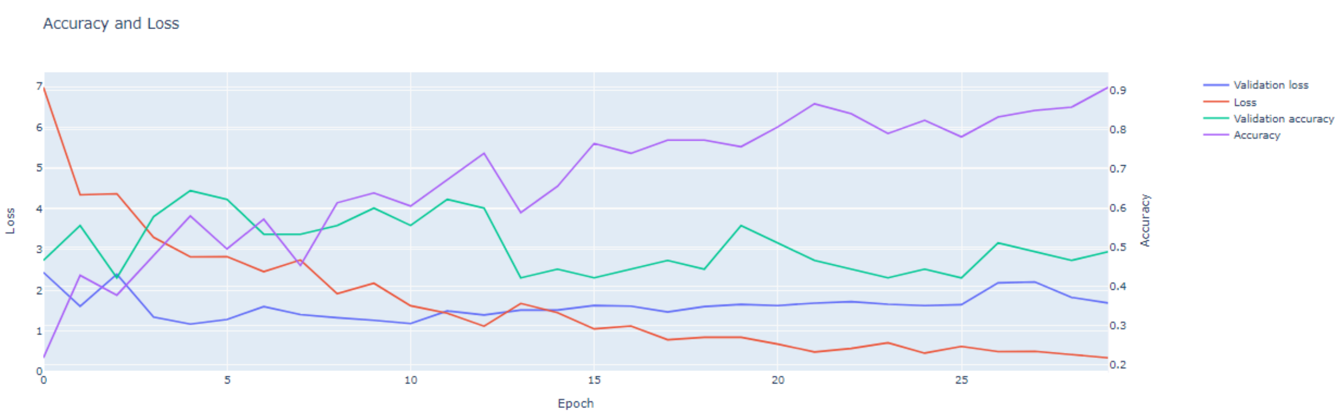


Figure 5

Figure 5 shows the graph of accuracy and loss from training the model with the classifier VGG16. The graph above, can be concluded with the performance of the accuracy will get higher if we train or compile the model with higher epochs or higher cycle of the full training dataset.

