University of Hertfordshire

Department of Physics, Astronomy and Mathematics

MSc Data Science

Project report: 7PAM2002 – Data Science Project

Breast Cancer Detection Using Transfer Learning

Saad Jalil Ahmed Shaikh

(20070328)

Supervisor: Ralf Napiwotzki

MSc Final Project Declaration

This report is submitted in partial fulfilment of the requirement for the degree of Master of Data science at the University of Hertfordshire (UH).

It is my work except, where indicated in the report.

Abstract

Research on restorative pictures heavily relies on transfer learning, however obtaining high-quality training datasets for the machine learning techniques might be difficult. Despite some study attempts, there haven't been many evaluation papers showing the uses of transfer learning in the restorative analysis of the image until this point. Furthermore, no audits have yet been conducted for the use of transfer learning to ultrasound breast imaging.

To summarise current approaches and highlight their benefits and drawbacks, this study investigates earlier efforts that centred on employing transfer learning to analyse breast cancer from ultrasound pictures. Future research areas for using transfer learning on ultrasonic imaging to detect and diagnose breast cancer are also emphasised in this research.

MobleNetV2 and InceptionV3 are deep learning approaches used to detect breast cancer. The primary difference between the two is that MobilenetV2 utilises depth-wise separable convolution whereas InceptionV3 uses standard convolution. MobileNetV2 hence has fewer parameters than InceptionV3. There is a little performance reduction as a result, though. We evaluated both models' accuracy and contrasted the findings.

Table of Contents

Chapter 1: Introduction

Currently, Breast cancer is the leading cause of death in women, affecting 12.5% of all females across all socioeconomic groups (Mutar, M.T.;Goyani, M.S.;Had, A.M.;Mahmood, A.S., 2019). The earliest detection of breast cancer is pivotal since it can reduce mortality rates by up to 40% (Coleman, 2017), (Smith & Webb, 2010). In recent years, ultrasound imaging technique is now an approved imaging technique for detecting breast cancer, particularly in young aged women who have dense breasts (Gilbert, F.J.; Pinker-Domenig, K., 2019). Ultrasonic (US) imaging is typically used so that tissue characteristics may be successfully obtained (Jonathan,L Jesneck.; Joseph, Y Lo.; Jay A Baker., 2007) (Hindi A,; Peterson C,' Barr RG., 2013). Studies have shown that using a variety of modalities, including US imaging, decreased the incorrect negative acceptance rate in other breast symptomatic techniques, like biopsy and mammography (MG) (Coleman, 2017). Using, ultrasonic imaging techniques to diagnose breast cancer can increase tumour findings by up to 17% (Myra K Feldman, Sanjeev Katyal, Margaret S Blackwood., 2009). Additionally, it is possible to reduction in the number of unwanted biopsies by around 40%, thus, would reduce the number of medications consumed (Jonathan,L Jesneck.; Joseph, Y Lo.; Jay A Baker., 2007). Another benefit of ultrasonic imaging is the use of non-ionizing radiation, which has no negative effects on health and just requires basic equipment (Hindi A,; Peterson C,' Barr RG., 2013). As a result, ultrasound scanners are more affordable and flexible than mammography machines (Jonathan,L Jesneck.; Joseph, Y Lo.; Jay A Baker., 2007) (Zhou, 2013). However, mammography and histological analyses are not stand-alone techniques to detect breast cancer (Myra K Feldman, Sanjeev Katyal, Margaret S Blackwood., 2009) (Hindi A,; Peterson C,' Barr RG., 2013), also they are integrated with ultrasound frameworks to demonstrate the results (Zhou, 2013). Many studies have made use of modern technology to increase ultrasonic imaging's demonstration power (Shengfeng L a, Wang Y a, Xin Y b, Baiying L a,Li Liu a, Shawn X L a, Dong N a, Tianfu W., 2019).

Machine learning, counting false positive rates, failing to recognise changes brought on by illness, diminished appropriateness for treatment checking, and subjectivity have all been found to be solutions to many of the problems with ultrasound's categorization, discovery, and division of breast cancer (Qinghua Huang, Fan Zhang, Xuelong Li. , 2018) (Sloun R J v, Cohen R, Yonina C

E., 2019). Nevertheless, many machine learning techniques perform effectively when a certain introduction is true when the training and the test sets of data originate from the identical highlight space and are of the same dissemination (Sinno J P; Q Yang., 2010). When the dispersion changes, the majority of the numerical values inside the models must be created from the scratch using freshly gathered preparation information (L J Brattain, B A Telfer, M Dhyani, J R Grajo, A E Samir , 2018) (Sinno J P; Q Yang., 2010). For therapeutic applications, such as breast ultrasound imaging technique, it's difficult to push for the core pre-planning information and develop models (Vahab Khoshdel, Ahmed Ashraf, Joe LoVetri, 2019). Therefore, it's advisable to reduce the effort and the requirement needed to push the preparation of information (Sinno J P; Q Yang., 2010) (Vahab Khoshdel, Ahmed Ashraf, Joe LoVetri, 2019). In such cases, information sharing from one task to the other task would be perfect (Oscar Day, Taghi M. Khoshgoftaar , 2017). It is possible to use a method that has already successfully been applied in a different space as the learning target with transfer learning (Karl Weiss, Taghi M. Khoshgoftaar, DingDing Wang , 2016). As a result, gathering and arranging material for learning takes less effort and time (Qinghua Huang, Fan Zhang, Xuelong Li. , 2018) (Karl Weiss, Taghi M. Khoshgoftaar, DingDing Wang , 2016).

Transfer learning is a theory built on the notion that earlier includes learned knowledge and is connected in exceptional circumstances to address current issues more effectively and rapidly (Brownlee, 2017). Transfer learning at that point necessitates good machine learning methods that archive and utilise previously obtained data (G I Parisi, R Kemker, J L Part, C Kanan, S Wermter, 20169) (Zhiyuan Chen, Bing Liu, 2016). Following the development of a few convolutional neural network models, to address visual classification assignments in common pictures created on common picture databases like ImageNet, exchange learning has most recently been linked to breast cancer imaging (Md Z Alom, T M. Taha, C Yakopcic, S Westberg, P Sidike, M S Nasrin, B C Esesn, A A S. Awwal, V K. Asari, 2018).

The effectiveness of using characteristics from the pre-trained deep CNNs to identify breast cancer using computer-aided determination (CADx) was evaluated in Hyunh et al 2016 research on the principal use of exchange learning in breast cancer imaging (B Huynh,K Drukker,M Giger, 2016). After that, Byra et al. published a study in which they suggested using ultrasonography and neural exchange learning to categorise breast damage (M Byra, M Galperin, H O Fournier, L Olson, M

O'Boyle, C Comstock, M Andre, 2019). According to research published in, Yap et al. (M H Yap, G Pons, J Marti, S Ganau, M Sentis, R Zwiggelaar, A K Davison, R Marti, M H Yap, G Pons, J Marti, S Ganau, M Sentis, R Zwiggelaar, A K Davison, R Marti, 2018), which advocated the use of sophisticated neural learning techniques for the detection of breast cancer. They looked at three different approaches using a pre-trained fully convolutional network, AlexNet, that includes a patch-based LeNet strategy, a U-Net demonstration, and an exchange learning technique. Following the distribution of these factors, a few distributions about the use of exchange learning for breast ultrasound imaging have been created (M Byra, M Galperin, H O Fournier, L Olson, M O'Boyle, C Comstock, M Andre, 2019) (Hadad, O.; Bakalo, R.; Ben-Ari, R.; Hashoul, S.; Amit, G., 2017). This study evaluates articles on breast cancer imaging that make use of exchange learning to describe contemporary methodologies and identify their strong points and weaknesses. Additionally, it provides immediate future avenues for learning and sharing in ultrasound breast cancer imaging. The audit will play a key role in assisting researchers in identifying impending strategies that will advance as well as regions that appear to benefit from deeper investigation into exchange learning-based ultrasound breast imaging.

The complete report is organized in the following chapters as follows:

Chapter 2: Aim and Objectives

Chapter 3: Background

Chapter 4: Problem Statement

Chapter 5: Methodology Used

Chapter 6: Applications of deep learning for breast cancer detection

Chapter 7: Model Development

Chapter 8: Result Analysis

Chapter 9: Conclusion

Chapter 2: Aim and Objectives

By lowering the annual global breast cancer mortality rate by 2.5%, the WHO Worldwide Breast Cancer Activity (GBCI) hopes to save 2.5 million breast cancer deaths between the years 2020 and 2040. Every breast cancer therapy should aim to remove as much cancer from your body as possible to prevent the disease from recurring.

We particularly looked for to find the following:

1. **Collect** ultrasound images for breast cancer.
2. **Apply** deep learning algorithms – MobileNetV2 and InceptionV3 on it.
3. **Investigate** the results.
4. **Examine** how deep learning algorithms are best for detecting and classifying breast cancer.
5. **Evaluate** the key events which are responsible for breast cancer.
6. **Develop** a model which can filter the images based on the type of breast cancer.

Chapter 3: Background

3.1 Neural Networks

An oriented graph is what a neural network is. A machine learning technique called a neural network is based on the design of a human neuron. There are millions of neurons in the human brain. Electrical and chemical impulses are sent and processed by it. These neurons are linked together by synapses, a unique structure. Neurons can transmit messages through synapses. Neural networks emerge from enormous numbers of simulated neurons. Neural networks are not only useful for categorization. Regression of continuous target attributes is also applicable (Sheetal, 2017).

Node layers, which include an input layer, a hidden layer, and an output layer, make up neural networks. Each node, or artificial neuron, is connected to each other and has a weight and threshold that go along with it. Any node whose output exceeds the defined threshold value is activated and begins providing data to the network's uppermost layer. Otherwise, no data is transmitted to the network's next tier (IBM Cloud, 2020).

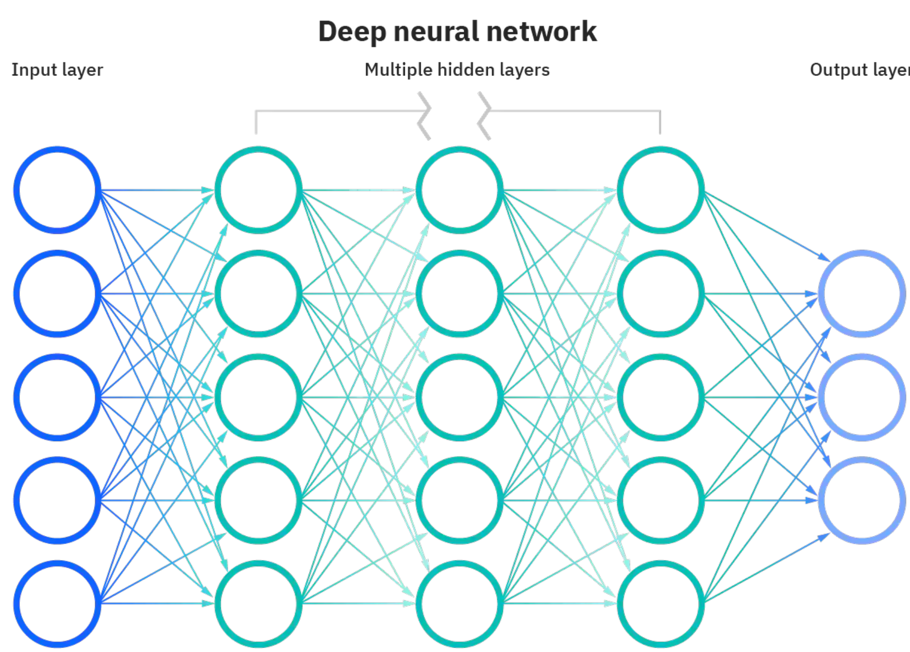

Figure 1 Neural Network

3.2 Convolution Neural Networks (CNN)

A particular type of neural network with many layers is called a convolutional neural network (CNN). It analyses data that is organised in a grid-like fashion and then extracts key features. Using CNNs has the enormous benefit of reducing the amount of image pre-processing required (Milecia, 2021).

