MANIPAL
ACADEMY of HIGHER EDUCATION
(Deemed to be University under Section 3 of the UGC Act, 1956)

MANIPAL SCHOOL OF INFORMATION SCIENCES
(A Constituent Unit of MAHE, Manipal)

Development of a Deep Learning Model for automated detection of Breast Cancer

Reg. Number	Name	Branch
201039004	PRAKASH B R	Embedded Systems
201039020	THEJAS L	Embedded Systems
201039040	SUSHANK SHARMA	Embedded Systems

Under the guidance of

Dr. Keerthana Prasad
Professor,
Manipal School of Information Sciences,
MAHE, MANIPAL

25/01/2021

MANIPAL SCHOOL OF INFORMATION SCIENCES
MANIPAL
(A constituent unit of MAHE, Manipal)

ABSTRACT

The biopsy is one of the most used modalities to identify breast cancer in women, where tissue is removed and studied by the pathologist under the microscope to look for abnormalities in tissue. This technique can be time-consuming, error-prone, and variable results depending on the pathologist's expertise level.

An automated and efficient approach aids in the diagnosis of breast cancer and reduces human effort. This project aims to develop a computerized method for diagnosing breast cancer tumors using histopathological images. We design a residual learning-based convolutional neural network using VGG-16 for breast cancer histopathological image classification in the proposed approach.

The proposed model learns rich and discriminative features from the histopathological images and classifies histopathological images into benign and malignant classes.

Keywords

Breast cancer; Convolutional Neural Network (CNN); Histopathological image; VGG-16; Residual learning.

CONTENTS

LIST OF FIGURES:

LIST OF TABLE:

1. Introduction

Breast cancer is the second leading cause of cancer-related death in women across the globe. International Agency for cancer (IARC) research reported that approximately 8.2 million deaths were caused by cancer in 2012, and about 27 million new cancer cases are expected by 2030. In general, breast cancer is diagnosed by visual inspection of medical images (e.g., breast mammograms, ultrasound, and MRI images). And biopsy. In biopsy examination, tissues are studied under the microscope by the pathologists. Generally, cancerous cells are examined visually, based on shape and size, malignancy degree, tissue distribution, and henceforth. If cancerous cells are present, then the biopsy is the only guaranteed way for diagnosis. The visual examination of the cancerous cell is time-consuming and requires an expert pathologist.

Recent advancements in machine learning and image processing have enabled computer-aided diagnosis (CAD) systems to detect and diagnose breast cancer from the histopathological images faster with very high accuracy. The CAD system analyzes the histopathological images of the sample tissue, finds the histopathological patterns corresponding to the cancerous and non-cancerous condition and classifies the histopathological images into benign and malignant classes. The significant challenges associated with breast cancer histopathological image classification include the inherent complexity in histopathological photos, such as cell overlapping, subtle differences between images, and uneven color distribution.

The objective of this project is to develop an accurate and reliable solution for breast cancer classification. This project shall also help systematically investigate the deep learning-based approaches for the automatic diagnosis of breast cancer.

Key highlights of this project work are as following:

- To develop a deep residual convolutional neural network model for breast cancer diagnosis from the histopathological images using VGG-16.
- Propose a data augmentation technique based on stain normalization, image patches generation, and affine transformation.
- Also, investigate the performances of in-depth features using VGG-16 with various parameters like Batch size, Optimizers, and Learning rate.

2. Material and methods

Deep learning is a data-driven learning approach that learns the features, semantic rules, and meaningful information directly from the data itself. It is beneficial for various tasks, including object detection, voice recognition, signal recognition, visual tracking, and image classification. CNN is a part of a deep learning family. It is widely used in medical image analysis such as MRI image segmentation, brain tumor detection in MRI images, retinal lesion detection in fundus images, and nuclei segmentation in histopathological images. The effectiveness of CNN has motivated us to explore CNN based methodology for breast cancer image classification. We are developing an improved residual learning-based convolutional neural network for breast cancer histopathological image classification in this project work. In the proposed approach, histopathological images are first augmented then the model is trained end-to-end on the expanded dataset in a supervised learning manner. The model learns the discriminative features and mapping rule that maps the input histopathological images to output labels during training.

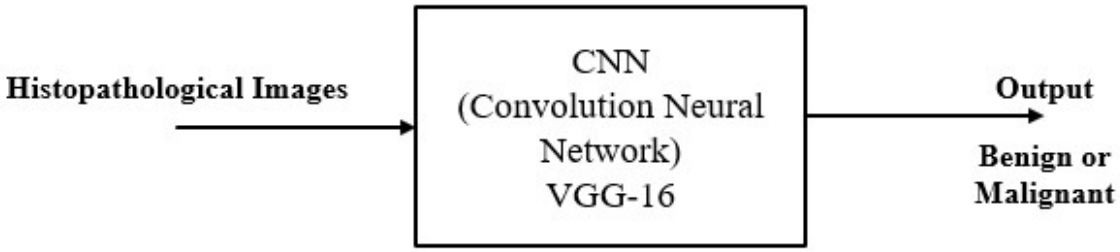


Figure 1: Block diagram of the proposed system

For a given unlabeled test image in the test phase, the trained model produces the probability of cancerous tumor presence in the histopathological image and classifies the image into benign or malignant class.

