BREAST CANCER CLASSIFICATION AND LOCATION DETECTION USING DEEP LEARNING METHODS

---***---

Abstract: - Early detection and classification of breast cancer can reduce mortality and provide the best possible treatment. Deep learning is used to overcome the shortcomings of machine learning. This study uses three datasets: ultrasound, histopathological, and mammography images to determine if a person has cancer and detect the location of cancer. Mask R-CNN segments and Patch detection algorithms are used to predict if the cancer is malignant or benign and locate the cancer. This approach outperforms other common methods in automatic classification of breast cancer images.

Key words: Breast Cancer, Ultrasound, Mammography, Histopathology, Classification, Location Detection, Mask R-CNN, Patch detection

1. Introduction

Breast cancer, the second most prevalent cause of mortality for women, is a common tumor that frequently affects females. Breast cancer has become increasingly common worldwide over time, and each year, more instances are reported. Early detection increases the likelihood of effective treatment and survival, but its diagnosis takes time and usually necessitates consensus amongst pathologists.

Breast cancer, the second most prevalent cause of mortality for women, is a common tumor that frequently affects females. Breast cancer has become increasingly common worldwide over time, and each year, more instances are reported. Early detection increases the likelihood of effective treatment and survival, but its diagnosis takes time and usually necessitates consensus amongst pathologists. Systems for computer-aided diagnosis (CAD) can increase the precision of diagnoses. There are two types of breast cancer: benign (not dangerous) and malignant (threatening). The prognosis for breast cancer depends heavily on the early discovery of tumors and the ability to differentiate between benign images and malignant images.

Detection of breast cancer in its early stages using image processing techniques includes four parts. In the first part the digital images (mammographies) are pre-processed to remove any kind of noise. Then in the second part the images undergo the segmentation process to enhance the tumor part. After this, in the third part, the important features in the segmented images are extracted. Finally, in the fourth part, with the help of the extracted features, the images are classified into normal, benign or malignant. Here, 'normal' represents the breast with no tumor, 'benign' represents the breast with non-cancerous tumor and 'malignant' represents breast with cancerous tumor.

Mammography is a highly well-liked and commonly utilised breast cancer screening approach since it is the only imaging modality that has been shown to significantly lower breast cancer mortality. In this study, we also classified breast cancer using mammography, ultrasound, and histopathology images. Clinicians have utilized CAD systems to help them interpret medical imaging to better diagnose illness and its crucial function is feature extraction. Since they are rigid, traditional feature extraction methods have drawbacks.

In the present-day scenario, to observe breast cancer mammograms are used and they are known to be the most effective scanning technique. In this paper the detection of cancer cells is done by machine learning technique. A method of improving a picture or extracting important information from it involves converting an image into a digital format and performing operations on it. An image is used as the input and features or qualities of the picture may be output in this sort of signal processing. An image processing system typically treats pictures as two-dimensional signals and processes them using specified signal processing techniques.

The following are the top three image processing techniques:

- Ultrasound
- Import a picture using digital or optical scanning.
- Picture analysis and processing, including data compression, picture magnification, and looking for patterns in images that are imperceptible to the human eye, such as satellite photographs.
- The final stage, output, is when the outcome may be altered using a report based on a picture or image analysis.

The first stage is to assess and represent the images. The image representation technique will address fundamental issues such as the ineffectiveness of capturing textural information and the limited capacity for feature classification, which results in poor retrieval performance. Similarity estimation is a major problem in content-based image retrieval and has a bigger impact on retrieval accuracy and retrieval time. The project seeks to answer problems such as "Which similarity measure is appropriate for a particular feature type and how to reduce the similarity calculation computation?" and "Is the texture function more representative and discriminatory in describing the given query's mammogram?".