Similar to conventional (Artificial Neural Networks) ANNs, convolutional neural networks (CNNs) are made up of neurons that learn to optimise themselves. The fundamental building block of countless ANNs, each neuron will continue to take in input and carry out an action (such as a scalar product followed by a non-linear function). The entire network will still express a single perceptual scoring function from the input raw picture vectors to the class score at the end (the weight). The last layer will include loss functions related to the classes, and all of the standard techniques created for conventional ANNs are still applicable. The main significant distinction between CNNs and conventional ANNs is that CNNs are typically utilised in the field of pattern detection within images. (Keiron, O,'Shea.;Ryan,Nash., 2015)

3.2.1 Convolutional Neural Network Architecture

The three layers of a CNN are typically convolutional, pooling, and fully connected.

Convolution Layer.

The foundational component of the CNN is the convolution layer. It carries the majority of the computational load on the network. This layer creates a dot product between two matrices, one of which is the kernel—a collection of learnable parameters—and the other of which is the constrained area of the receptive field. Compared to a picture, the kernel is smaller in space but deeper (Mayank, 2020).

Pooling Layer

The primary goal of the pooling layer is to gradually reduce the input image's spatial size, which lowers the number of computations required by the network. By lowering the size, pooling conducts downsampling and transmits only the crucial data to CNN's subsequent layers. Convolutional layers are followed by pooling layers. Max pooling, Average pooling, and L2-norm pooling are the three most popular pooling methods (Raghuveer., 2019)

Fully connected Layer.

The final layer of the convolutional neural network is the Fully Connected Layer, sometimes referred to as the Hidden Layer.

This layer is a combination of the Affine function and Non-Linear function.

Affine Function y = Wx + b

Non-Linear Function Sigmoid, TanH and ReLu

Flatten Layer, a one-dimensional layer, provides input to the fully connected layer (1D Layer). The Affine function receives the data from the Flatten Layer first, followed by the Non-Linear function. One FC (Fully Connected) or one Hidden Layer is the combination of one Affine function and one Non-Linear function (IndianTechWarrior., n.d.).

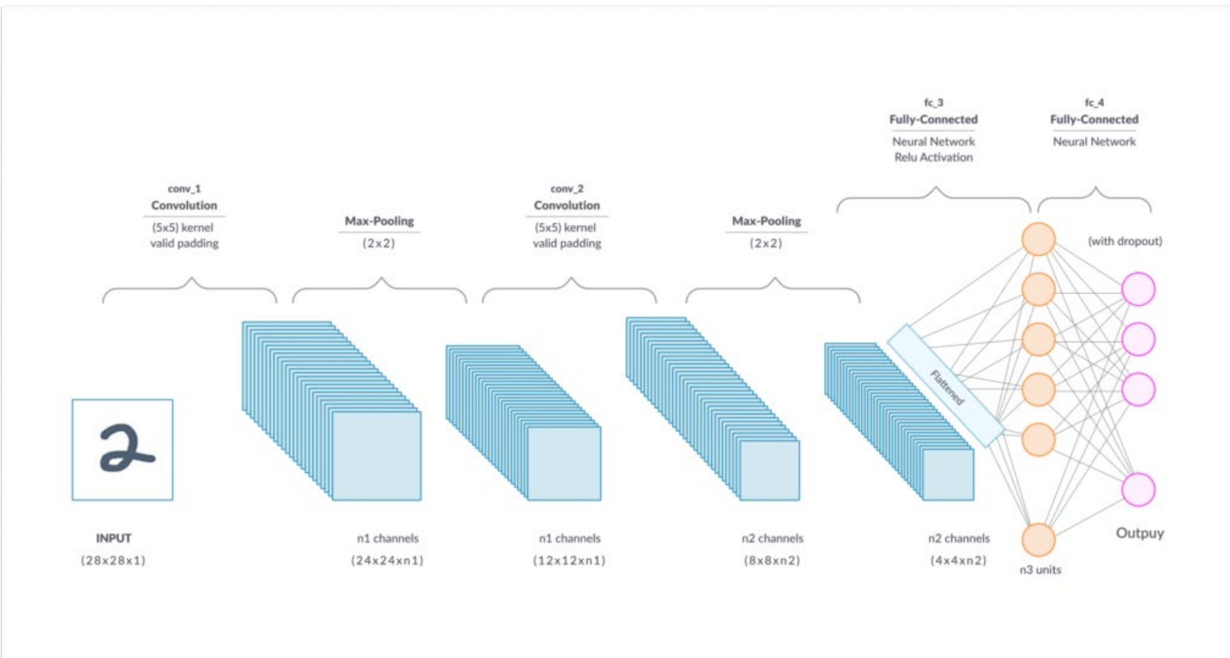

Figure 2 Architecture of CNN

3.3 Transfer Learning

Transfer learning allows you to use feature representations from a model that has already been trained rather than having to create a new model from start. These models can be included in the process of training a new model or utilised directly to make predictions on new problems. Reduced training time and generalisation errors are produced by incorporating previously trained models into new models. (Mwiti, 2022)

Transfer learning is a method widely used for developing models for machine learning without worrying about the quantity of information that is currently accessible (Dogan, 2018). Transfer learning might help with this problem because it could need some knowledge and computer skills to prepare a thorough demonstration. Through exchange learning, a pre-created demonstration may frequently be applied to several issues (B Chu, V Madhavan, O Beijbom, J Hoffman, T Darrell , 2016). For instance, a system that has been developed to do one thing, such as recognise different types of cells, may be fine-tuned to do another thing, such as categorising tumours. When completing computer vision-related homework, transfer learning may be a useful tactic. According to theories of transfer learning (B Chu, V Madhavan, O Beijbom, J Hoffman, T Darrell , 2016) (J Yosinski, J Clune, Y Bengio, H Lipson, 2014) skills learned from enormous image databases like ImageNet are incredibly transferrable to a different number of picture recognition tasks.

There are two ways to transfer data from one model to another. The approach that is most frequently used is to replace the last layer of the already produced model with one that has been randomly initialised (M Huh, P Agrawal, A A. Efros, 2016). After that, all other parameters are cleaned away without being updated while the top-layer parameters, so to speak, are readied for the modern task. This approach may be considered as the application of the swapped model as a highlight extractor since the settled parcel serves as one. The top layer continues to function as a regular, fully linked neural organising layer in the interim, making no unusual assumptions about the input (M Huh, P Agrawal, A A. Efros, 2016) (Z Li, D Hoiem., 2017).

When there is not much data available, this method of exchange learning could be the only reason for developing a model without going into overfitting (Alexandre Gonfalonieri, 2019). This occurs frequently because using fewer parameters reduces the likelihood of overfitting. When current information is available for planning, which is unusual in therapeutic situations, it is also possible to unfreeze traded parameters and arrange the entire network (M Huh, P Agrawal, A A. Efros,

2016) (TensorFlow, n.d.). In this case, the distinctive values of the parameters are effectively switched (Carremans, 2018). Utilizing a pre-trained demonstration can provide the demonstration with a wonderful start and speed up merging and fine-tuning rather than randomly initialising the weights.

To retain the pretraining initialization, scaling back the learning rate by an order of magnitude is a standard procedure (Carremans, 2018) (Csefalvay, 2019). It is standard practice to create fair randomly started layers until they merge using initially frozen parameters (fchollet, 2020) (Koehrsen, 2019), unfreezing all parameters, and then fine-tuning the overall arrangement. This prevents the exchanged parameters from being modified too soon. Exchange learning is especially useful when there is a little quantity of information for one job and a huge volume of information for another assignment that is equivalent, or when a model has already been created on such material (N Best, J Ott, E J. Linstead , 2020)]. Even though the destinations are disconnected and there is sufficient information to prepare a demonstration from scratch, using a pre-trained model to initialize the parameters is still more ideal than irregular initialization (K He, R Girshick, P Dollar, 2019).

Shorter preparation periods, improved neural network performance, and fewer input requirements are some of transfer learning's main advantages (B Neyshabur, H Sedghi, C Zhang, 2021). The early layer parameters in neural systems that have been previously trained on a large number of images are comparable regardless of the specific work that a neural arrangement has been performed on (Karl Weiss, Taghi M. Khoshgoftaar, DingDing Wang , 2016) (B Neyshabur, H Sedghi, C Zhang, 2020). The most accurate CNN layers for occurrence frequently learn edges, surfaces, and designs (B Chu, V Madhavan, O Beijbom, J Hoffman, T Darrell , 2016), and these layers catch the elements that are generally helpful for comprehending distinctive images [(B Neyshabur, H Sedghi, C Zhang, 2021). It is possible to think about highlights that recognise edges, corners, forms, surfaces, and other features as generic highlight extractors and use them in a variety of contexts (B Chu, V Madhavan, O Beijbom, J Hoffman, T Darrell , 2016) (J Yosinski, J Clune, Y Bengio, H Lipson, 2014).

As we get closer to the harvest, the layers tend to memorise increasingly precise properties (L Liu, J Chen, P Fieguth, G Zhao, R Chellappa, M Pietikäinen , 2019) (Y Yan, W Ren, X Cao, 2019). The last layer of a structure that has been prepared for categorization may be particularly intriguing for that classification job (A Carkacioglu , F Y vural, 2003). If the model had been trained to display malignancies, one unit would have reacted as it did to a particular tumour image (B Huynh,K Drukker,M Giger, 2016) (M H Yap, G Pons, J Marti, S Ganau, M Sentis, R Zwiggelaar, A K Davison, R Marti, M H Yap, G Pons, J Marti, S Ganau, M Sentis, R Zwiggelaar, A K Davison, R Marti, 2018). When all the layers except the topmost layer are switched, transfer learning is most prevalent (Brownlee, 2017) (Silver, D L. Y Qiang,Li Lianghao, 2013). A pre-trained demonstration may commonly have its first n layers switched to a goal configuration, with the subsequent layers being randomly initialised (S Imai, S Kawai, H Nobuhara , 2020).

Exchanges may also be made to the final layers (J Yosinski, J Clune, Y Bengio, H Lipson, 2014). To create a tumour acknowledgement display that accepts inputs from both colourful and grayscale photos, consider a tumour recognition proof model that was built on grayscale photographs. It may be more efficient to replace the subsequent layers and retrain the earlier ones because there are not enough informational points to fully prepare an unused display (Z Q Zhao, P Zheng, S T Xu, X Wu, 2019) (Transfer Learning (C3W2L07), 2017). Exchange learning is therefore extremely helpful when a sizable pre-existing knowledge base may be transferred to a target problem and have insufficient knowledge for a new domain that has to be controlled by a neural network (B Neyshabur, H Sedghi, C Zhang, 2020) (Transfer Learning (C3W2L07), 2017).

Because information annotators frequently supply enormous named datasets, they are typically very useful for medicinal image management (M Byra, M Galperin, H O Fournier, L Olson, M O'Boyle, C Comstock, M Andre, 2019) (Hadad, O.; Bakalo, R.; Ben-Ari, R.; Hashoul, S.; Amit, G., 2017). Additionally, preparation time is constrained since, in the event of difficult goal tasks, it might reduce the time needed to completely build a contemporary deep neural network (L Liu, J Chen, P Fieguth, G Zhao, R Chellappa, M Pietikäinen , 2019) (A Carkacioglu , F Y vural, 2003). Exchange learning has allowed analysts working in the data-limited field of restorative imaging to comprehend the problem of limited test datasets and advance execution (Sinno J P; Q Yang., 2010). Exchange learning may be categorised as cross-domain or cross-modal depending on whether the source and target information come from the same area (J Zhang, W Li, P Ogunbona, D Xu, 2019) (D Nguyen, S Sridharan, D T Nguyen, S Denman, D Dean, C Fookes, 2020).

Cross-domain transfer learning may be a noticeable technique for finishing a variety of tasks in restorative ultrasound image research (Shengfeng L a, Wang Y a, Xin Y b, Baiying L a,Li Liu a, Shawn X L a, Dong N a, Tianfu W., 2019)]. Large test datasets are often used in machine learning to pre-train models, and a wealth of training data ensures high efficiency; however, this is often not the case in the field of restorative imaging (J Schmidt, M R. G. Marques, S Botti, M A. L. Marques, 2019). When compared to exchange learning from a neural network that has been pre-trained with substantial preparation tests in another area, such as the common image database of ImageNet, domain-specific models created from scratch can perform significantly better in the event of small preparation tests. (R N D'souza, P-Y Huang, F-C Yeh, 2020) (Amit, G ; Ben-Ari, R ; Hadad, O ; Monovich, E ; Granot, N ; Hashoul, S, 2017).