2.1. BreaKHis dataset

To evaluate the performance of the model, a publicly available BreaKHis dataset is used. This dataset includes 7909 breast cancer histopathological images of 82 patients. Out of 7909 images, 2480 images belong to a benign class, and 5429 images belong to a malignant class. These images are acquired from the biopsy slides of breast tissue in four magnification factors 40X, 100X, 200X, and 400X magnification factors from the BreaKHis dataset. The images in the dataset are the colored (RGB) images of size 700 × 460 pixels.

We consider Fibroadenoma as benign and ductal carcinoma as malignant with 400X magnification factors for this project work. We have a total of 237 Benign and 788 Malignant for the above specification.

2.2. Experiment Setup

The experimental setup used for evaluating the proposed method is having a dataset that has been randomly divided into the training set of 0.8 (80%) and test set in the ratio of 0.1 (10%) for 400X magnification factor. We have further

divided the training set into a rate where the testing set is 0.1 (10%), and validation set 0.1 (10%), then applied a data augmentation approach on a train set.

2.3. Image Normalisation

Normalization is an approach applied during data preparation to change numeric columns' values in a dataset to use a standard scale when the data features have different ranges. Hematoxylin and eosin (H&E) stain is one of the primary tissue stains for histology in this project. This is because H stain makes nuclei easily visible in blue against a cytoplasm's pink background (and other issue regions). For automated image analysis, these H&E stained images need to be normalized. This is because of the significant variation in image colors arising from both sample preparation and imaging conditions.

Figure 2 explains the image normalisation of the proposed system where image 1 denotes the original histopathological image and image 2 is the derived normalised image used for this proposed model.

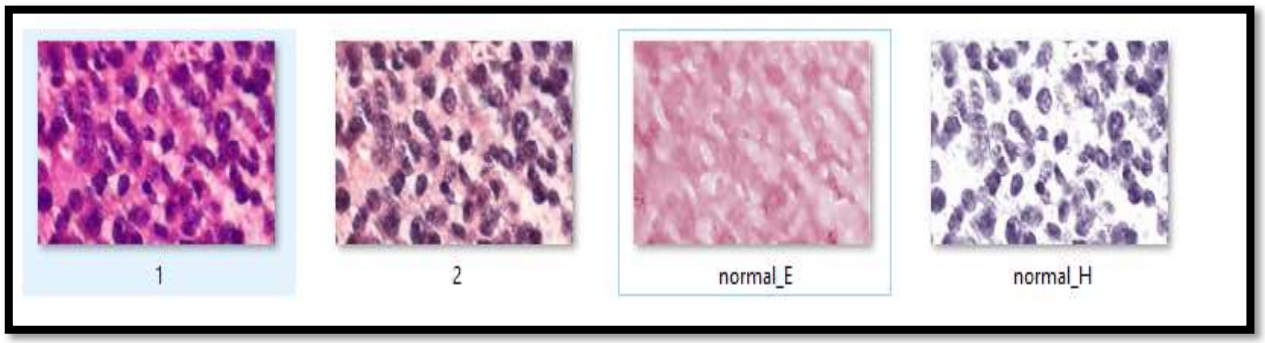


Figure 2: Image normalisation of the proposed system

2.4. Data Augmentation

The data augmentation is an essential step in having enough diverse samples and learning a deep network from the images. The data augmentation process includes random resizing, rotating, cropping, and flipping methods. Out of which, we have

made use of rotation and flip. The proposed model created 6345 images from 1025 using the data augmentation technique.

3. VGG16 | (Convolutional Neural Network model)

VGG-16 is a network with 16 layers proposed by the Visual Geometric Group. These 16 layers contain the trainable parameters, and there are other layers also like the Maxpool layer, but those do not contain any trainable parameters.