Recent years have seen the introduction of novel techniques for breast cancer diagnosis. Deep learning (DL), a branch of machine learning and artificial intelligence that focuses on the intricate structure of image attributes, is self-learning. To supplement the characteristics that are derived from the data using DL approaches, some fresh models are used. Artificial intelligence is used in machine learning, which enables computers to automatically learn from their experiences and get better over time. Machine learning's fundamental principle is to develop algorithms that take input data and use statistical analysis to predict outcomes, updating outputs as new information becomes available. In order to find patterns in the data and improve conclusions based on more instances, learning starts with observations or data, such as examples, first hand experience, or teaching. The basic objective is for computers to learn on their own without assistance from humans and to change their behaviour as a result.

For classification, we have used Convolutional Neural Network along with DenseNet169, DenseNet121, ResNet101 in our model. Comparison is done between Densenet and ResNet101. After the classification is done, to identify distinct items in a picture and create a bounding box around a particular object, techniques called object detection and segmentation are used. Such techniques for object recognition and segmentation are Mask R-CNN and patch detection.

The device determines the required region properties, such as size, Euler number, and so on, by conducting appropriate morphological operations, and displays the detected border image beside the tumor area. Mask R-CNN has the ability to construct a bounding box for the target object as well as further mark and categorize whether the pixels in the bounding box belong to the item or not, allowing for the identification of the object, marking of the object's border, and detection of critical spots. Patch detection is a method to detect metastatic cancer in tiny image patches acquired from bigger digital pathology images. Patch detection provides binary target visualisation per tissue slice.

2. Related Work

A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. A DL method based on TL was proposed by Khan S, Islam N, Jan Z, Din IU, Rodrigues JJPC [1]. They employed three pretrained models to distinguish between cancerous and benign cells: GoogLeNet, VGGNet, and ResNet. They test the effectiveness of their strategy using cytology pictures.

Deep convolutional neural networks for breast cancer screening. To prevent overfitting when working with tiny datasets, the authors Chougrad H, Zouaki H, Alheyane O [2], introduced a deep-CNN model that included TL. They used the DDSM, INbreast, BCDR, and MIAS datasets to assess the performance of the proposed model. The DDSM dataset demonstrated a 0.98 area under the curve and 97.35% accuracy (AUC). In contrast to the BCDR database, which produced 96.67% accuracy and 0.96 AUC, the INbreast dataset produced 95.5% accuracy.

Breast cancer classification from histopathological images using patch-based deep learning modeling. The researchers Hirra, I.; Ahmad, M.; Hussain, A.; Ashraf, M.U.; Saeed, I.A.; Qadri, S.F.; Alghamdi, A.M.; Alfakeeh, A.S [3], suggested a patch-based Deep Learning (DL) technique called Pa-DBN-BC for categorising and identifying BC on histopathology pictures (DBN). Via supervised finetuning and unsupervised pre-training stages, the feature is retrieved. The network automatically extracts features from picture patches. Logistic Regression is used.

A new transfer learning based approach to magnification dependent and independent classification of breast cancer in histopathological images. For the purpose of categorising pictures of histological breast cancer, the researchers Boumaraf S, Liu X, Zheng Z, Ma X, Ferkous C [4], presented the TL model. The ResNet-18 model served as the foundation for the authors' block-wise fine-tuning approach. Using global contrast normalisation and data augmentation methods, the model's performance was enhanced.

Improved breast cancer classification through combining graph convolutional network and convolutional neural network. A method known as BDR-CNN-GCN, developed by the researchers Zhang Y-D, Satapathy SC, Guttery DS, Go´rriz JM, Wang S-H [5], combines a CNN with a graph-convolutional network (GCN). The batch normalisation and dropout layer were merged with an elementary eight-layer CNN. This model was coupled to a two-layer GCN to create the final BDR-CNN-GCN model. In the MIAS dataset, this model's performance was evaluated, and it scored 96.10% accuracy.

Breast cancer detection using deep convolutional neural networks and support vector machines [6]. The support vector machine (SVM) and AlexNet were used by the authors to boost classification accuracy while also increasing the number of input photos. Performance is assessed using the DDSM and CBIS-DDSM datasets. The method's accuracy is 71.01%, while applying SVM increases it to 87.2%.