One of the causes of this is frequently the pre-trained case's difficult gauging from the unrefined picture to the join vectors necessary for a certain activity, such as categorization in the context of medicine., which requires a noteworthy preparing test for made stride hypothesis (I R i Haque, J Neubert, 2020) (Amit, G ; Ben-Ari, R ; Hadad, O ; Monovich, E ; Granot, N ; Hashoul, S, 2017). In expansion, an extraordinarily outlined minor network will be compelling for the fast-planning datasets habitually experienced in restorative imaging (Sinno J P; Q Yang., 2010) (I R i Haque, J Neubert, 2020) (S Azizi, P Mousavi, P Yan , Amir T, J T Kwak , S Xu , Baris T , P Choyke, P Pinto, B Wood, Purang A, 2017). In a few circumstances, cross-modal trade learning triumphs over cross-domain trade learning (Amit, G ; Ben-Ari, R ; Hadad, O ; Monovich, E ; Granot, N ; Hashoul, S, 2017) (Nima T, Laura J, Q Li, Jeffrey N C, Z Wu, X Ding, 2020). Distinctive imaging modalities, counting as attractive reverberation imaging (MRI), mammography, computed tomography, and ultrasound (US), are regularly utilized inside the model workflow in restorative circumstances, especially in breast imaging (F M Calisto, Nuno J N, Jacinto C N, 2020) (Nagwa Dongola, MD, 2020)

Mammography (also known as X-rays) and ultrasound are the first-line screening methods for evaluating breast cancer since they can both be prepared considerably more quickly than Magnetic resonance imaging (MRI) and Computed Tomography Scan (CT) (Eggertson, 2004). It is difficult to construct datasets and carry out ground-truth explanations because breast MRI is significantly more time-consuming, expensive, and frequently utilised to screen higher-risk individuals (Hadad, O.; Bakalo, R.; Ben-Ari, R.; Hashoul, S.; Amit, G., 2017). The best strategy in these circumstances is cross-modal transfer learning (W Li, S Gu, X Zhang, Tao C, 2020) (E Zhong, W Fan,O Yang, Olivier V, J Ren, 2010). A few experiments (Hadad, O.; Bakalo, R.; Ben-Ari, R.; Hashoul, S.; Amit, G., 2017) have demonstrated the benefit of cross-modal transfer learning over cross-domain transfer learning for a single task and in the case of limited training datasets. For transfer learning, two common techniques are feature extraction and finetuning. (Elif B, H Dogan, M. Erçin, S. Ersoz, M. Ekinci, 2019).

Chapter 4: Problem Statement

It shows that only 20% to 50% of patients with breast cancer in its early stages get diagnosed in the majority of low- and middle-income nations, in contrast to the majority of high-income countries. The majority of research conducted in the developed world reveals a link between breast cancer in late clinical stages and gaps of more than three months between the onset of symptoms and the beginning of therapy. In high-income nations, this gap typically lasts 30 to 48 days, but in lower and middle-income countries, it lasts 3 to 8 months.

The biggest delays take place during the provider gap or the period between the initial medical consultation and the beginning of therapy. The little information suggests that access limitations and subpar cancer care are to blame for provider delays in low- and middle-income nations. There is hardly any research on these topics in these countries, which are most in need of it to design cost-effective public policies that enhance health systems and address particular access hurdles and gaps in the quality of care for the early detection and treatment of breast cancer.

Chapter 5: Methodology Used

5.1. Data Upload

1. The dataset was gathered from Kaggle and consists of 693 ultrasound pictures of breast cancer from 3 different classes: benign, malignant, and normal. Below is the link to the dataset used.
   https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data

2. Implementing Python3 programmes requires the use of Jupyter Notebook and the Python libraries Numpy, Matplotlib, Pandas, Seaborn, Sklearn, Tensorflow, and Keras.

5.2. Feature Extraction

1. Feature extraction method uses a CNN model which has been properly trained on a larger dataset like ImageNet, making it an included extractor for the underutilised target space, such as breast ultrasound imaging (Veronika C, M de Bruijne, J P W Pluim , 2019). The well-trained CNN model resolves all of the convolutional layers while cleaning up the completely associated layers in particular (B Chu, V Madhavan, O Beijbom, J Hoffman, T Darrell , 2016) (Csefalvay, 2019). To deal with an unrelated task, such as a task to classify breast cancer, the convolution layers are used as a stabilised highlight extractor. After that, the classifier that can generate fully linked layers is given the retrieved characteristics (N Best, J Ott, E J. Linstead , 2020). Finally, not the complete organise but rather what may be considered (PyTorch, 2017) the preparatory stage of the current classifier is prepared (S Imai, S Kawai, H Nobuhara , 2020) (Transfer Learning (C3W2L07), 2017).

3. The procedures of feature extractor and fine-tuning were found to be two transfer learning procedures. They also have the advantage of not requiring neural network preparation, which makes it necessary to coordinate the extracted features when using image analysis forms (Veronika C, M de Bruijne, J P W Pluim , 2019). However, many researchers have conducted in-depth investigations to distinguish the approach that produces the most noteworthy results. Three diverse preparation strategies are proposed in (M Byra, M

Galperin, H O Fournier, L Olson, M O'Boyle, C Comstock, M Andre, 2019): a CNN design taught from scratch, a transfer learning method employing a pre-trained VGG16 CNN engine that is advanced and prepared to utilize ultrasound pictures, and a fine-tuned learning strategy utilizing deep learning parameters.

5.3. Benign Tumours

1. Benign tumours are those that stay in their original location and do not spread to other bodily areas. They do not propagate to surrounding structures or distant parts of the body. Frequently, benign tumours are well-defined and develop at a slow rate.
2. Most benign tumours are not harmful. They may, however, become enormous and exert pressure on other structures, causing pain or other health problems. A large benign lung tumour, for instance, may constrict the trachea and make breathing challenging. It would need to be removed surgically right away. Benign tumours are unlikely to recur after removal. Examples of benign tumours are cutaneous lipomas and uterine fibroids.
3. Some benign tumours have the potential to turn cancerous. These are regularly monitored and could require surgical removal. For instance, since they have the potential to develop into cancer, colon polyps, another word for an aberrant clump of cells, are routinely surgically removed.

5.4. Malignant Tumours

1. Malignant tumour cells proliferate uncontrollably and disperse locally or farther afield. These are cancerous tumours that pose a threat (ie, they attack other destinations). They disseminate to distant places via the lymphatic or blood systems. This dispersal is referred to as metastasis. Although it can occur anywhere in the body, metastasis most frequently affects the liver, lungs, brain, and bone.

2. Malignant tumours must be treated to stop their fast spread. If they are discovered early, surgery is the most popular form of treatment, with chemotherapy or radiation serving as

backup options. Chemotherapy or immunotherapy may be used as a systemic treatment if cancer has spread.

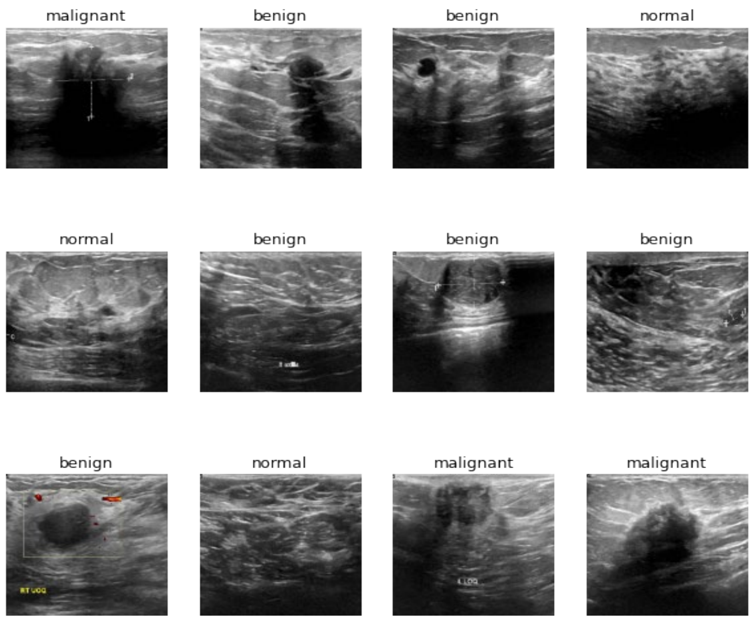

Figure 3 Image Before Augmentation

5.5. Data Augmentation

The dataset has been increased since more labelled data improves how well the CNN model function. The augmentation procedure for visual data includes picture altering techniques including rotation, translation, scaling, and flipping layouts. The data augmentation of our dataset is shown below

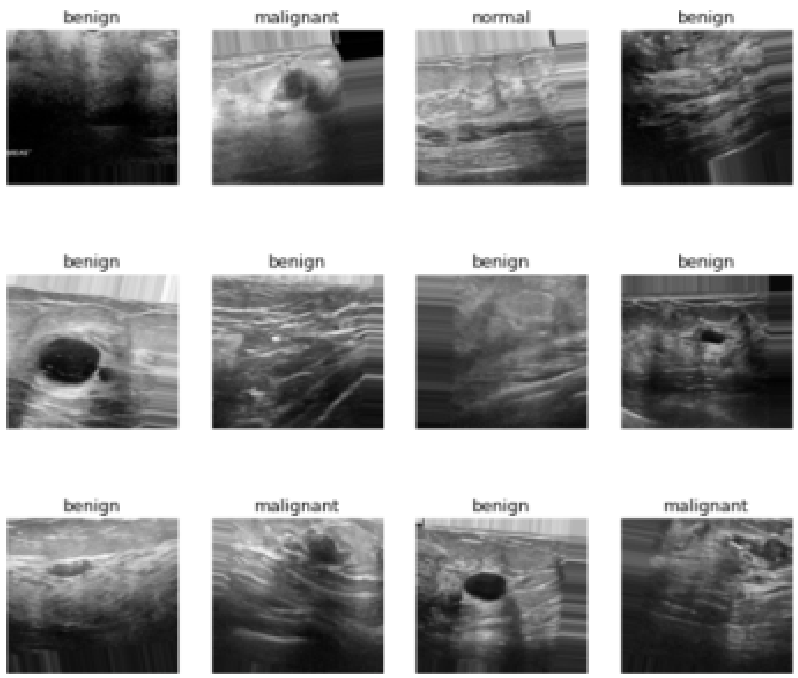

Figure 4 Image after augmentation

Chapter 6: Applications of Deep Learning in Breast Cancer

Breast imaging is essential for locating breast cancer early on as well as for monitoring and evaluating the illness while it is being treated. The techniques most often used for breast imaging are computerised breast tomosynthesis, ultrasound, appealing reverberation imaging, and advanced mammography. For axillary lymph hub differentiation, axillary lymph hub categorization, and distant arrangement in breast cancer imaging, atomic pharmaceutical imaging techniques are used. Breast imaging may use deep learning, a branch of manufactured insights, as each of these techniques is now automated.

Deep learning is now used for a wide range of tasks, including injury localization and division, image creation and editing, cancer risk prediction, and the forecast and evaluation of therapy response. Although it is clear that more extensive studies are needed, particularly for ultrasound and attractive reverberation imaging, to fully assess the added value of deep learning in breast cancer imaging, researchers show comparable and, in fact, more frequent displays of deep learning calculations than radiologists. Before deep learning in clinical breast care can be applied to its fullest, several legitimate and ethical problems need to be taken into account.

Chapter 7: Model Development

This chapter deals with the model developed for breast cancer detection. First, we will develop MobilenetV2 and then InceptionV3.

Given its lightweight design, MobilenetV2 will be used. It employs depth-aware separate convolutions, which essentially means that each colour channel undergoes a separate convolution rather than being added together and smoothed. Due to clever convolution at a deep level, each input channel for MobileNets receives a single channel. After the pointwise convolution, the outputs of the depth-aware convolution are mixed using a 1 x 1 convolution. A routine convolution filters and mixes inputs in one step to create a contemporary set of outputs. The depth-aware distinguishing convolution may divide this into two layers: a layer for merging and a layer for filtering. Due to this factorization, the computation and demonstrate estimate are both significantly reduced.