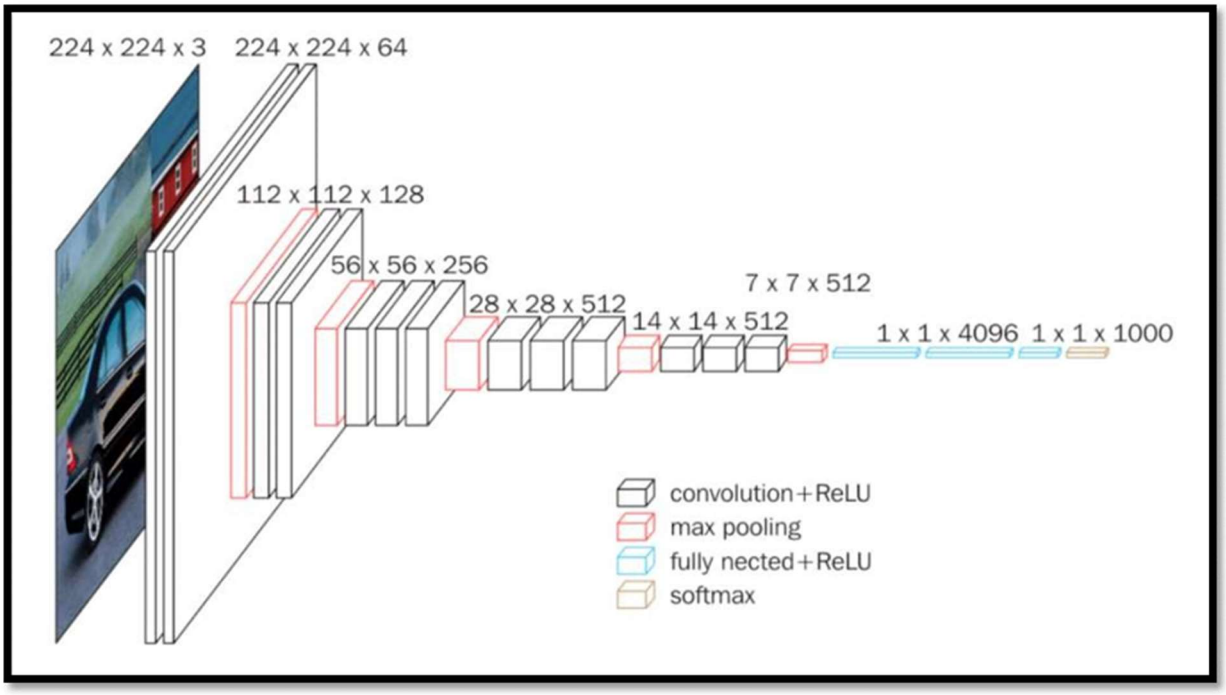


Figure 3: The architecture of the VGG16 CNN Model

1. **Input Layer:** It accepts color images as an input with the size 224 x 224 and 3 channels, i.e., Red, Green, and Blue.
2. **Convolution Layer:** The images pass through a stack of convolution layers where every convolution filter has a minimal receptive field of 3 x 3 and stride of 1. Every convolution kernel uses row and column padding so that the input and output feature maps' size remains the same. In other words, the resolution after the convolution is performed the same.
3. **Max pooling:** It is performed over a max-pool window of size 2 x 2 with stride equals 2, which means the max pool windows are non-overlapping windows.

4. Not every convolution layer is followed by a max pool layer, as at some places, a convolution layer Follows another convolution layer without the max-pool layer in between.
5. The first two fully connected layers have 4096 channels each. The third fully connected layer, which is also the output layer, has 1000 channels, one for each category of images in the ImageNet database.
6. The hidden layers have ReLU as their activation function.

The input to the network is an image of dimensions *(224,224,3).* The first two layers have 64 channels of 3*3 filter size and the same padding. After a max pool layer of stride (2,2), two layers have convolution layers of 256 filter size and filter size (3,3). This is followed by a max-pooling layer of stride (2,2), which is the same as the previous layer. Then there are two convolution layers of filter size (3,3) and 256 filters. After that, there are two sets of 3 convolution layer and a max pool layer. Each has *512* filters of *(3, 3* size with the same padding. This image is then passed to the stack of two convolution layers. In these convolution and max-pooling layers, the filters we use are *3*3* instead of *11*11* in AlexNet and *7*7* in ZF-Net. Some of the layers also use *1*1* pixel to manipulate the number of input channels. There is a padding of *1-pixel* (same padding) done after each convolution layer to prevent the spatial feature of the image.

After the convolution and max-pooling layer stack, we got a *(7,7, 512)* feature map. We flatten this output to make it a *(1,25088)* feature vector. After this, there are *three fully* connected layers. The first layer takes input from the last feature vector and outputs a *(1, 4096)* vector, and the second layer also outputs a vector of size *(1, 4096)*. Still, the third layer output 1000 channels for *1000* classes of ILSVRC challenge, then after the production of the 3rd fully connected layer is passed to softmax layer to normalize the classification vector. After the output of the classification vector, top-5 categories for evaluation. All the hidden layers use

ReLU as its activation function. ReLU is more computationally efficient because it results in faster learning, and it also decreases the likelihood of vanishing gradient problem.

To perform localization, we need to replace the class score by bounding box location candidates. Bounding box location is represented by a 4-D vector (center coordinates, height, width). There are two versions of localization architecture. One is bounding box is shared among different candidates (the output is a *four*-parameter vector), and the other is determining package is class-specific (the result is *4000* parameter vector). The paper experimented with both approaches on VGG -16 (D) architecture. Here we also need to change loss from classification loss to regression loss functions that penalize the deviation of predicted loss from the ground truth.