Integrating spatial fuzzy clustering with level set methods for automated medical image segmentation. In this research, a brand-new fuzzy level set technique is put forth by Bing Lan Li et al. [7], to help with automated medical image segmentation. The initial segmentation by spatial fuzzy clustering, in which the centroid and the scope of each subclass are adaptively estimated in order to minimize a predefined cost function, can directly evolve into this process. The outcomes of fuzzy clustering are also used to estimate the Level Set evolution's governing parameters. For picture segmentation, level set approaches use dynamic variational bounds. Using geographical fuzzy clustering, the new fuzzy level set technique automates the level set segmentation's setup and parameter tuning. The outcomes attest to its viability for picture segmentation.

3. Problem Statement

One in eight breast cancers are missed by screening mammography. Breast density increases the chance of obtaining incorrectly negative findings in women. False-negative mammograms can make women feel insecure when they actually have breast cancer because they mislead them.

Despite the success CNN architectures have had in diagnosing breast cancer, there are still too many hyperparameters, which makes it difficult to get improved results.

Additionally, the current system merely categorized the inputs as positive or negative. The presence of the tumor has not been determined in what part of the body.

4. Methodology

The purpose of our work is to offer a breast cancer diagnostic model. The suggested model can identify the tumour if it is malignant and has a box-like form in addition to classifying the pictures as benign or malignant.

The process of finding cancer comprises mostly four steps: are shown in the following flow diagram:

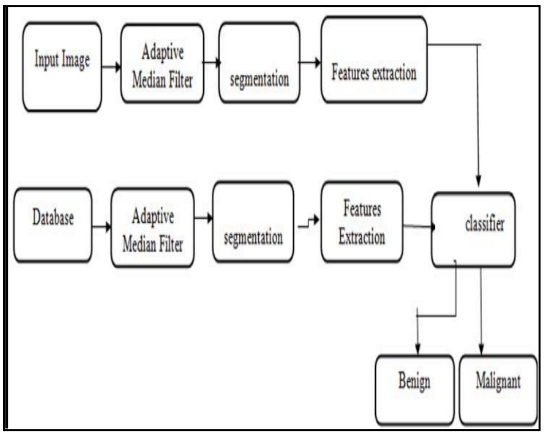


FIGURE 4.1 BLOCK DIAGRAM

- Phase 1: Pre Processing
- Phase 2: Image Processing
- Phase 3: Feature Extraction
- Phase 4: Classification

4.1 Dataset Collection

Three different types of datasets are used in this research:
- Ultrasound
- Mammography
- Histopathology

Breast ultrasound pictures from women aged 25 to 75 comprise the baseline data. This information was gathered in 2018. There are 600 female patients in all. There are 780 photos in the collection, with an average size of 500 by 500 pixels. The photos are made up of PNG files. The images taken in the real world are shown alongside the originals. The photographs are divided into three categories: benign, malignant, and normal.

Invasive ductal carcinoma (IDC) is the most frequent form of breast cancer. Pathologists usually focus on the regions containing the IDC when assessing the aggressiveness of a whole mount sample. One of the primary pre-processing stages for automated aggressiveness evaluation is determining the precise IDC points inside a whole mount slide. There were 277524 total photographs, 198738 images of people without cancer, 78786 images of persons with cancer. 162 whole mount slide pictures of Breast Cancer (BCa) specimens that had been scanned at 40x made up the original dataset. 277,524 patches of size 50 50 were produced, of which 128 738 IDC negative patches and 78,786 IDC positive patches were retrieved.

The Digital Database for Screening Mammography (DDSM) has been revised and standardised as the CBIS-DDSM (Curated Breast Imaging Subset of DDSM) [8]. 2,620 digitalized film mammography studies may be found in the DDSM database. It includes illustrations of healthy, malignant, and benign disorders with validated pathology data.
Because of the scale of the database and the ground truth verification, the DDSM is a useful tool in the design and testing of decision support systems.
An experienced mammographer selected and put together a subset of the DDDSM data for the CBIS-DDSM collection. After being decompressed, the pictures were changed to DICOM format.