Imagenet is a common tool for categorising photographs. Millions of images taken using smartphones are used in a competition every year with 1000 categories. The models used in the imagenet classification tasks are compared against one another in terms of how well they performed. It provides a "standard" evaluation of how well a photo categorization model is working as a consequence. A wide range of exchange learning demonstration models employs imagenet weights. If using exchange learning, may adjust your demonstration by adding extra layers to make it more applicable to your application. Although it is not essential, using the imagenet weights often provides advantages since it expedites model joining.

The input shape must be either (224, 224, 3) (with channels last data format) or (3, 224, 224) (with channels first data format), and it may only be supplied if the included top is False. Its width and height should both be no less than 32, and it should have precisely 3 input channels. One acceptable value would be (200, 200, 3). The included top is used to include the top three completely linked tiers of the network. Therefore, the MobileNet architecture is developed as follows by taking the mentioned details:

```
#Initializing the mobilenet architecture
mobilenet_model = tf.keras.applications.MobileNet(
                weights="imagenet",
                input_shape=(224,224,3),
                include_top=False,
            )
```

```
Downloading data from https://storage.googleapis.com/tensorflow/keras-applications/mobilenet/mobilenet_1_0_224_tf_no_top.h5
17227776/17225924 [==============================] - 0s 0us/step
17235968/17225924 [==============================] - 0s 0us/step
```

The model summary table provides information on how closely the model and the dependent variable are related. The multiple correlation coefficient measures the relationship between the values of the dependent variable as predicted and as observed. The model summary is given as:

```
In [14]: mobilenet_model.summary()
```

```
Model: "mobilenet_1.00_224"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 224, 224, 3)]     0

 conv1 (Conv2D)              (None, 112, 112, 32)      864

 conv1_bn (BatchNormalizatio  (None, 112, 112, 32)     128
 n)

 conv1_relu (ReLU)           (None, 112, 112, 32)      0

 conv_dw_1 (DepthwiseConv2D)  (None, 112, 112, 32)     288

 conv_dw_1_bn (BatchNormaliz  (None, 112, 112, 32)     128
 ation)

 conv_dw_1_relu (ReLU)       (None, 112, 112, 32)      0
```

The boolean property trainable is additionally a property of models and layers. Its value can change. When layer trainable is set to False, all of the layer's weights are changed from trainable to non-trainable. When training with the fit() or with any custom loop that depends on trainable weights to apply gradient changes, the state of a frozen layer won't be changed. The layer is "frozen" in this situation.

The sequential model has layers that are stacked linearly. CNNs often use a sequential architecture in their construction. However, not all architectural designs are linear stacks. The simplest modelling technique accessible in Keras is sequential modelling. It enables us to construct models progressively. For a straightforward stack of layers, where each layer has precisely one input tensor and one output tensor, the sequential method works well. When a model has multiple inputs or outputs, the sequential model is inappropriate. Each layer has several inputs and outputs.

To successfully model the input layer and construct a neural network model, the Keras layers are used. Flatten function flattens the multi-dimensional input tensors into a single dimension. From there, we may transfer those inputs into every neuron of the model.

A layer that is closely coupled to the layer above it in any neural network means that every neuron in that layer is associated with every other neuron in the above layer. This layer is the one that is mostly used in neural networks.

A training technique called dropout randomly ignores certain neurons. They randomly "drop out." Accordingly, any weight adjustments are not made to the cell on the return journey, and their effects on the activation of downstream neurons are temporally erased.

The rectified linear activation function, or ReLU in simple terms, give the output zero if the inputs are negative and the input directly if the input is positive. It has developed into the default activation function for numerous kinds of neural networks since a model that uses it is typically more effective and easier to train.

This idea is broadened by Softmax to include many classes. Softmax achieves this by assigning decimal probabilities to every class in a multi-class problem. They must have decimal probabilities that add up to 1.0. Training converges more quickly than it would without the additional constraint. We have considered all the above-mentioned concepts for the development of the model.

```
# making the all the layer of the mobilenet is untrainable as we don't wnat to train those again.
for layer in mobilenet_model.layers:
    layer.trainable = False
```

```
# Making the mobilenet model into Sequential model and customizing the model
model = Sequential()
model.add(mobilenet_model)
model.add(Flatten())
model.add(Flatten())
model.add(BatchNormalization())
model.add(Dense(512,activation='relu'))
model.add(Dropout(0.4))
model.add(Dense(256,activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(3,activation='softmax'))
```

Again, the model summary is given as:

```
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 mobilenet_1.00_224 (Functio  (None, 7, 7, 1024)       3228864
 nal)

 flatten (Flatten)           (None, 50176)             0

 flatten_1 (Flatten)         (None, 50176)             0

 batch_normalization (BatchN  (None, 50176)            200704
 ormalization)

 dense (Dense)               (None, 512)               25690624

 dropout (Dropout)           (None, 512)               0

 dense_1 (Dense)             (None, 256)               131328

 dropout_1 (Dropout)         (None, 256)               0

 dense_2 (Dense)             (None, 128)               32896

 dropout_2 (Dropout)         (None, 128)               0

 dense_3 (Dense)             (None, 3)                 387

=================================================================
Total params: 29,284,803
Trainable params: 25,955,587
Non-trainable params: 3,329,216
_________________________________________________________________
```

The ModelCheckpoint callback is used to frequently save a model or set of weights (in a checkpoint file) during training with the model. fit() function. After that, the model or weights can be loaded to resume training using the saved state.

A callback called ReduceLROnPlateau is used to lower the learning rate after a metric reaches a plateau. This callback monitors a quantity and lowers the learning rate by a factor value (new lr =

lr * factor) if an improvement is not visible after a consecutive number of epochs when a monitored measure stops improving, discontinue training. This technique we have used for early stopping.

The cross-entropy loss function is utilised when there are two or more label classes. As expected, labels must be provided as integers. There should be n floating point values per feature for y pred and only one floating point value per feature for y true or y actual.

Adam optimization is a stochastic gradient descent method that is based on an adaptive estimate of first- and second-order moments. Therefore, let's apply all the concepts mentioned above and create a checkpoint, ear;y_stopping, reduce_learnningrate and callbacks_list for the model.

```
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
checkpoint = ModelCheckpoint("./model.h5", monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
early_stopping = EarlyStopping(monitor='acc',
                          min_delta=0,
                          patience=5,
                          verbose=1,
                          restore_best_weights=True
                          )
reduce_learningrate = ReduceLROnPlateau(monitor='acc',
                              factor=0.2,
                              patience=5,
                              verbose=1,
                              min_delta=0.0001)
callbacks_list = [early_stopping,checkpoint,reduce_learningrate]
```

```
#Model compilation
model.compile(loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              optimizer = tf.keras.optimizers.Adam(learning_rate=0.001),
              metrics=['accuracy'])
```

A machine learning model's model fitting is a gauge of how well it generalises to data that is comparable to the data it was trained on. Results from a well-fitted model are more precise. An overfitted model too closely resembles the data. The evaluation stage in the model-development process establishes whether the model is the best fit for the issue at hand and the relevant data. The Keras model has a function called evaluate that performs the model evaluation. Test data is one of its three primary reasons.

```
In [21]: history = model.fit(train_data, validation_data = test_data, epochs = 20, callbacks=callbacks_list)

Epoch 1/20
22/22 [==============================] - ETA: 0s - loss: 0.7153 - accuracy: 0.7374WARNING:tensorflow:Early stopping condition
ed on metric `acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy

Epoch 1: val_accuracy did not improve from 0.65517
WARNING:tensorflow:Learning rate reduction is conditioned on metric `acc` which is not available. Available metrics are: los
s,accuracy,val_loss,val_accuracy,lr
22/22 [==============================] - 15s 697ms/step - loss: 0.7153 - accuracy: 0.7374 - val_loss: 0.8917 - val_accuracy:
0.6322 - lr: 0.0010
Epoch 2/20
22/22 [==============================] - ETA: 0s - loss: 0.6363 - accuracy: 0.7662WARNING:tensorflow:Early stopping condition
ed on metric `acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy

Epoch 2: val_accuracy did not improve from 0.65517
WARNING:tensorflow:Learning rate reduction is conditioned on metric `acc` which is not available. Available metrics are: los
s,accuracy,val_loss,val_accuracy,lr
22/22 [==============================] - 15s 676ms/step - loss: 0.6363 - accuracy: 0.7662 - val_loss: 0.7957 - val_accuracy:
0.6437 - lr: 0.0010
Epoch 3/20
```

```
In [22]: loss_test, acc_test = model.evaluate(train_data,verbose=1)
         loss_val, acc_val = model.evaluate(test_data
                                            , verbose=1)
         print("Train: accuracy = %f  ;  loss_v = %f" % (acc_val, loss_val))
         print("Test: accuracy = %f  ;  loss = %f" % (acc_test, loss_test))

22/22 [==============================] - 14s 619ms/step - loss: 0.2812 - accuracy: 0.9004
3/3 [==============================] - 1s 311ms/step - loss: 0.7781 - accuracy: 0.7471
Train: accuracy = 0.747126  ;  loss_v = 0.778137
Test: accuracy = 0.900433  ;  loss = 0.281204
```

The accuracy score is the number of accurate predictions obtained. The divergence from the desired goal state(s) is termed as Loss values. Val accuracy or validation accuracy displays the precision of predictions made after each training session for a validation set that was randomly divided. Nevertheless, overall accuracy kept rising.

The learning algorithm will iterate over the entire training dataset a predetermined number of times depending on the number of epochs, which is a hyperparameter. One epoch signifies that each sample in the training dataset has had an opportunity to have the internal model parameters changed.

Loss and Val loss provide the values of the cost functions for the training and cross-validation sets of data, respectively. Validation data show that neurons using dropout do not discard random neurons.

The function for plotting the model history is given as:

```
# function for plotting the model history
def perf_plot(history):
  train_accuracy = history.history['accuracy']
  val_accuracy = history.history['val_accuracy']
  train_loss = history.history['loss']
  val_loss = history.history['val_loss']
  fig, ax = plt.subplots(nrows=2, ncols=1, figsize=(12, 10))
  ax[0].set_title('Training Accuracy vs. Epochs')
  ax[0].plot(train_accuracy, color = 'blue', marker = 'o', markerfacecolor = 'red',markersize = 10, label='Train Accuracy')
  ax[0].plot(val_accuracy, color = 'red', marker = 'o', markerfacecolor = 'blue',markersize = 10, label='Validation Accuracy')
  ax[0].set_xlabel('Epochs')
  ax[0].set_ylabel('Accuracy')
  ax[0].legend(loc='best')
  ax[1].set_title('Training/Validation Loss vs. Epochs')
  ax[1].plot(train_loss, color = 'blue', marker = 'o', markerfacecolor = 'red',markersize = 10, label='Train Loss')
  ax[1].plot(val_loss, color = 'red', marker = 'o', markerfacecolor = 'blue',markersize = 10, label='Validation Loss')
  ax[1].set_xlabel('Epochs')
  ax[1].set_ylabel('Loss')
  ax[1].legend(loc='best')
  plt.tight_layout()
  plt.show()
```

The plots between accuracy and epochs, training loss and epochs are given as:

```
perf_plot(history)
```

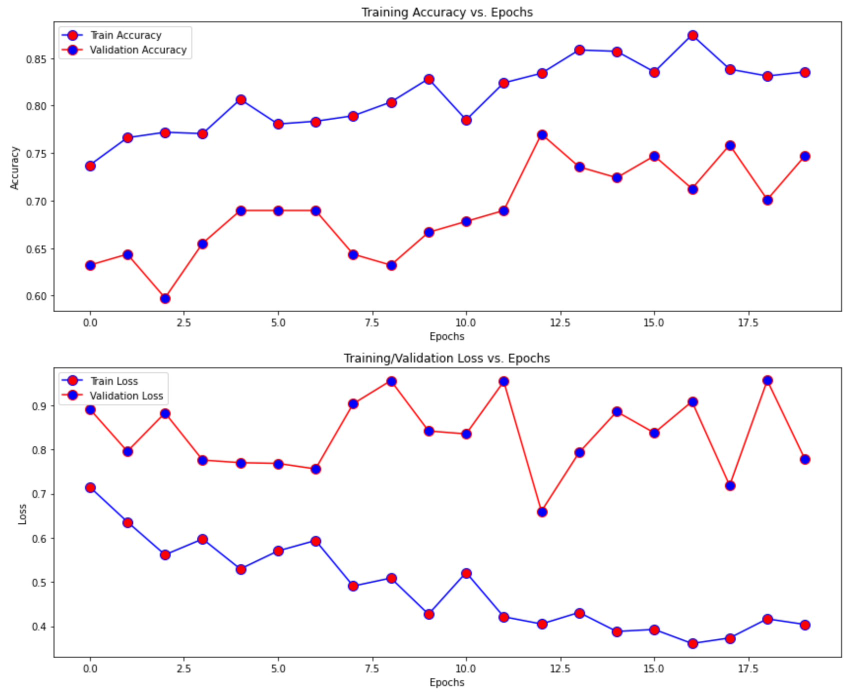

Figure 5 Training accuracy vs epochs and Training validation vs Epochs for MobileNet model