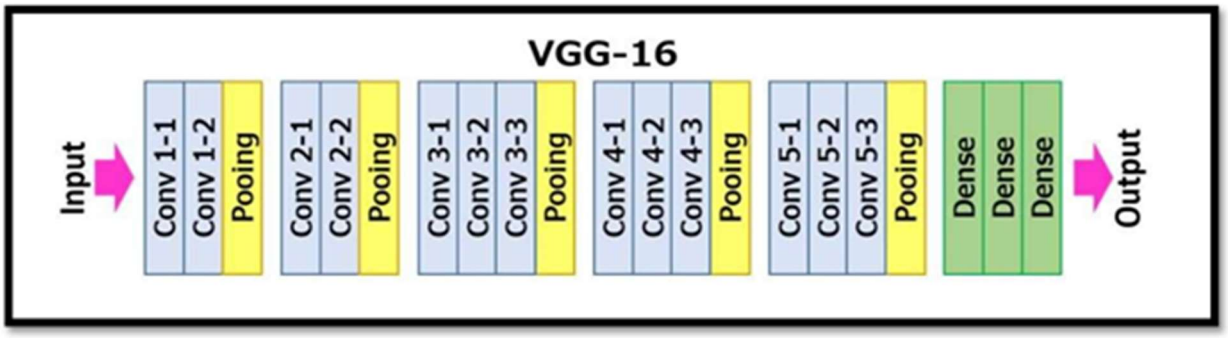


Figure 4: VGG-16 architecture map

The input to the network is an image of dimensions *(224, 224, 3)*. The first two layers have *64* channels of *3*3* filter size and the same padding. After a max pool layer of stride (2, 2), two layers have convolution layers of 256 filter size and filter size *(3, 3)*. Then there are *two* convolution layers of filter size *(3, 3)* and *256* filters. After that, there are *two* sets of *3* convolution layer and a max pool layer. Each has *512* filters of *(3, 3)* size with the same padding. This image is then passed to the stack of two convolution layers. In these convolutions and max-pooling layers, the filters used are of the size *3*3*. There is a padding of *1-pixel* (same

padding) done after each convolution layer to prevent the spatial feature of the image. After the convolution and max-pooling layer stack, a *(7, 7, 512)* feature map is obtained. After this, there are *three fully* connected layers.

After this, there are three fully connected layers, the first layer takes input from the last feature vector and outputs a (1, 4096) vector, the second layer also outputs a vector of size (1, 4096), but the third layer output two channels for two classes.

4. Results

We made use of VGG16 as our primary model. Accordingly, we obtained a statistical result wherein the data augmentation part has been carried out successfully where we used the BreaKHis dataset to train the model.
We developed different models for this project using VGG-16 by varying the following parameters: Learning rate, Epoch, Mini-batch size, and optimizers.

Optimizers: During the training process, we tweak and change the model's parameters (weights) to minimize the loss function and make our predictions correct and optimized. They tie together the loss function and model parameters by updating the model according to its output.
Learning Rate: In machine learning and statistics, the learning rate is a tuning parameter in an optimization algorithm that determines the step size at each iteration while moving toward a minimum of a loss function.
Talking about the training and testing accuracy, this model helped us obtain nearly 97% of training accuracy and 87.5% testing accuracy. The proof of the same has been represented below:
Epoch: An Epoch can be described as one complete cycle through the entire training dataset. It indicates the number of passes that the machine learning algorithm has completed during that training.

The table below represents the results obtained for different models with varying parameters:

Table 1: Different models with varying parameters

Model Name	Train	Vaidation	Test	Optimizer	Learning Rate	Epochs	Training Accuracy	Validation Accuracy	Testing Accuracy	Normalised
VGG16-Final1	614	307	104	RMS	2.00E-05	50	0.775	0.7773	0.7692	NO
VGG16-Final2	614	307	104	RMS	2.00E-05	20	0.971	0.9062	0.9038	NO
VGG16-Final3	614	307	104	ADAM	0.01	20	0.9875	0.9102	0.8942	NO
VGG16-Final4	4124	1586	635	ADAM	0.01	10	0.9469	0.9102	0.9333	NO
VGG16-Final5	614	307	104	ADAM	0.01	10	0.9968	0.8477	0.9333	YES
VGG16-Final6	4124	1586	635	ADAM	0.01	10	0.8906	0.8984	0.8844	YES
VGG16-Final7	4124	1586	635	ADAM	0.001	10	0.9094	0.8633	0.8799	YES
VGG16-Final8	4124	1586	635	ADAM	0.00001	10	0.7281	0.7933	0.7844	YES
VGG16-Final_Jan20	4124	1586	635	ADAM	0.001	30	0.9563	0.9492	0.9377	YES

a. Graphical representation of different models for *training and validation accuracy* and the *training and validation loss* :