Machine learning results

```
Epoch 1/30
4/4 [==============================] - 69s 18s/step - loss: 6.9787 - acc: 0.2185 - val_loss: 2.4386 - val_acc: 0.4667
Epoch 2/30
4/4 [==============================] - 68s 18s/step - loss: 4.3418 - acc: 0.4286 - val_loss: 1.6060 - val_acc: 0.5556
Epoch 3/30
4/4 [==============================] - 66s 17s/step - loss: 4.3680 - acc: 0.3782 - val_loss: 2.3947 - val_acc: 0.4222
Epoch 4/30
4/4 [==============================] - 67s 19s/step - loss: 3.2998 - acc: 0.4790 - val_loss: 1.3415 - val_acc: 0.5778
Epoch 5/30
4/4 [==============================] - 66s 19s/step - loss: 2.8212 - acc: 0.5798 - val_loss: 1.1693 - val_acc: 0.6444
Epoch 6/30
4/4 [==============================] - 67s 19s/step - loss: 2.8246 - acc: 0.4958 - val_loss: 1.2849 - val_acc: 0.6222
Epoch 7/30
4/4 [==============================] - 66s 17s/step - loss: 2.4570 - acc: 0.5714 - val_loss: 1.5989 - val_acc: 0.5333
Epoch 8/30
4/4 [==============================] - 67s 19s/step - loss: 2.7440 - acc: 0.4538 - val_loss: 1.4043 - val_acc: 0.5333
Epoch 9/30
4/4 [==============================] - 74s 20s/step - loss: 1.9156 - acc: 0.6134 - val_loss: 1.3274 - val_acc: 0.5556
Epoch 10/30
4/4 [==============================] - 69s 19s/step - loss: 2.1721 - acc: 0.6387 - val_loss: 1.2642 - val_acc: 0.6000
Epoch 11/30
4/4 [==============================] - 66s 19s/step - loss: 1.6201 - acc: 0.6050 - val_loss: 1.1822 - val_acc: 0.5556
Epoch 12/30
4/4 [==============================] - 67s 19s/step - loss: 1.4350 - acc: 0.6723 - val_loss: 1.4926 - val_acc: 0.6222
Epoch 13/30
4/4 [==============================] - 68s 18s/step - loss: 1.1166 - acc: 0.7395 - val_loss: 1.3934 - val_acc: 0.6000
Epoch 14/30
4/4 [==============================] - 67s 19s/step - loss: 1.6772 - acc: 0.5882 - val_loss: 1.5144 - val_acc: 0.4222
Epoch 15/30
4/4 [==============================] - 66s 17s/step - loss: 1.4493 - acc: 0.6555 - val_loss: 1.5145 - val_acc: 0.4444
Epoch 16/30
4/4 [==============================] - 67s 17s/step - loss: 1.0520 - acc: 0.7647 - val_loss: 1.6274 - val_acc: 0.4222
Epoch 17/30
4/4 [==============================] - 72s 18s/step - loss: 1.1237 - acc: 0.7395 - val_loss: 1.6109 - val_acc: 0.4444
Epoch 18/30
4/4 [==============================] - 69s 19s/step - loss: 0.7871 - acc: 0.7731 - val_loss: 1.4669 - val_acc: 0.4667
Epoch 19/30
4/4 [==============================] - 66s 19s/step - loss: 0.8451 - acc: 0.7731 - val_loss: 1.6013 - val_acc: 0.4444
Epoch 20/30
4/4 [==============================] - 67s 17s/step - loss: 0.8451 - acc: 0.7563 - val_loss: 1.6543 - val_acc: 0.5556
Epoch 21/30
4/4 [==============================] - 68s 18s/step - loss: 0.6797 - acc: 0.8067 - val_loss: 1.6255 - val_acc: 0.5111
Epoch 22/30
4/4 [==============================] - 67s 17s/step - loss: 0.4853 - acc: 0.8655 - val_loss: 1.6859 - val_acc: 0.4667
Epoch 23/30
4/4 [==============================] - 66s 17s/step - loss: 0.5722 - acc: 0.8403 - val_loss: 1.7207 - val_acc: 0.4444
Epoch 24/30
4/4 [==============================] - 70s 18s/step - loss: 0.7115 - acc: 0.7899 - val_loss: 1.6584 - val_acc: 0.4222
Epoch 25/30
4/4 [==============================] - 70s 18s/step - loss: 0.4578 - acc: 0.8235 - val_loss: 1.6248 - val_acc: 0.4444
Epoch 26/30
4/4 [==============================] - 68s 19s/step - loss: 0.6206 - acc: 0.7815 - val_loss: 1.6472 - val_acc: 0.4222
Epoch 27/30
4/4 [==============================] - 89s 25s/step - loss: 0.4940 - acc: 0.8319 - val_loss: 2.1825 - val_acc: 0.5111
Epoch 28/30
4/4 [==============================] - 74s 21s/step - loss: 0.5030 - acc: 0.8487 - val_loss: 2.2039 - val_acc: 0.4889
Epoch 29/30
4/4 [==============================] - 72s 19s/step - loss: 0.4231 - acc: 0.8571 - val_loss: 1.8256 - val_acc: 0.4667
Epoch 30/30
4/4 [==============================] - 69s 18s/step - loss: 0.3430 - acc: 0.9076 - val_loss: 1.6871 - val_acc: 0.4889
```

Figure 6 Result in machine learning

Figure above shows the testing and graph result that we obtained using the VGG16 algorithm which includes the accuracy of training and validation along with the loss of data quality. As the figure shows, the number of times the model cycle through the data or we call it an epoch is 30 times and this is because we are lacking data. In order for the machine to learn, a large sum of data was needed for the machine but we only have in total 164 of data images for both training and validation. Concerning the lack of data, we choose to increase our epoch into 30 so the machine can learn from the previous data again until we achieve a satisfied result.

This result can be seen in the figure below showing the accuracy of both training and validation together with the loss of data quality. The first epoch scored a poor result achieving accuracy only of 21% for training and 33% for validation, both with a high loss. But when running it again, we can see that the loss is getting lower and the accuracy is getting higher meaning that machine is actually learning. On the 15th epoch, we achieved an accuracy of 82% for training and 57% for validation. Both show great results but still suffer from high loss. But on the 30th epoch, the final result shows that we achieved accuracy of 91% in training and 49% for validation. On the other hand, the loss percentage of training was 34% and 160% for validation.The reason which we believe why validation suffered a high loss and a middle score was because of lack of data in validation but can be corrected with more data in validation along with much higher epoch.