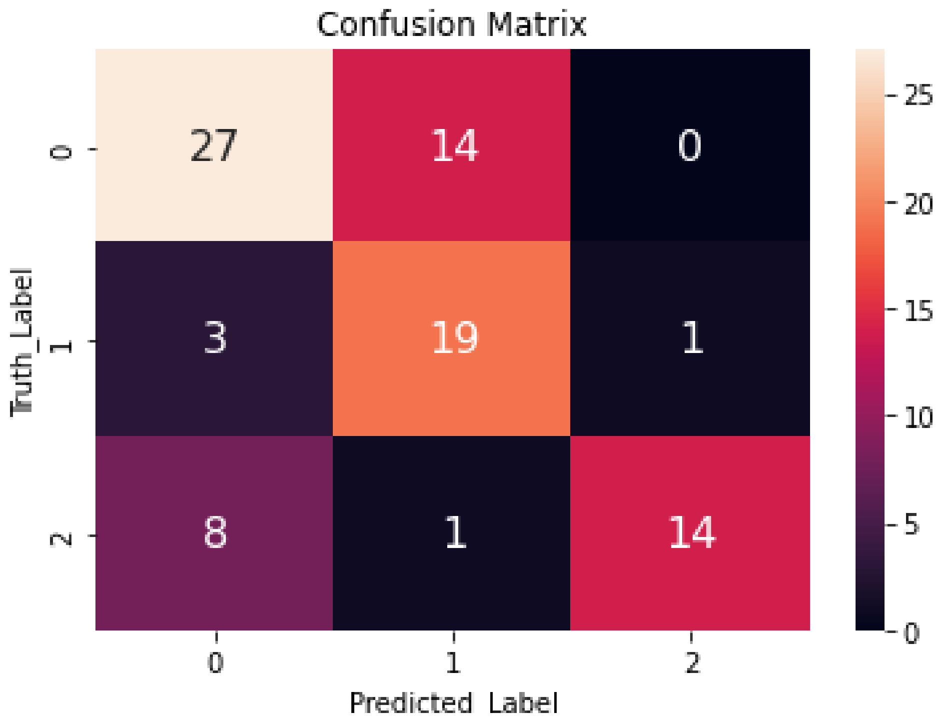

Figure 6 Confusion Matrix for MobileNet model

Figure 6 depicts Confusion Matrix for the MobileNet model.

Figure (6) Matrix shows the colour scheme for three classes (Normal, Benign and Malignant). This graph shows how often the MobileNet model successfully recognised each tumour. This matrix also shows which tumours the model misclassified. The numbers 0, 1, and 2 on the x-axis represent the three classes (Normal, Benign, and Malignant). The model has accurately predicted class 0—a normal class—27 times, but it has also forecast incorrectly 14 times. The model has properly predicted Class 1, a benign class, 19 times, whereas Class 1 has been predicted wrongly 4 times. The model correctly identified the malignant class, which is a class 2 on the truth label, 14 times, and incorrectly identified it 9 times.

We have prepared the test method for the visualization of test results:

```
#visualing the test reslult
def test(data):
  plt.figure(figsize= (15,13))
  for i in data:
    images = i[0] # i was an touple of images and label
    labels = i[1]
    for i in range(12):
          ax = plt.subplot(3, 4, i+1)
          image = images[i]
          ex_image = np.expand_dims(image, axis = 0) #expanding the dimension from 3D to 4D
          Actual_label = class_names[int(labels[i])] #converting into numpy from tensor and Actual class label)
          plt.imshow(image)#printing the image
          prediction = model.predict(ex_image) #prediction on the test image
          predicted_label = class_names[np.argmax(prediction)] #Predicted class name
          confidence = round(100 * np.max(prediction), 2) # Confidence on prediction
          plt.title(f"Actual class: {Actual_label}, \n Predicted: {predicted_label}, \n Confidence:, {confidence}")
          plt.axis('off')

    break
```

According to the results, we have classified the images for the test_data into benign, normal and malignant. The results are shown as follows:

```
test(test_data)
```

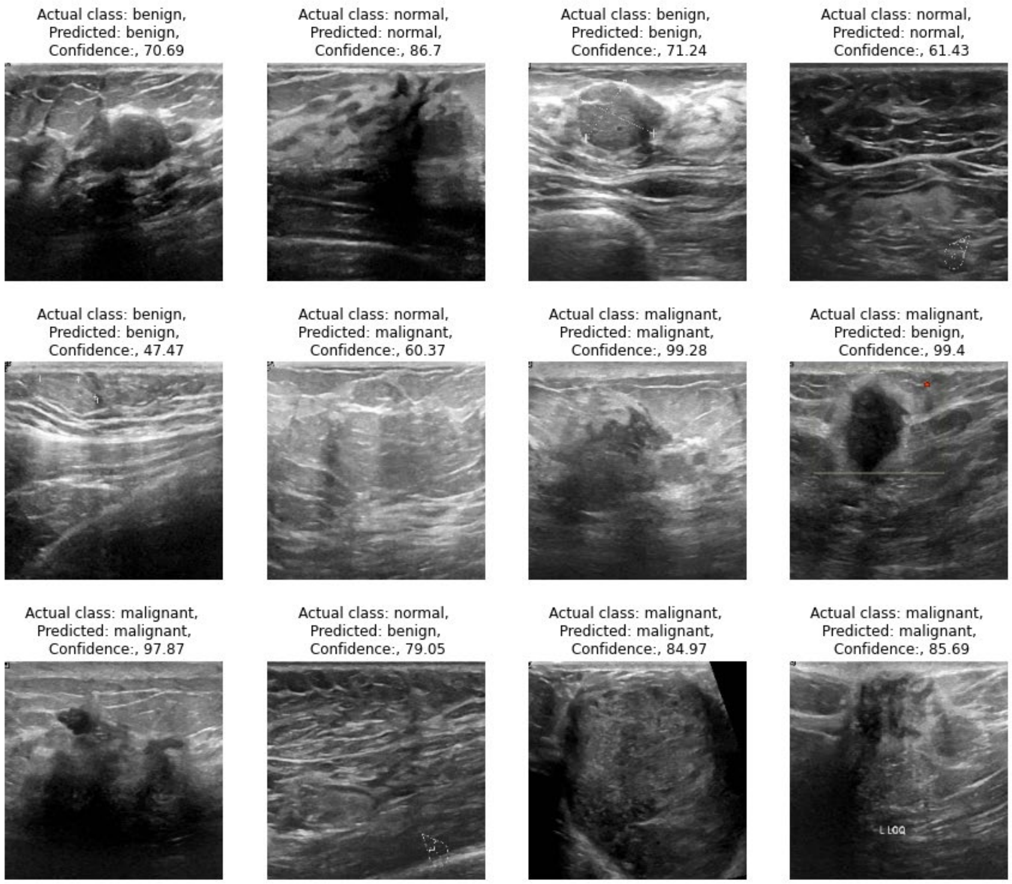

Figure 7 MobileNet result

Next, we applied MobileNetV2 architecture.

A convolutional neural network architecture called MobileNetV2 attempts to operate more effectively on mobile devices. It is built on an inverted residual structure, and the bottleneck levels are connected by residual connections. Lightweight depthwise convolutions are used as a source of non-linearity in the middle expansion layer to weed out features. There are two different kinds of blocks in MobileNetV2. In block one, there is only one more stride. A block that may shorten by two strides is another. Every sort of block has three layers. This time, the first layer consists of ReLU6-based 1 x 1 convolutions. The second layer is the depthwise convolution. There is no non-linearity in the third layer, which is once more a 1 x 1 convolution. Deep networks are stated to only have the power of a linear classifier on the non-zero volume portion of the output domain if ReLU is applied once more. The MobileNetV2 architecture also includes 19 extra bottleneck layers and a 32-filter first fully convolution layer. The model is shown.

```
#Initializing the mobilenet V2 architecture
mobilenet2_model = tf.keras.applications.MobileNetV2(
                weights="imagenet",
                input_shape=(224,224,3),
                include_top=False,
            )
```

```
mobilenet2_model.summary()
```

```
Model: "mobilenetv2_1.00_224"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_3 (InputLayer)           [(None, 224, 224, 3  0           []
                                )]

 Conv1 (Conv2D)                 (None, 112, 112, 32  864         ['input_3[0][0]']
                                )

 bn_Conv1 (BatchNormalization)  (None, 112, 112, 32  128         ['Conv1[0][0]']
                                )

 Conv1_relu (ReLU)              (None, 112, 112, 32  0           ['bn_Conv1[0][0]']
                                )

 expanded_conv_depthwise (Depth  (None, 112, 112, 32  288        ['Conv1_relu[0][0]']
 wiseConv2D)                    )
```

```python
# making the all the layer of the mobilenet_v2 is untrainable as we don't wnat to train those again.
for layer in mobilenet2_model.layers:
    layer.trainable = False
```

```python
# Making the mobilenet2 model into Sequential model and customizing the model
model = Sequential()
model.add(mobilenet2_model)
model.add(Flatten())
model.add(BatchNormalization())
model.add(Dense(512,activation='relu'))
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Dense(256,activation='relu'))
model.add(Dropout(0.3))
model.add(BatchNormalization())
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(3,activation='softmax'))
```

```python
#Model compilation
model.compile(loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
             optimizer = tf.keras.optimizers.Adam(learning_rate=0.001),
             metrics=['accuracy'])
```

```python
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
checkpoint = ModelCheckpoint("./mobilenetv2.h5", monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
early_stopping = EarlyStopping(monitor='acc',
                          min_delta=0,
                          patience=5,
                          verbose=1,
                          restore_best_weights=True
                          )

reduce_learningrate = ReduceLROnPlateau(monitor='acc',
                              factor=0.2,
                              patience=5,
                              verbose=1,
                              min_delta=0.0001)
callbacks_list = [early_stopping,checkpoint,reduce_learningrate]
```

```
historty_2 = model.fit(train_data, validation_data = test_data, epochs = 25, callbacks = callbacks_list)
```

```
Epoch 1/25
22/22 [==============================] - ETA: 0s - loss: 1.2149 - accuracy: 0.4805WARNING:tensorflow:Early stopping conditio
ned on metric `acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy

Epoch 1: val_accuracy improved from -inf to 0.56322, saving model to ./mobilenetv2.h5
WARNING:tensorflow:Learning rate reduction is conditioned on metric `acc` which is not available. Available metrics are: los
s,accuracy,val_loss,val_accuracy,lr
22/22 [==============================] - 23s 867ms/step - loss: 1.2149 - accuracy: 0.4805 - val_loss: 1.6512 - val_accuracy:
0.5632 - lr: 0.0010
Epoch 2/25
22/22 [==============================] - ETA: 0s - loss: 0.8945 - accuracy: 0.6176WARNING:tensorflow:Early stopping conditio
ned on metric `acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy

Epoch 2: val_accuracy did not improve from 0.56322
WARNING:tensorflow:Learning rate reduction is conditioned on metric `acc` which is not available. Available metrics are: los
s,accuracy,val_loss,val_accuracy,lr
22/22 [==============================] - 15s 688ms/step - loss: 0.8945 - accuracy: 0.6176 - val_loss: 1.1939 - val_accuracy:
0.4828 - lr: 0.0010
Epoch 3/25
```