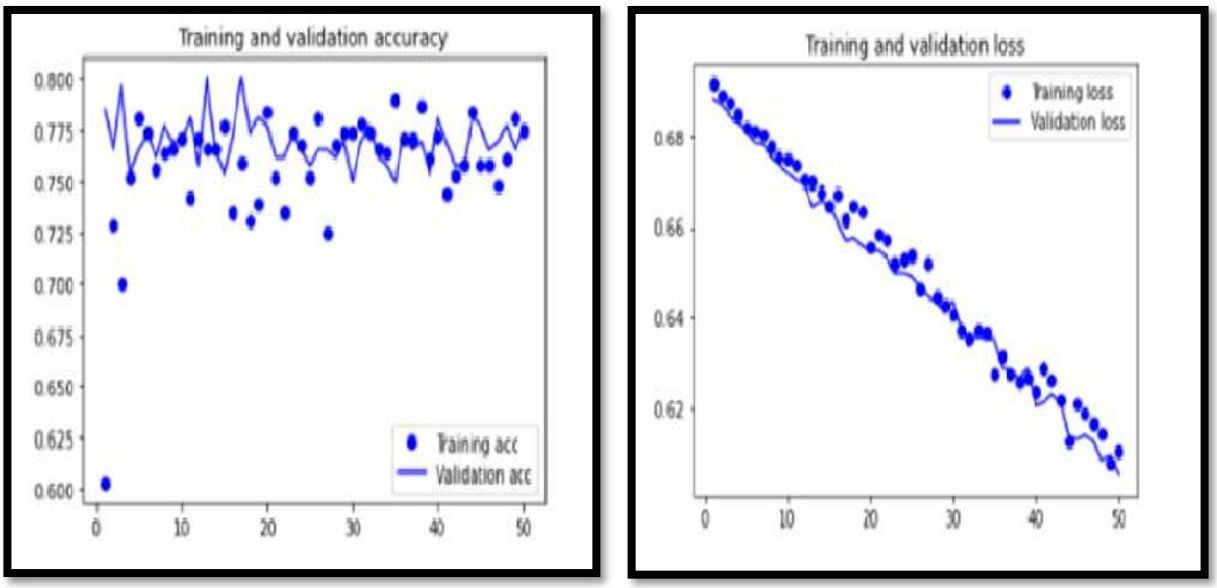


Figure 5: Graphical representation of VGG16-Final 1, training and validation accuracy, and the training and validation loss.

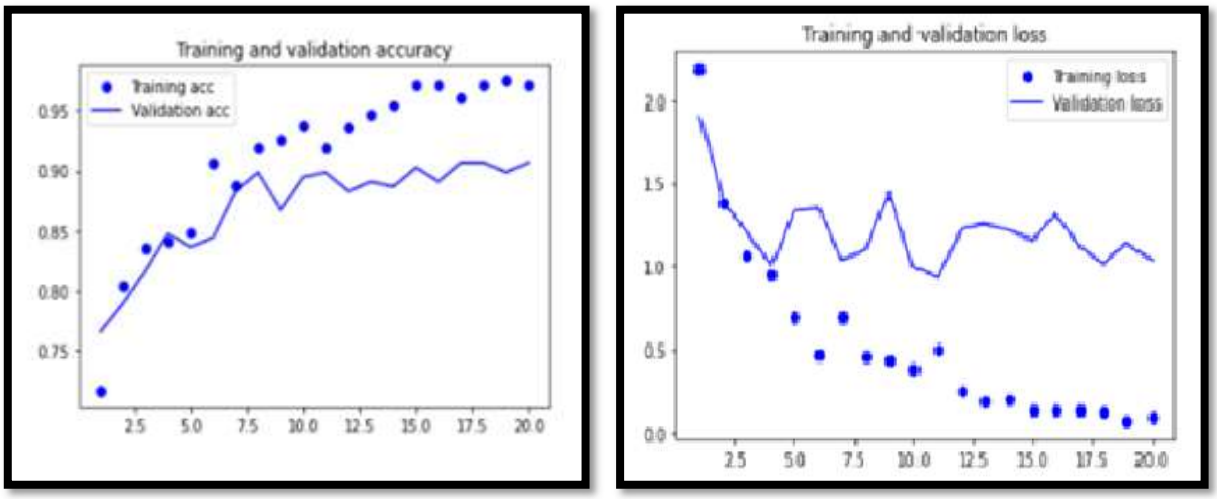


Figure 6: Graphical representation of VGG16-Final 2, training and validation accuracy, and the training and validation loss.

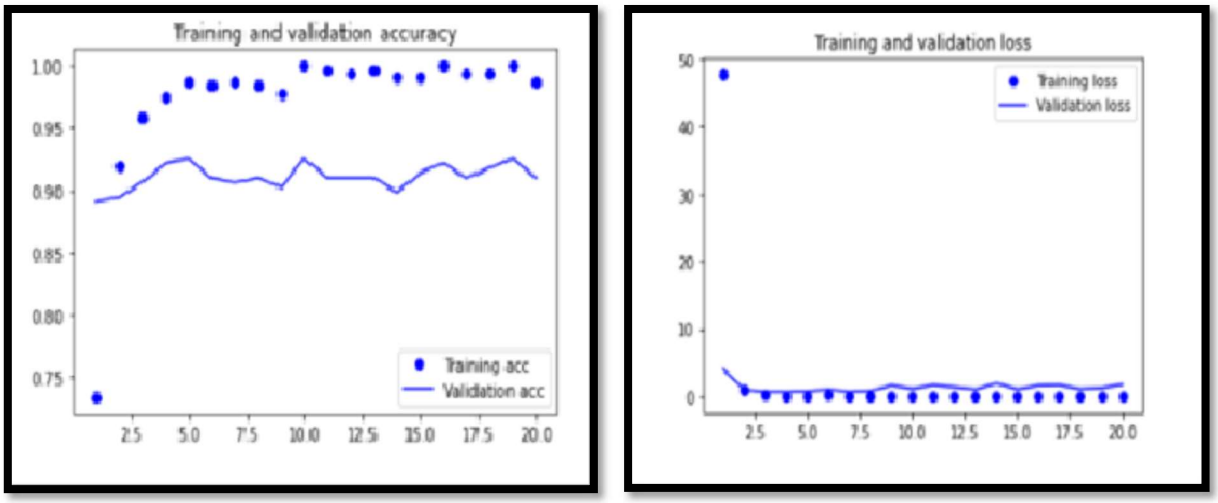


Figure 7: Graphical representation of VGG16-Final 3, training and validation accuracy, and the training and validation loss.