4.2 Training and Testing the Dataset

The dataset has been divided into two halves once the hybrid machine learning algorithms have been applied, with the first half being used for training and the second for testing. The complete dataset is separated into the following categories: 25% is allotted to the training set, while the remaining 75% goes to the testing set. With the use of the training set, the model should first be trained, and then it should be confirmed using the test set. The trained model file is created following the completion of the data training, and the testing data are then given to the trained model file. A trained model's prediction is authorised or authenticated through validation in machine learning or deep learning.

In contrast, evaluation in machine learning refers to the examination or testing of the complete machine learning model and its effectiveness under diverse conditions. It entails evaluating the performance of deep learning algorithms, the machine learning model training process, and the precision of the predictions made in various scenarios.

4.3 Algorithms Used

4.3.1 Classification

Classification is the process of predicting the class of a group of data items. It is used in supervised learning, target marketing, medical diagnosis, and credit clearing. Image classification is used to identify and characterise characteristics in a photograph, such as the item or sort of land cover it reflects. Classification is an important step in digital image analysis, used to reduce over grinding in mills. It can be done by lazy learners or eager learners, who construct a classification model before obtaining data for classification. Examples include Artificial Neural Networks, Decision Trees, and Naive Bayes.

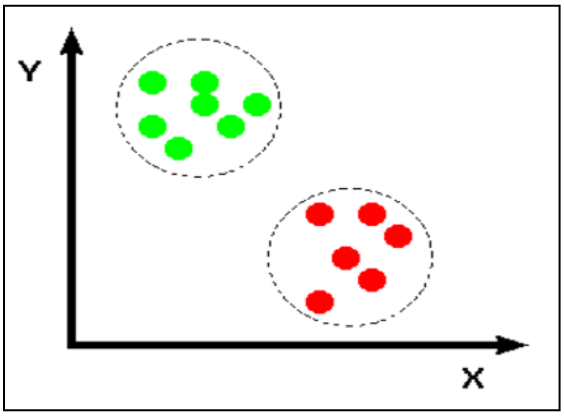


FIGURE 4.1 CLASSIFICATION

• LAZY LEARNERS: The training data is merely stored by lazy learners, who then wait for the testing data to emerge. When this occurs, classification is carried out using the stored training data's most relevant information. Lazy learners spend more time predicting than motivated learners do throughout instruction. As an example, using cases as a guide.

• EAGER LEARNERS: Using the available training data, eager learners construct a classification model prior to obtaining data for classification. It must be able to commit to a single hypothesis that encompasses every scenario. The model architecture causes enthusiastic learners to train slowly and anticipate slowly. Example: Artificial Neural Networks, Decision Trees, and Naive Bayes In this study, eager learners—also known as gradient boosting—were utilized.

4.3.2 Gradient Boosting

Gradient boosting classifiers are a set of machine learning approaches that merge numerous failed learning models into a powerful prediction model [9]. Decision trees are widely used in gradient boosting. Boosting is a method of improving the performance of difficult students. In boosting, each new tree is a fit on a modified version of the starting data set. As it constructs a decision tree, the method begins by assigning equal weight to each observation. Following the inspection of the first tree, the weights of observations that are difficult to classify are increased, while the weights of data that are easy to recognise are dropped. This weighted data is then used to build the second tree. The purpose here is to improve the prospects of the first tree.So, Tree 1 plus Tree 2 is our new model. After determining the classification error from this new 2-tree ensemble model, a third tree is created to predict the rectified residuals. It takes a specific number of repetitions to complete this procedure. We may find data that the previous trees failed to accurately classify. Therefore, the weighted total of the predictions given by the previous tree models constitutes the predictions of the final ensemble model.

4.3.3 Bagging

With replacement, a portion of data from a training sample is chosen at random. The decision trees for each group of data are then trained. We end up having an ensemble of multiple models as a consequence, and their average is significantly more reliable than a single decision-tree, which is significantly more reliable in predictive analysis. Bagging's expansion, Random Forest.