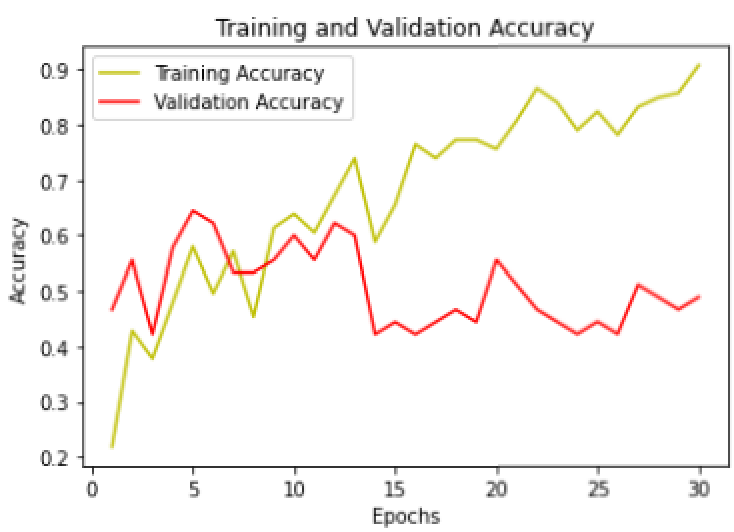


Figure 7 Graph of the Training and Validation Accuracy for Breast Cancer data.

Figure 8 Graph of the Training and Validation Loss for Breast Cancer data.

Figure 7 and 6 was a graph also called as learning curves for both Training and Validation in terms of accuracy and loss in data quality. This graph was created through the data in figure 6 and was stored in a variable which later was used to plot the graph by using `matplotlib`. From the graph, we can see that the accuracy for both training and validation actually increases by each epoch and once reached a maximum accuracy of 91% for training and 64% validation . As the loss, the line plot gradually decreases with each epoch and reaches a minimum value for both training and validation of 34% and 110%. for both training and validation. Both of the figures show that training and validation datasets are suitably representative.

Undeniably, machine learning with the classification algorithm, VGG16, has actually shown a great result in classifying the images with the dataset of breast cancer.

Analysis

Inference Engine

Since it serves as the system's primary processing component, the inference engine is referred to as the expert system's "brain". It uses the knowledge base and inference rules to draw conclusions or infer new information. Also, it aids in determining an error-free response to user requests. Expert systems use either forward or backward chaining for proposing solutions or deducing information.

In this expert system, forward chaining is more appropriate for diagnosing breast cancer disease because it is data-driven and uses available data to make a decision. Forward-chaining provides a solid foundation for reaching conclusions, particularly in medical diagnosis.

Deducing the grade of breast cancer cells begins with determining if the patient has symptoms of breast cancer (e.g Lump in breast). Doctors then use a mammogram to look for early signs of breast cancer. If the mammogram is suspicious, then cancer cells are removed from the breast and checked in the lab to be given a grade. The grade is then used for the patient's prognosis.

Rules:

a. IF lump detected in breast AND mammography suspicious AND shows malignancy AND prognosis good THEN Grade 1.
b. IF lump detected in breast AND mammography suspicious AND shows malignancy AND prognosis moderate THEN Grade 2.
c. IF lump detected in breast AND mammography suspicious AND shows malignancy AND prognosis bad THEN Grade 3.
d. IF lump detected in breast AND mammography suspicious AND non-malignant cyst THEN non-cancerous.
e. IF lump detected in breast AND mammography not suspicious THEN no biopsy needed.
f. IF lump not detected in breast THEN self exam in 1 month.

Handling uncertainties and conflicting knowledge

There are numerous sources of uncertainty, including variation in specific data values and the sample of data collected from the domain. So to represent uncertain knowledge, where we are not sure about the predicates, we need uncertain reasoning or probabilistic reasoning. Uncertainties can be solved by using the Certainty Factor. The Certainty Factor (CF) is a numeric value which tells us about how likely an event or a statement is supposed to be true. The value of the Certainty factor lies between -1.0 to +1.0, where the negative 1.0 value suggests that the statement can never be true in any situation, and the positive 1.0 value defines that the statement can never be false.

The two conflicting knowledge that we can identify from our expert system is:

1. IF lump detected in breast AND mammography suspicious THEN shows malignancy. (MB=0.8)
2. IF lump detected in breast AND mammography suspicious THEN non-cancerous. (MD =0.3)

Certainty Factor calculation:
Cf (cancer) = MB (cancer) - MD (cancer)
Cf (cancer) = 0.8 - 0.3
Cf (cancer) = 0.6 ≈ **60%**

Conclusion

Extraction of high-level traits from breast histopathology images can improve the diagnostic process' effectiveness. In order to extract high-level characteristics from breast pictures, this study's primary goal is to use VGG16, a pre-trained model from CNN deep learning. To do that, we eliminated the final layer in VGG16 that was entirely connected. VGG-16 was one of the best performing architectures. The performance of the accuracy will increase if we train or assemble the model with more epochs or more cycles of the entire training dataset, according to the accuracy from training the model using the classifier VGG16. The results of this study show that image processing and augmentation approaches can considerably improve a model's performance, allowing this expert system to be applied to real-world scenarios. In computer vision, transfer learning can also be a suitable strategy when dealing with a limited number of images. It might help medical professionals make an early diagnosis of planning.