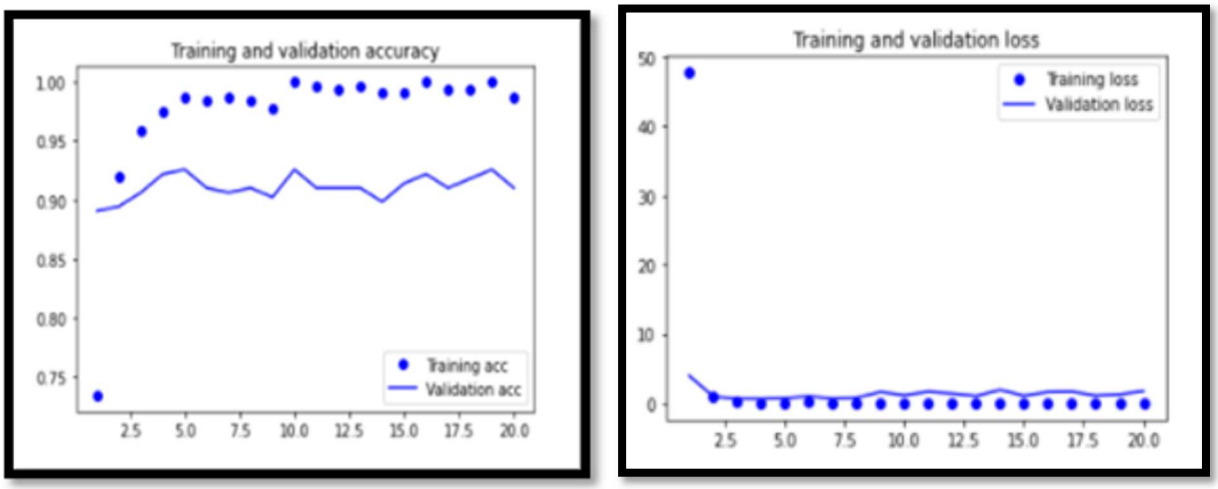


Figure 8: Graphical representation of VGG16-Final 4, training and validation accuracy,

and the training and validation loss.

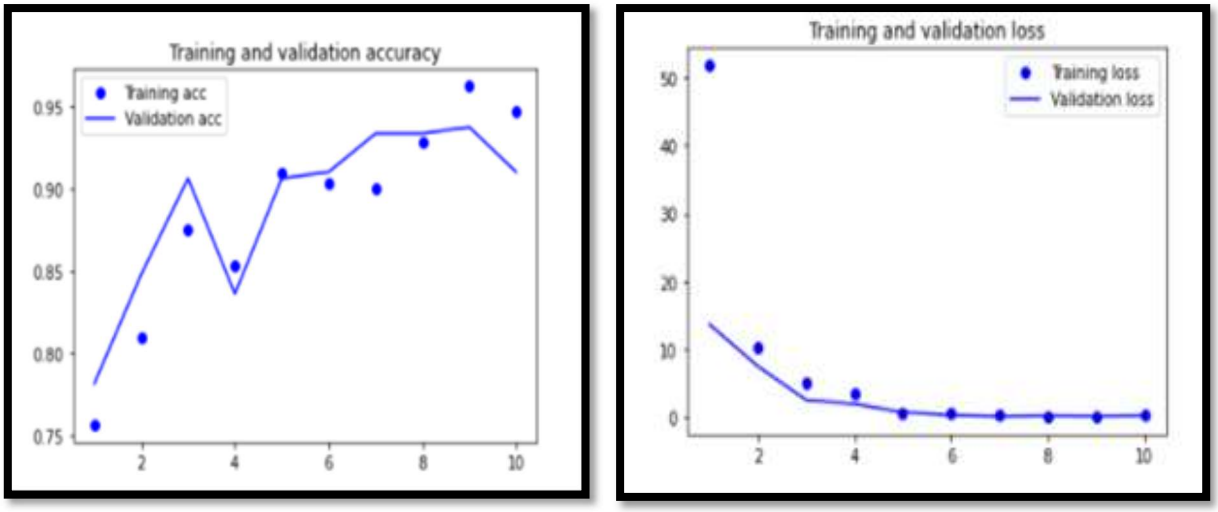


Figure 9: Graphical representation of VGG16-Final 5, training and validation accuracy, and the training and validation loss.