4.3.4 Boosting

• The term "boosting" describes a group of learners who turn weak learners into strong learners.
• It gradually learns from the errors produced by an earlier random sample, in this case a tree.
• Instruction is given to the weaker students in phases, with each step intended to improve the one before it. Early learners first checked the data for errors before fitting basic models to it.
• All of the weak learners with their higher error accuracy are somehow merged to produce a strong classifier with a better accuracy.
• When a hypothesis classifies an input wrongly, its weight is increased, increasing the chance that the next hypothesis will classify it correctly.
• By combining the whole collection at the end, weak learners are turned into models with greater performance.

4.3.4 SVM Algorithm

SVM, or Support Vector Machine, is one of the most extensively used supervised learning algorithms for classification and regression issues. The majority of its applications, however, are machine learning classification tasks.

The SVM technique seeks to create the optimum line or decision boundary that can partition n-dimensional space into classes in order to quickly classify fresh data points in the future. This ideal choice's boundary is known as a hyperplane.

SVM is used to choose the hyperplane's extreme vectors and points. The support vectors used in the SVM approach are used to represent these extreme circumstances.

4.3.5 Random Forest

Data scientists routinely employ Random Forest, one of the most prominent and widely used algorithms. Random forest is a supervised machine learning method that is effective for classification and regression issues. It produces decision trees from a large number of samples by utilising the average of the samples for regression and the great majority of the data for categorization.

One of the Random Forest Algorithm's key properties is its ability to handle data sets with both continuous variables, as in regression, and categorical variables, as in classification. It performs better in classification and regression tasks.The commonly used machine learning technique known as random forest was created by combining the results of many decision trees to get at a single conclusion was developed by Leo Breiman and Cutler.Its adaptability and usefulness, which can handle classification and regression issues, are what spur its widespread usage.

4.3.6 KNN Algorithm

K-nearest neighbours, often known as KNN or k-NN, is a supervised learning classifier that predicts or categorises how a single data point will be categorised. Although it may be applied to classification or regression problems, because it is based on the idea that equivalent points can be found close together, it is most commonly applied to classification problems. One of the most fundamental supervised learning-based machine learning algorithms is K-Nearest Neighbour. The KNN approach assigns the new instance to the category that is most similar to the present categories, assuming that the new example and the preceding instances are identical.

4.3.7 XG Boost

A distributed, scalable gradient-boosted decision tree, XGBoost is referred to as Extreme Gradient Boosting or XGBoost. It features parallel tree boosting and is the best machine learning application for regression, classification, and ranking problems. Understanding the machine learning theories and methods that support ensemble learning, gradient boosting, decision trees, and supervised machine learning is necessary for using XGBoost. A model is trained in supervised machine learning using algorithms to find patterns in a collection of features and labels. The labels on the characteristics of a fresh dataset are then predicted using the model.

4.3.8 LSTM

One of the most complicated subfields in deep learning is LSTM. It is difficult to understand LSTM as a concept. It deals with algorithms that try to imitate how the brain works and discover the hidden connections in the sequential data given.

4.3.9 BI LSTM

Using the bidirectional long-short term memory (bi-lstm) technique, any neural network may access sequence information in both directions—backwards (from future to past) or forwards (from past to present). A bidirectional LSTM, as opposed to a normal LSTM, allows input that flows in both directions.

4.3.10 DENSE NET

A specific kind of convolutional neural network called a DenseNet uses dense connections between layers using dense blocks to directly link all layers (with matching feature-map sizes). DenseNet is one of the most recent developments in visual object recognition neural networks. Although ResNet and DenseNet are quite similar, there are some significant differences. While DenseNet concatenates (+) the output of the previous layer with the output of the succeeding layer, ResNet uses an additive technique (+) that combines the output of the previous layer (identity) with the output of the next layer.

High-level neural networks have vanishing gradient problems, which is one of the main reasons DenseNet was created. To put it another way, the information disappears before it reaches its destination because of the greater separation between the input layer and the output layer. There are several variations of the DenseNet, including the DenseNet-121, DenseNet-160, and DenseNet-201.