The plots between training accuracy and epochs, training loss and epochs are given as:

```
#Visualizing the model training phase of mobilenetv2
perf_plot(historty_2)
```

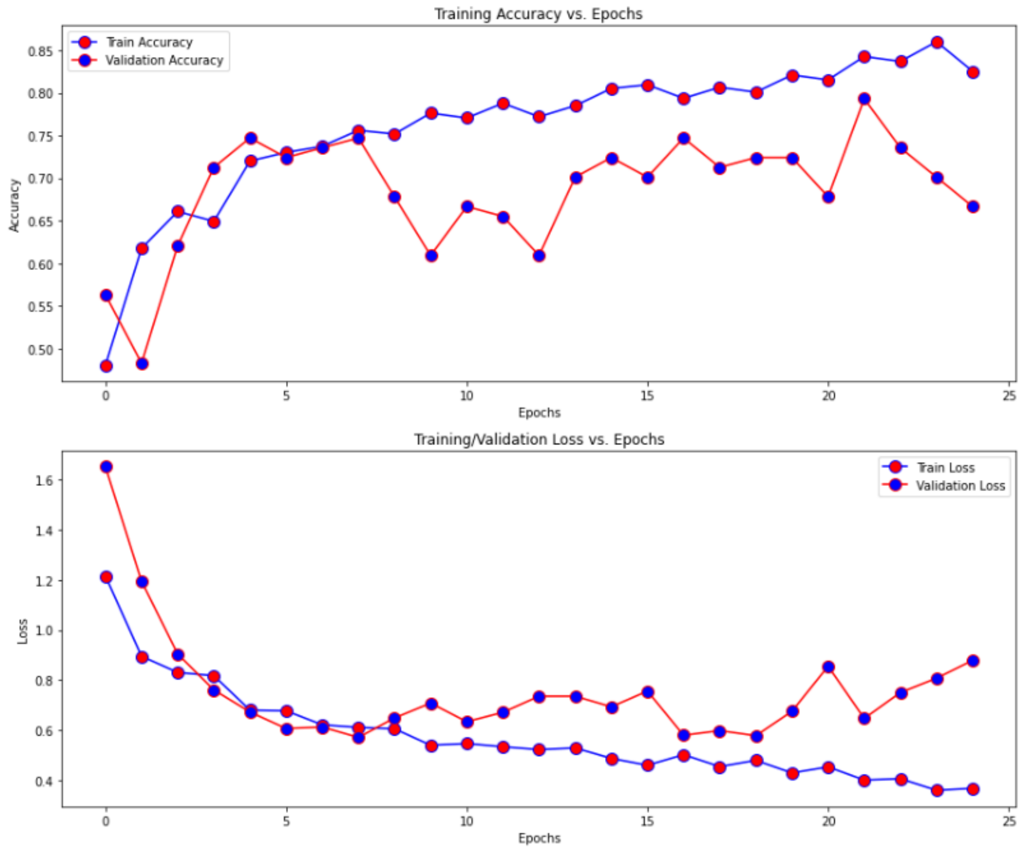

Figure 8 Training Accuracy vs Epoch & Training validation vs Epochs for MobileNet_V2

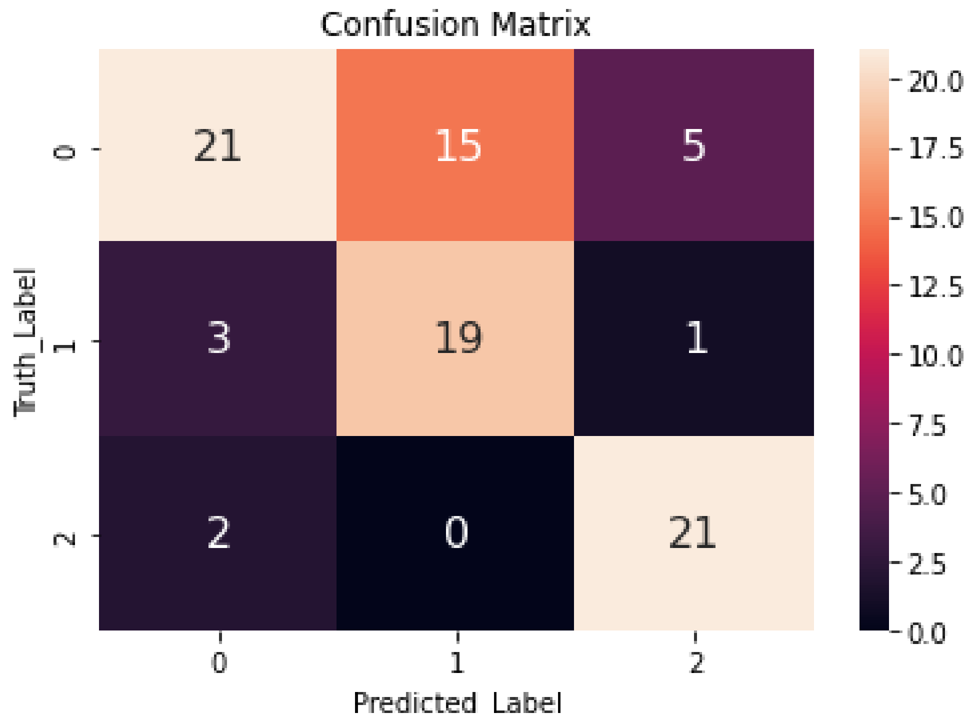

Figure 9 Confusion Matrix for MobileNet_V2

Figure 9 depicts Confusion Matrix for the MobileNet_V2 model.

Figure (9) Matrix shows the colour scheme for three classes (Normal, Benign and Malignant). This graph shows how often the MobileNet_V2 model successfully recognised each tumour. This matrix also shows which tumours the model has misclassified. The numbers 0, 1, and 2 on the x-axis represent the three classes (Normal, Benign, and Malignant). The model has accurately predicted class 0—a normal class—21 times, but it has also forecast incorrectly 20 times. The model has properly predicted Class 1, a benign class, 19 times, whereas Class 1 has been predicted wrongly 4 times. The model correctly identified the malignant class, which is a class 2 on the truth label, 21 times, and incorrectly identified it just 2 times.

Finally, we will apply InceptionV3 architecture for breast cancer detection.

On the ImageNet dataset, it has been demonstrated that the image recognition model Inception v3 can achieve higher than 78.1% accuracy. The model is the result of numerous concepts that various researchers have created over time.

Some of the symmetric and asymmetric building blocks that make up the model itself include convolutions, average pooling, max pooling, concatenations, dropouts, and completely connected layers. The activation inputs are also subjected to batch normalisation, which is heavily utilised by the model. Using SoftMax, the loss is calculated.

This method decreases the computational efficiency of a network by reducing the number of parameters employed in it. Additionally, it keeps an eye on the network's efficiency.

More compact convolutions undoubtedly, replacing bigger convolutions with smaller ones speed up training. Consider a 5 x 5 filter that has 25 parameters. Two 3 x 3 filters can replace it, leaving just 18 (3 x 3 + 3 x 3) parameters.

Asymmetrical convolutions Instead of a 3 x 3 convolution, a 1 x 3 convolution might be followed by a 3 x 1 convolution. The number of parameters would be a little bit larger than the suggested asymmetric convolution if a 3 x 3 convolution were swapped out for a 2 x 2 convolution.

A mini-CNN is used as an extra classifier during training. Due to its interlayer positioning, any loss adds to the overall network loss. Inception v3 uses auxiliary classifiers as a regularizer, whereas GoogLeNet uses them for a deeper network.

To lower grid size, pooling methods are frequently utilised. The computational cost limitations, however, are addressed using a more practical approach.

```
#Initializing the VGG16 architecture
in_model = tf.keras.applications.inception_v3.InceptionV3(
                weights="imagenet",
                input_shape=(224,224,3),
                include_top = False

            )
```

```
for layer in in_model.layers:
    layer.trainable = False
```

```
in_model.summary()
```

```
Model: "inception_v3"
__________________________________________________________________________________________________
 Layer (type)                   Output Shape         Param #     Connected to
==================================================================================================
 input_14 (InputLayer)          [(None, 224, 224, 3  0           []
                                )]

 conv2d_94 (Conv2D)             (None, 111, 111, 32  864         ['input_14[0][0]']
                                )

 batch_normalization_113 (Batch  (None, 111, 111, 32  96         ['conv2d_94[0][0]']
 Normalization)                 )

 activation_94 (Activation)     (None, 111, 111, 32  0           ['batch_normalization_113[0][0]']
                                )

 conv2d_95 (Conv2D)             (None, 109, 109, 32  9216        ['activation_94[0][0]']
                                )
```

```
# Making the mobilenet2 model into Sequential model and customizing the model
model = Sequential()
model.add(in_model)
model.add(Flatten())
model.add(BatchNormalization())
model.add(Dense(512,activation='relu'))
model.add(Dropout(0.4))
model.add(BatchNormalization())
model.add(Dense(256,activation='relu'))
model.add(Dropout(0.3))
model.add(BatchNormalization())
model.add(Dense(128,activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(3,activation='softmax'))
```

```
model.summary()
```

```
Model: "sequential_10"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 inception_v3 (Functional)   (None, 5, 5, 2048)        21802784

 flatten_9 (Flatten)         (None, 51200)             0

 batch_normalization_210 (Ba  (None, 51200)            204800
 tchNormalization)

 dense_31 (Dense)            (None, 512)               26214912

 dropout_21 (Dropout)        (None, 512)               0

 batch_normalization_211 (Ba  (None, 512)              2048
 tchNormalization)

 dense_32 (Dense)            (None, 256)               131328

 dropout_22 (Dropout)        (None, 256)               0

 batch_normalization_212 (Ba  (None, 256)              1024
 tchNormalization)

 dense_33 (Dense)            (None, 128)               32896

 dropout_23 (Dropout)        (None, 128)               0

 dense_34 (Dense)            (None, 3)                 387

=================================================================
Total params: 48,390,179
Trainable params: 26,483,459
Non-trainable params: 21,906,720
_________________________________________________________________
```

```
#Model compilation
model.compile(loss = tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
             optimizer = tf.keras.optimizers.Adam(learning_rate=0.001),
             metrics=['accuracy'])
```

```
from tensorflow.keras.callbacks import ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
checkpoint = ModelCheckpoint("./incep_model.h5", monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')
early_stopping = EarlyStopping(monitor='acc',
                          min_delta=0,
                          patience=5,
                          verbose=1,
                          restore_best_weights=True
                          )
reduce_learningrate = ReduceLROnPlateau(monitor='acc',
                              factor=0.2,
                              patience=5,
                              verbose=1,
                              min_delta=0.0001)
callbacks_list = [early_stopping,checkpoint,reduce_learningrate]
```

```
historty_3 = model.fit(train_data, validation_data = test_data, epochs = 30, callbacks = callbacks_list)
```

```
Epoch 1/30
22/22 [==============================] - ETA: 0s - loss: 1.1495 - accuracy: 0.5224WARNING:tensorflow:Early stopping conditio
ned on metric `acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy

Epoch 1: val_accuracy improved from -inf to 0.52874, saving model to ./incep_model.h5
WARNING:tensorflow:Learning rate reduction is conditioned on metric `acc` which is not available. Available metrics are: los
s,accuracy,val_loss,val_accuracy,lr
22/22 [==============================] - 25s 899ms/step - loss: 1.1495 - accuracy: 0.5224 - val_loss: 1.3666 - val_accuracy:
0.5287 - lr: 0.0010
Epoch 2/30
22/22 [==============================] - ETA: 0s - loss: 0.8588 - accuracy: 0.6479WARNING:tensorflow:Early stopping conditio
ned on metric `acc` which is not available. Available metrics are: loss,accuracy,val_loss,val_accuracy

Epoch 2: val_accuracy improved from 0.52874 to 0.56322, saving model to ./incep_model.h5
WARNING:tensorflow:Learning rate reduction is conditioned on metric `acc` which is not available. Available metrics are: los
s,accuracy,val_loss,val_accuracy,lr
22/22 [==============================] - 18s 802ms/step - loss: 0.8588 - accuracy: 0.6479 - val_loss: 1.3710 - val_accuracy:
0.5632 - lr: 0.0010
Epoch 3/30
```

```
loss_test, acc_test = model.evaluate(train_data,verbose=1)
loss_val, acc_val = model.evaluate(test_data
                                   , verbose=1)
print("Train: accuracy = %f  ;  loss_v = %f" % (acc_val, loss_val))
print("Test: accuracy = %f  ;  loss = %f" % (acc_test, loss_test))
```

```
22/22 [==============================] - 16s 738ms/step - loss: 0.3450 - accuracy: 0.8600
3/3 [==============================] - 1s 312ms/step - loss: 0.8361 - accuracy: 0.6897
Train: accuracy = 0.689655  ;  loss_v = 0.836085
Test: accuracy = 0.860029  ;  loss = 0.344993
```

```
perf_plot(historty_3)
```