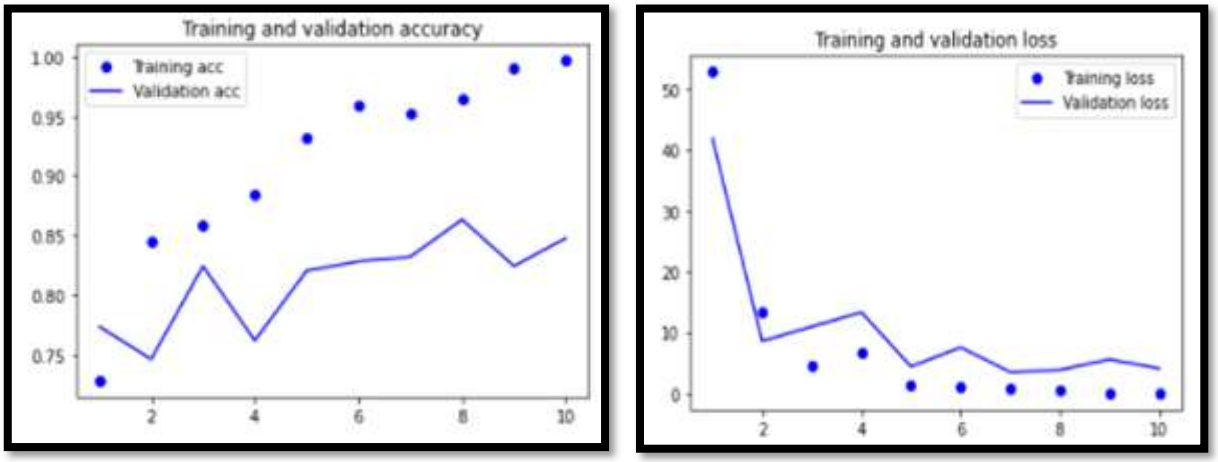


Figure 10: Graphical representation of VGG16-Final 6, training and validation accuracy, and the training and validation loss

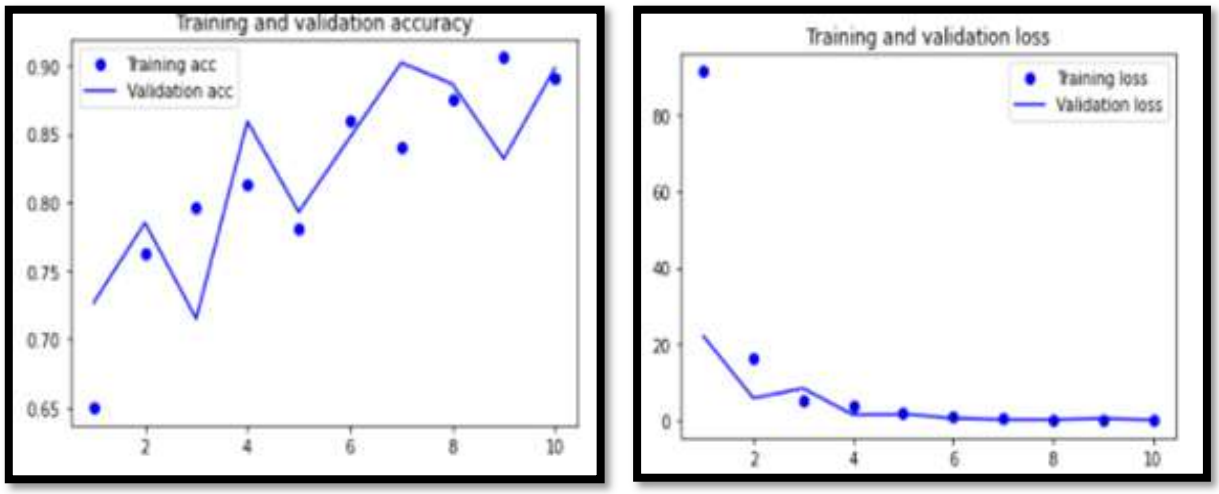


Figure 11: Graphical representation of VGG16-Final 7, training and validation accuracy, and the training and validation loss

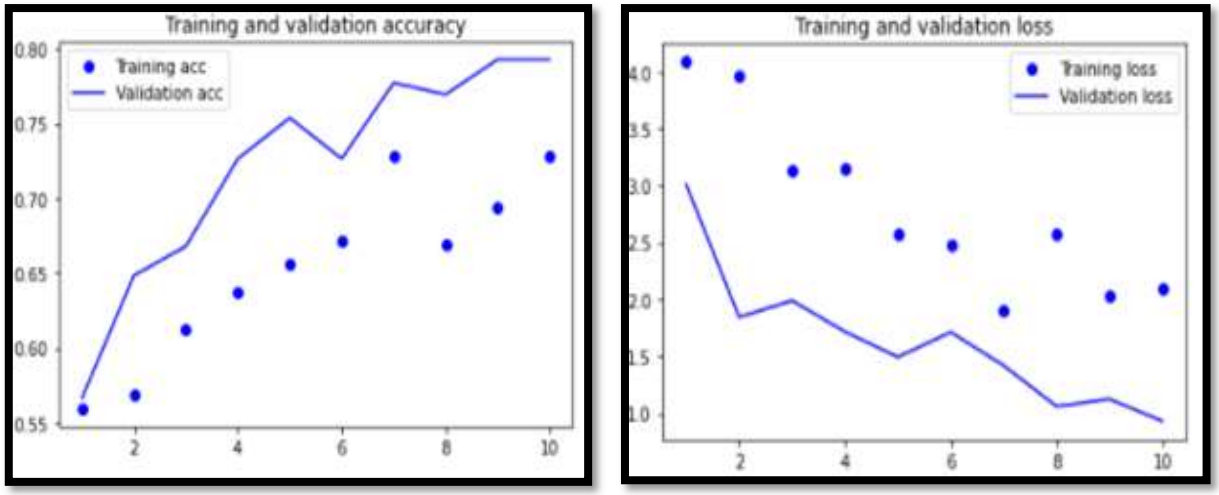


Figure 12: Graphical representation of VGG16-Final 8, training and validation accuracy, and the training and validation loss

The below model is the most consistent among the different models developed:

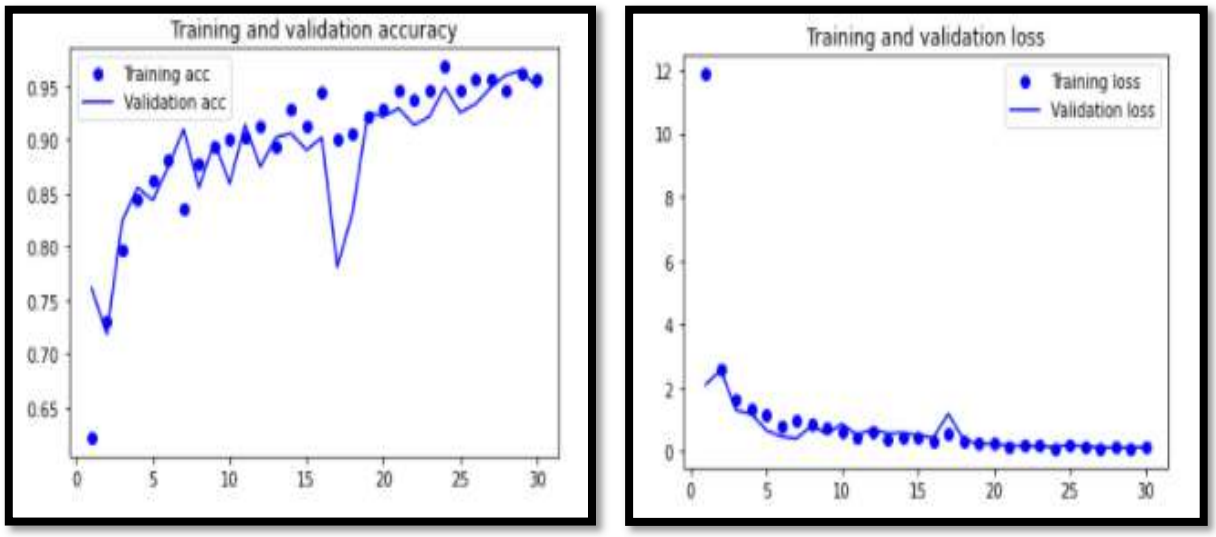


Figure 13: Graphical representation of VGG16-Final_Jan20, training and validation accuracy, and the training and validation loss

Model Name	Train	Vaidation	Test	Optimizer	Learning Rate	Epochs	Training Accuracy	Validation Accuracy	Testing Accuracy	Normalised
VGG16-Final_Jan20	4124	1586	635	ADAM	0.001	30	0.9563	0.9492	0.9377	YES

The test accuracy of the histopathological images corresponds to **93.77%**:

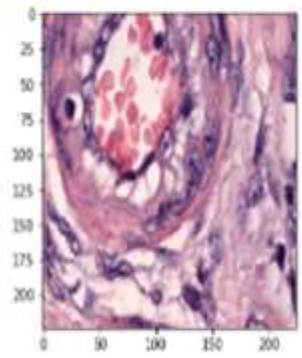
b. The result calculation after the training of the histopathological images and displaying the desired result:

```
img = image.load_img(path,target_size=(224,224))
img = np.asarray(img)
plt.imshow(img)
img = np.expand_dims(img, axis=0)
from keras.models import load_model
saved_model = load_model("vgg16_final_18.h5")

x=saved_model.predict(img)
print(x)
if(x>0.5):
    x=1;
else:
    x=0;
print(x)

if x>0:
  print( "Malignant")
else:
  print("Benign")
```

```
WARNING:tensorflow:11 out of the last 11 calls to <function Model.make_predict_function.<locals>.predict_function at 0x000001B3
10704040> triggered tf.function retracing. Tracing is expensive and the excessive number of tracings could be due to (1) creati
ng @tf.function repeatedly in a loop, (2) passing tensors with different shapes, (3) passing Python objects instead of tensors.
For (1), please define your @tf.function outside of the loop. For (2), @tf.function has experimental_relax_shapes=True option t
hat relaxes argument shapes that can avoid unnecessary retracing. For (3), please refer to https://www.tensorflow.org/tutorial
s/customization/performance#python_or_tensor_args and https://www.tensorflow.org/api_docs/python/tf/function for  more details.
[[0.99998975]]
1
Malignant
```



5. Conclusion

In conclusion, the current study focused on improving the classification accuracy of the BreakHis data. It was comprised of the histopathological images separated into two different classes as benign or malignant. Our study demonstrates that deep learning models trained in an end-to-end fashion can be even more accurate and potentially readily transferable across diverse histopathological platforms. Deep learning methods can significantly improve breast cancer detection accuracy on screening histopathological images as the available training datasets and computational resources expand.

6. References

1] M. To˘gaçar, K.B. Özkurt, B. Ergen, et al., BreastNet: A novel convolutional neural network model through histopathological images for the diagnosis of breast cancer, Physica A (2019).

2] Kadir Can Burçak Ömer Kaan Baykan, Harun Uğuz: A new deep convolutional neural network model for classifying breast cancer histopathological images and the hyperparameter optimization of the proposed model

3] Mahesh Gour, Sweta Jain, T. Sunil Kumar: Residual learning-based CNN for breast cancer histopathological image classification.