Reference

Cancer grade. Breast Cancer Now. (2022, December 9). Retrieved January 12, 2023, from https://breastcancernow.org/information-support/facing-breast-cancer/diagnosed-breast-cancer/cancer-grade

Team, Great Learning. “Introduction to VGG16 | What Is VGG16?” *Great Learning Blog: Free Resources What Matters to Shape Your Career!*, 1 Oct. 2021, www.mygreatlearning.com/blog/introduction-to-vgg16. Accessed 18 Dec. 2022.

Team, Keras. “Keras Documentation: VGG16 and VGG19.” *VGG16 and VGG19*, keras.io/api/applications/vgg. Accessed 18 Dec. 2022.

Cassimiro, Gabriel. “Transfer Learning With VGG16 and Keras.” *Medium*, 17 June 2021, towardsdatascience.com/transfer-learning-with-vgg16-and-keras-50ea161580b4. Accessed 20 Dec. 2022.

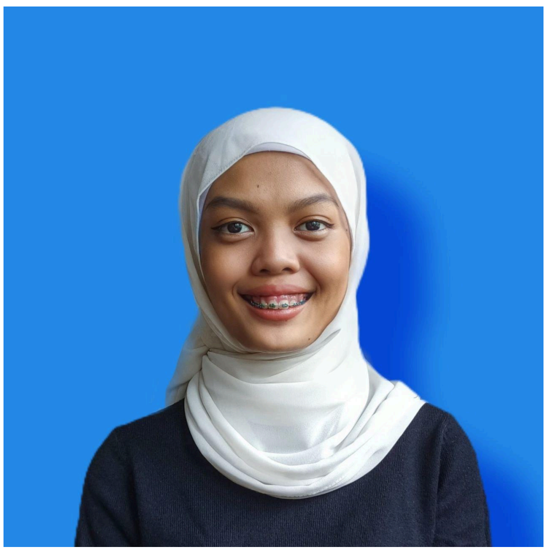
Team, Keras. “Keras Documentation: Losses.” *Losses*, keras.io/api/losses. Accessed 27 Dec. 2022.

“Keras Preprocessing Image Data Generator.” *TensorFlow*, www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator. Accessed 27 Dec. 2022.

Team, Keras. “Keras Documentation: The Sequential Model.” *The Sequential Model*, keras.io/guides/sequential_model. Accessed 28 Dec. 2022.

Maithani, Mohit. “Guide to Tensorflow Keras Optimizers.” *Analytics India Magazine*, 18 Jan. 2021, analyticsindiamag.com/guide-to-tensorflow-keras-optimizers. Accessed 28 Dec. 2022.

Contribution of members

 **Muna Nazihah Binti Mohamad Asman A192817**	***Muna Nazihah*** *is an undergraduate student at the National University of Malaysia (UKM) who is currently pursuing a Bachelor of Computer Science with Honours. Her parts in report-writing includes the Abstract, Knowledge Acquisition, Knowledge Representation, Machine Learning Technique, Inference engine and also Handling uncertainties & conflicting knowledge.*
 **Fahda Nadia Bt Md Shahizan A192732**	***Fahda Nadia*** *is an undergraduate student at the National University of Malaysia (UKM), pursuing a Bachelor of Computer Science degree. She took part in the representation of knowledge with the help of Muna Nazihah, machine learning technique, and machine learning results.*

 **Saimon Mah Wei Yung A197431**	***Saimon Mah*** *is an undergraduate student at the National University of Malaysia (UKM), pursuing a Bachelor of Computer Science degree. Took part in handling the machine learning coding, Knowledge Acquisition, Machine Learning Technique, Interface, Machine Learning Result and giving ideas or guidance in handling other parts for improvement of the project.*
 **Malarkodi A/P Panjawarnam**	***Malar*** *is an undergraduate student at the National University of Malaysia(UKM), pursuing a Bachelor of Computer Science degree. She took part in the report-writing in Introduction, and Inference engine(rules for forward chaining) with the help of Muna Nazihah, machine learning technique (performance measurement), and Conclusion.*