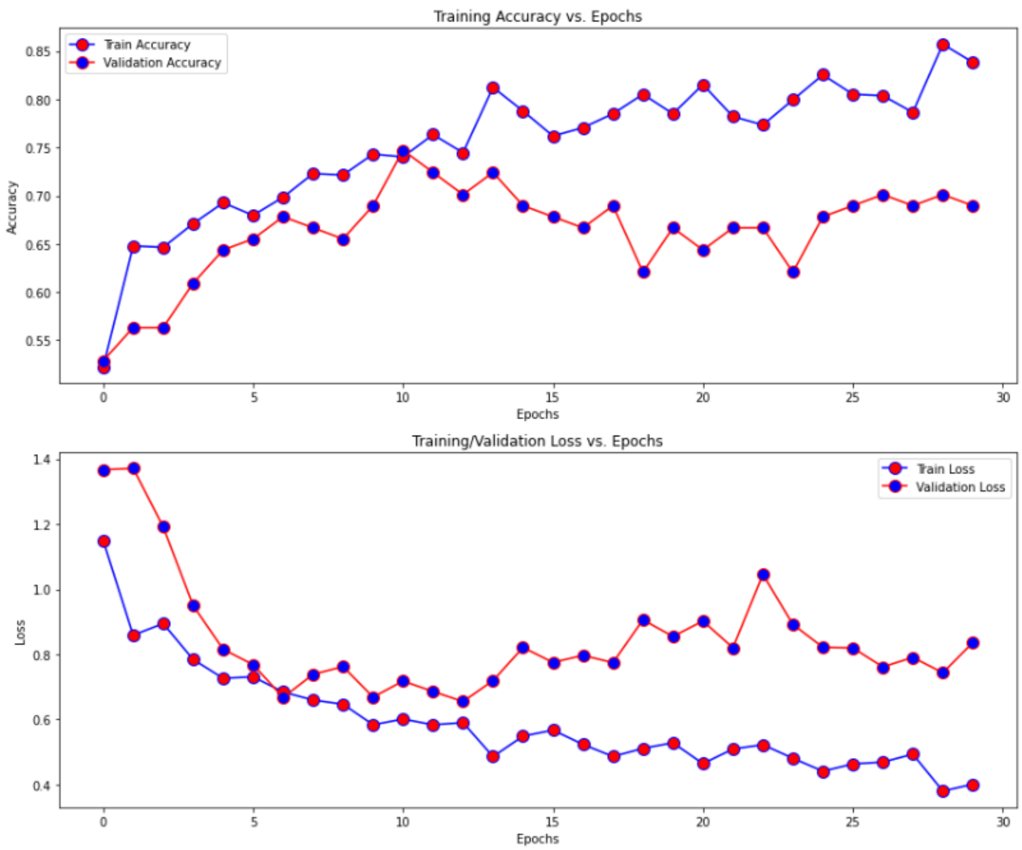

Figure 10 Training Accuracy vs Epoch & Training validation vs Epochs for Inception_V3

```
test(test_data)
```

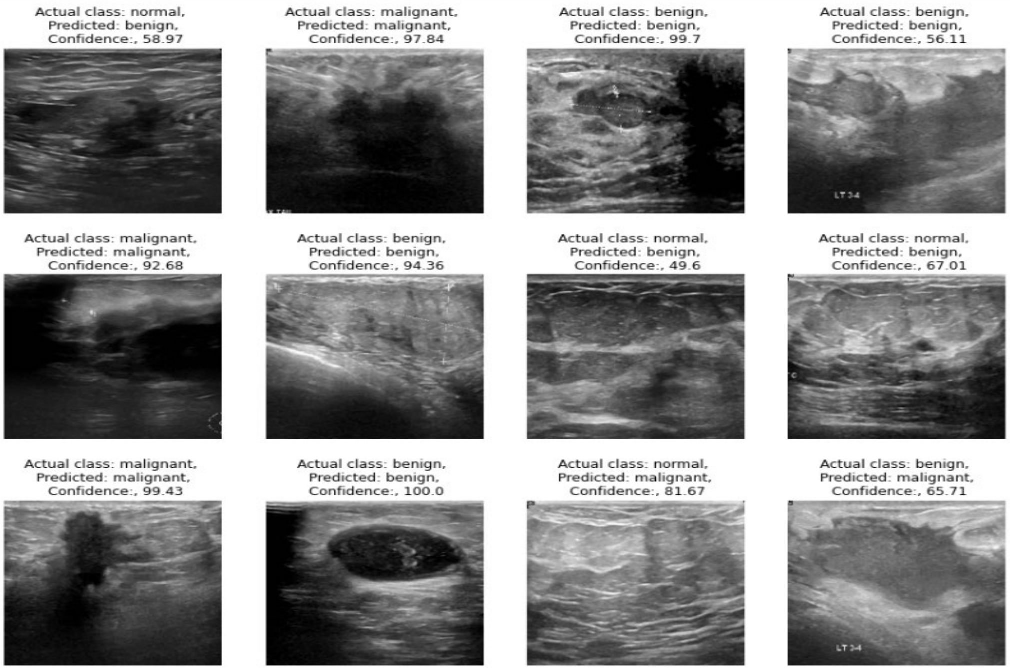

Figure 11 Inception_V3 Result

Chapter 8: Result Analysis

As per the methods, we applied for breast cancer detection, the accuracies are as follows:

1. MobilenetV2: 82.54%
2. InceptionV3: 83.84%

Therefore, according to the results; it's clear that InceptionV3 has got the highest accuracy and can be used for breast cancer detection apart from other methods

The figure 5 depicts the Accuracy graph. It shows the training accuracy which started from 0.73 and gradually it increased but again decreased at 0.80 (contributing for an average result) The second graph of the figure 5 model (MobileNet) is the Validation Loss graph: This graph gives us the accuracy of the model (basically it says about the difference between actual and predictive result of the model) This should be less. Here we can see that the training loss was initially 0.9 and it gradually decreased to 0.8. In the second line of this graph, we can observe that the validation loss initially was 0.7 and now it has come down to 0.45. This model gave the results which weren't too bad. But for better efficiency I have used here the second method to train models and that method is MobileNetV2 Architecture. The graph is shown in figure 8 After giving input to this training model, we obtained the following graphs and the results are visible that it is better than the previous one. However, there is not much major difference in it. Training accuracy have increased here gradually but there is not much difference in the validation accuracy line graph.

The third graph which is shown in figure 10 is the InceptionV3 Architecture graph, where data is customized into sequential model and then compiled and fitted. Here by using this method we can see that the training accuracy have abruptly increased and reached from 0.55 to 0.85. Also, we can observe that in the training validation loss graph the validation and training loss is decreased which was supposed to decrease. This method giving us the most accurate results and the trial run through this method is the most confident making this training model the best out of three.

Chapter 9: Conclusion

Given that breast cancer is the most common and dangerous illness, detecting it is an important, but also difficult task. There is less possibility to recover from it, and it is becoming worse every year. Deep learning and machine learning approaches are utilised to diagnose breast cancer. According to our study and previous studies, machine learning algorithms produce superior outcomes in their particular fields. The earlier study was carried out using a variety of machine learning approaches, with some dataset augmentation and enhancement for improved performance. Data science regularly uses freshly discovered deep learning technology. CNN, a deep learning-based method, is employed for the categorization of breast cancer picture data. CNN mostly uses the dataset from the image. Therefore, we applied deep learning-based MobilenetV2 and InceptionV3 techniques for breast cancer detection and got valuable results.

References

1. A Carkacioglu , F Y vural, 2003. SASI: A Generic Texture Descriptor for Image Retrieval. [Online]
Available at: https://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.4.9808
[Accessed July 2022].

2. A Kensert, P J H, Ola S, 2019. Transfer Learning with Deep Convolutional Neural Networks for Classifying Cellular Morphological Changes. SLAS Discov.

3. Alexandre Gonfalonieri, 2019. 5 Ways to Deal with the Lack of Data in Machine Learning. [Online]
Available at: https://www.kdnuggets.com/2019/06/5-ways-lack-data-machine-learning.html
[Accessed July 2022].

4. Amit, G ; Ben-Ari, R ; Hadad, O ; Monovich, E ; Granot, N ; Hashoul, S, 2017. Classification of breast MRI lesions using small-size training sets: comparison of deep learning approaches. [Online]
Available at: https://ui.adsabs.harvard.edu/abs/2017SPIE10134E..1HA/abstract
[Accessed July 2022].

5. Anon., n.d. Md Zahangir Alom, Tarek M. Taha, Christopher Yakopcic, Stefan Westberg, Paheding Sidike, Mst Shamima Nasrin, Brian C Van Esesn, Abdul A S. Awwal, VijayanK. Asari. [Online].

6. B Chu, V Madhavan, O Beijbom, J Hoffman, T Darrell , 2016. Best Practices for Fine-Tuning Visual Classifiers to New Domains. [Online]
Available at: https://link.springer.com/chapter/10.1007/978-3-319-49409-8_34
[Accessed July 2022].

7. B Huynh,K Drukker,M Giger, 2016. Computer-Aided Diagnosis of Breast Ultrasound Images Using Transfer Learning From Deep Convolutional Neural Networks. Medical Physics The International Journal of Medical Physics Research and Practice, 43(6).

8. B Neyshabur, H Sedghi, C Zhang, 2020. What is being transferred in transfer learning?. [Online]
Available at: https://arxiv.org/abs/2008.11687
[Accessed July 2022].

9. B Neyshabur, H Sedghi, C Zhang, 2021. What is being transferred in transfer learning?. [Online]
Available at: https://arxiv.org/abs/2008.11687
[Accessed July 2022].

10. Brownlee, J., 2017. A Gentle Introduction to Transfer Learning for Deep Learning. [Online]
Available at: https://machinelearningmastery.com/transfer-learning-for-deep-learning/
[Accessed July 2022].

11. Carremans, B., 2018. Handling overfitting in deep learning models. [Online]
Available at: https://towardsdatascience.com/handling-overfitting-in-deep-learning-models-c760ee047c6e
[Accessed July 2022].

12. Coleman, C., 2017. Early Detection and Screening for Breast Cancer.. [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/28365057/
[Accessed June 2021].

13. Csefalvay, C. v., 2019. Transfer learning: the dos and don'ts. [Online]
Available at: https://medium.com/starschema-blog/transfer-learning-the-dos-and-donts-165729d66625
[Accessed July 2022].

14. D Nguyen, S Sridharan, D T Nguyen, S Denman, D Dean, C Fookes, 2020. Meta Transfer Learning for Emotion Recognition. [Online]
Available at: https://arxiv.org/abs/2006.13211
[Accessed July 2022].

15. Dogan, I., 2018. Transfer Learning. [Online]
Available at: https://www.isikdogan.com/blog/transfer-learning.html
[Accessed July 2022].

16. E Zhong, W Fan,O Yang, Olivier V, J Ren, 2010. Cross validation framework to choose amongst models and datasets for transfer learning. 2010th.

17. Eggertson, L., 2004. MRIs more accurate than mammograms but expensive. CMAJ.

18. Elif B, H Dogan, M. Erçin, S. Ersoz, M. Ekinci, 2019. Transfer learning with pre-trained deep convolutional neural networks for serous cell classification. Multimedia Tools and Applications.

19. F M Calisto, Nuno J N, Jacinto C N, 2020. BreastScreening: On the Use of Multi-Modality in Medical Imaging Diagnosis. [Online]
Available at: https://arxiv.org/abs/2004.03500
[Accessed July 2022].

20. fchollet, 2020. Transfer learning & fine-tuning. [Online]
Available at: https://keras.io/guides/transfer_learning/
[Accessed July 2022].

21. G I Parisi, R Kemker, J L Part, C Kanan, S Wermter, 20169. Continual lifelong learning with neural networks: A review. [Online]

Available at: https://pubmed.ncbi.nlm.nih.gov/30780045/
[Accessed July 2022].

22. Gilbert, F.J.; Pinker-Domenig, K., 2019. Diagnosis and Staging of Breast Cancer: When and How to Use Mammography, Tomosyn thesis, Ultrasound, Contrast-Enhanced Mammography, and Magnetic Resonance Imaging.. [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/32096932/
[Accessed June 2021].

23. Hadad, O.; Bakalo, R.; Ben-Ari, R.; Hashoul, S.; Amit, G., 2017. Classification of breast lesions using cross-modal deep learning. [Online]
Available at: https://www.semanticscholar.org/paper/Classification-of-breast-lesions-using-cross-modal-Hadad-Bakalo/971a9b72fc872e912104e83db21e7ccd0ac65f76
[Accessed July 2022].

24. Hindi A,; Peterson C,' Barr RG., 2013. Artifacts in diagnostic ultrasound. [Online]
Available at: https://www.dovepress.com/artifacts-in-diagnostic-ultrasound-peer-reviewed-fulltext-article-RMI#
[Accessed June 2022].

25. I R i Haque, J Neubert, 2020. Deep learning approaches to biomedical image segmentation. Informatics in Medicine Unlocked, Volume 18.

26. IBM Cloud, E., 2020. Neural Networks. [Online]
Available at: https://www.ibm.com/cloud/learn/neural-networks
[Accessed June 2022].

27. IndianTechWarrior., n.d. IndianTechWarrior Fully Connected Layers in Convolutional Neural Networks. [Online]
Available at: https://indiantechwarrior.com/fully-connected-layers-in-convolutional-neural-networks/
[Accessed June 2022].

28. J Schmidt, M R. G. Marques, S Botti, M A. L. Marques, 2019. Recent advances and applications of machine learning in solid-state materials science. npj Computational Materials.

29. J Yosinski, J Clune, Y Bengio, H Lipson, 2014. How transferable are features in deep neural networks?. [Online]
Available at: https://arxiv.org/abs/1411.1792
[Accessed July 2022].

30. J Zhang, W Li, P Ogunbona, D Xu, 2019. Recent Advances in Transfer Learning for Cross-Dataset Visual Recognition: A Problem-Oriented Perspective. [Online]
Available at: https://arxiv.org/abs/1705.04396
[Accessed July 2022].

31. Jonathan,L Jesneck.; Joseph, Y Lo.; Jay A Baker., 2007. Breast mass lesions: computer-aided diagnosis models with mammographic and sonographic descriptors. [Online] Available at: https://pubmed.ncbi.nlm.nih.gov/17562812/
[Accessed June 2022].

32. K He, R Girshick, P Dollar, 2019. Rethinking ImageNet Pre-Training. [Online] Available at: https://openaccess.thecvf.com/content_ICCV_2019/html/He_Rethinking_ImageNet_Pre-Training_ICCV_2019_paper.html
[Accessed July 2020].

33. Karl Weiss, Taghi M. Khoshgoftaar, DingDing Wang , 2016. A survey of transfer learning. Journal of Big Data.

34. Keiron, O,'Shea.;Ryan,Nash., 2015. An Introduction to Convolutional Neural Networks. [Online]
Available at: https://arxiv.org/pdf/1511.08458.pdf
[Accessed June 2022].

35. Koehrsen, W., 2019. Transfer Learning with Convolutional Neural Networks in PyTorch. [Online]
Available at: https://towardsdatascience.com/transfer-learning-with-convolutional-neural-networks-in-pytorch-dd09190245ce
[Accessed July 2022].

36. L J Brattain, B A Telfer, M Dhyani, J R Grajo, A E Samir , 2018. Machine learning for medical ultrasound: status, methods, and future opportunities. [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/29492605/
[Accessed July 2022].

37. L Liu, J Chen, P Fieguth, G Zhao, R Chellappa, M Pietikäinen , 2019. From BoW to CNN: Two Decades of Texture Representation for Texture Classification. International Journal of Computer Vision .

38. M Byra, M Galperin, H O Fournier, L Olson, M O'Boyle, C Comstock, M Andre, 2019. Breast mass classification in sonography with transfer learning using a deep convolutional neural network and color conversion. [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/30589947/
[Accessed July 2022].

39. M H Yap, G Pons, J Marti, S Ganau, M Sentis, R Zwiggelaar, A K Davison, R Marti, M H Yap, G Pons, J Marti, S Ganau, M Sentis, R Zwiggelaar, A K Davison, R Marti, 2018. Automated Breast Ultrasound Lesions Detection Using Convolutional Neural Networks. [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/28796627/
[Accessed July 2022].

40. M Huh, P Agrawal, A A. Efros, 2016. What makes ImageNet good for transfer learning?. [Online]
Available at: https://arxiv.org/abs/1608.08614
[Accessed July 2022].

41. Mayank, M., 2020. Convolutional Neural Networks, Explained. [Online]
Available at: https://towardsdatascience.com/convolutional-neural-networks-explained-9cc5188c4939
[Accessed June 2022].

42. Md Z Alom, T M. Taha, C Yakopcic, S Westberg, P Sidike, M S Nasrin, B C Esesn, A A S. Awwal, V K. Asari, 2018. The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches. [Online]
Available at: https://arxiv.org/abs/1803.01164
[Accessed July 2022].

43. Milecia, M., 2021. What Is a Convolutional Neural Network? A Beginner's Tutorial for Machine Learning and Deep Learning. [Online]
Available at: https://www.freecodecamp.org/news/convolutional-neural-network-tutorial-for-beginners/
[Accessed June 2022].

44. Mutar, M.T.;Goyani, M.S.;Had, A.M.;Mahmood, A.S., 2019. Pattern of Presentation of Patients with Breast Cancer in Iraq in 2018: A Cross-Sectional Study.. [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/31721627/
[Accessed June 2021].

45. Mwiti, D., 2022. Transfer Learning Guide: A Practical Tutorial With Examples for Images and Text in Keras. [Online]
Available at: https://neptune.ai/blog/transfer-learning-guide-examples-for-images-and-text-in-keras
[Accessed June 2022].

46. Myra K Feldman, Sanjeev Katyal, Margaret S Blackwood., 2009. US artifacts Radiographics. [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/19605664/
[Accessed July 2022].

47. N Best, J Ott, E J. Linstead , 2020. Exploring the efficacy of transfer learning in mining image-based software artifacts. [Online]
Available at: https://journalofbigdata.springeropen.com/articles/10.1186/s40537-020-00335-4
[Accessed MJuly 2022].

48. Nagwa Dongola, MD, 2020. Mammography in Breast Cancer. [Online]
Available at: https://emedicine.medscape.com/article/346529-

overview?src=wnl_edit_specol&uac=159555SR.
[Accessed August 2022].

49. Nima T, Laura J, Q Li, Jeffrey N C, Z Wu, X Ding, 2020. Embracing imperfect datasets: A review of deep learning solutions for medical image segmentation. Med Image Anal.

50. Oscar Day, Taghi M. Khoshgoftaar , 2017. A survey on heterogeneous transfer learning. Journal of Big Data.

51. PyTorch, 2017. How the pytorch freeze network in some layers, only the rest of the training?. [Online]
Available at: https://discuss.pytorch.org/t/how-the-pytorch-freeze-network-in-some-layers-only-the-rest-of-the-training/7088
[Accessed 2022].

52. Qinghua Huang, Fan Zhang, Xuelong Li. , 2018. Machine Learning in Ultrasound Computer-Aided Diagnostic Systems: A Survey. [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/29687000/
[Accessed July 2022].

53. R N D'souza, P-Y Huang, F-C Yeh, 2020. Structural Analysis and Optimization of Convolutional Neural Networks with a Small Sample Size. [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/31965034/
[Accessed 2022].

54. Raghuveer., L. V. S., 2019. A review on Practical Diagnostic of Tomato Plant Diseases. [Online]
Available at:
https://www.researchgate.net/publication/332570921_A_review_on_Practical_Diagnostic_of_Tomato_Plant_Diseases
[Accessed June 2022].

55. S Azizi, P Mousavi, P Yan , Amir T, J T Kwak , S Xu , Baris T , P Choyke, P Pinto, B Wood, Purang A, 2017. Transfer learning from RF to B-mode temporal enhanced ultrasound features for prostate cancer detection. Int J Comput Assist Radiol Surg.

56. S Imai, S Kawai, H Nobuhara , 2020. Stepwise PathNet: a layer-by-layer knowledge-selection-based transfer learning algorithm. [Online]
Available at: https://www.nature.com/articles/s41598-020-64165-3
[Accessed July 2022].

57. Sheetal, S., 2017. Artificial Neural Network (ANN) in Machine Learning. [Online]
Available at: https://www.datasciencecentral.com/artificial-neural-network-ann-in-machine-learning/#:~:text=Artificial%20Neural%20networks%20(ANN)%20or,an%20animal's%20central%20nervous%20systems.
[Accessed June 2022].

58. Shengfeng L a, Wang Y a, Xin Y b, Baiying L a,Li Liu a, Shawn X L a, Dong N a, Tianfu W., 2019. Deep Learning in Medical Ultrasound Analysis: A Review. Engineering, 5(2).

59. Silver, D L. Y Qiang,Li Lianghao, 2013. Lifelong Machine Learning Systems: Beyond Learning Algorithms. [Online]
Available at: https://repository.hkust.edu.hk/ir/Record/1783.1-55154
[Accessed July 2022].

60. Sinno J P; Q Yang., 2010. A Survey on Transfer Learning. IEEE Transactions on Knowledge and Data Engineering, 22(10), pp. 1345-1359.

61. Sloun R J v, Cohen R, Yonina C E., 2019. Deep learning in ultrasound imaging. [Online]
Available at: https://arxiv.org/abs/1907.02994
[Accessed July 2022].

62. Smith, N. & Webb, A., 2010. Introduction to Medical Imaging: Physics, Engineering and Clinical Applications. [Online]
Available at:
https://assets.cambridge.org/97805211/90657/frontmatter/9780521190657_frontmatter.pdf
[Accessed June 2021].

63. TensorFlow, n.d. Overfit and underfit. [Online]
Available at: https://www.tensorflow.org/tutorials/keras/overfit_and_underfit
[Accessed July 2022].

64. Transfer Learning (C3W2L07). 2017. [Film] Directed by DeepLearningAI. s.l.: YouTube.

65. Vahab Khoshdel, Ahmed Ashraf, Joe LoVetri, 2019. Enhancement of Multimodal Microwave-Ultrasound Breast Imaging Using a Deep-Learning Technique. [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/31546925/
[Accessed July 2022].

66. Veronika C, M de Bruijne, J P W Pluim , 2019. Not-so-supervised: A survey of semi-supervised, multi-instance, and transfer learning in medical image analysis. Med Image Anal.

67. W Li, S Gu, X Zhang, Tao C, 2020. Transfer learning for process fault diagnosis: Knowledge transfer from simulation to physical processes. Computers & Chemical Engineering, Volume 139.

68. Y Yan, W Ren, X Cao, 2019. Recolored Image Detection via a Deep Discriminative Model. IEEE Transactions on Information Forensics and Security.

69. Z Li, D Hoiem., 2017. Learning without Forgetting. [Online]
Available at: https://ieeexplore.ieee.org/ielaam/34/8520726/8107520-aam.pdf
[Accessed July 2022].

70. Z Q Zhao, P Zheng, S T Xu, X Wu, 2019. Object Detection With Deep Learning: A Review. [Online]
Available at: https://pubmed.ncbi.nlm.nih.gov/30703038/
[Accessed July 2022].

71. Zhiyuan Chen, Bing Liu, 2016. Lifelong Machine Learning Sys Lec on AI AND MC. [Online]
Available at: https://www.cs.uic.edu/~liub/2016-Lifelong-Machine-Learning-final.pdf
[Accessed July 2022].

72. Zhou, Y., 2013. Ultrasound Diagnosis of Breast Cancer. Journal of Medical Imaging and Health Informatics, Volume 3.