4.3.8 RES NET

Applications for computer vision use the Residual Network (ResNet) deep learning model. It has a convolutional neural network (CNN) architecture that supports many convolutional layers—hundreds or even thousands. The limited layer count of the earlier CNN systems may have an impact on performance.

However, the "vanishing gradient" issue cropped up as more layers were applied. In order to train neural networks, backpropagation minimises the loss function and uses gradient descent to identify the weights that minimise it. The gradient will ultimately become so thin that it "disappears" if there are too many levels.

Performance will also get saturated or degrade with each additional layer. In ResNet, the concept of "skip connections" is an innovative solution to the vanishing gradient problem. ResNet stacks a large number of identity mappings (convolutional layers that are initially inactive), skips over those levels, and utilises the activations from the layer before.

Compressing the network's layers by bypassing them may speed up initial training. Once all layers are extended during retraining, the network's remaining components, also known as the residual portions, are free to explore more of the input picture's feature space. The majority of ResNet models skip two or three layers at once, interspersed with nonlinearity and batch normalisation. HighwayNets, more complex ResNet designs, may learn "skip weights," which dynamically decide how many layers to skip.

4.4 Location Detection

4.4.1 MASK R-CNN

Mask R-CNN is a kind of region-based convolutional neural network that provides bounding boxes and a confidence score for each object's class label. The quicker R-CNN object detection architecture is built upon by Mask R-CNNs, which do more than only forecast the class and not just the object's enclosing box coordinates but also its mask.The most sophisticated Convolutional Neural Network (CNN) for image segmentation is Mask R-CNN, sometimes referred to as Mask RCNN.

Mask R-CNN was built on top of Faster R-CNN, a convolutional neural network based on regions. Understanding how Mask R-CNN works requires an understanding of picture segmentation. The computer vision issue A digital picture is divided into segments (groups of pixels sometimes referred to as image objects) using the image segmentation technique. Segmentation is used to locate objects and boundaries (such as lines, curves, etc.). Semantic segmentation and Instance segmentation are the two main categories covered by Mask R-CNN for image segmentation. Mask R-CNN was created with Faster R-CNN. Faster R-CNN offers a class name and bounding-box offset for each candidate item; however, Mask R-CNN adds a third branch, which returns the object mask. Because it varies from the class and box outputs, the additional mask output necessitates the extraction of a significantly more accurate spatial arrangement of an item. Mask R-CNN is created by modifying Faster R-CNN and simultaneously adding a branch for bounding box detection and a branch for predicting an object mask (Region of Interest). Mask R-CNN aids in the identification of the tumour in the image collection by emphasising the damaged region.

4.4.2 Patch Detection

Patch detection is a method to detect metastatic cancer in tiny image patches acquired from bigger digital pathology images. Patch detection provides binary target visualisation per tissue slice. The damaged region of a cancerous breast is highlighted during patch detection, which helps locate the cancer's exact location.

5. Implementation

5.1 Ultrasound Images

Sound waves are used during a technique known as a breast ultrasound to see within your breasts. It may be used by your healthcare provider to detect breast problems. It also allows your physician to evaluate how effectively the blood is reaching certain areas of your breasts. This test is usually done when a change has been noted on a mammogram or when a change is felt but not evident on an ultrasound.

To create the photographs of your breasts, the medical professional sweeps a transducer—a tool that resembles a wand—across your skin. Your breast tissue reflects the sound waves that the transducer emits. You can't hear it because the sound waves are too high-pitched. The transducer then captures the sound waves that have bounced. These are used to create images of the interior of your breasts. Due to the fact that ultrasound does not involve radiation, it is safe to use during pregnancy. Due to the fact that it doesn't use dye, it is also safe for persons who are allergic to contrast dye.

We use several classification algorithms such as CNN, ResNet, VGG, and DenseNets to first classify the input images as Malignant or Benign. The achieved results are discussed in the Results sections. Once the Images are classified , we then use the MASK-RCNN algorithm to detect the location of the cancer.

5.2 Histopathology Images

Histopathology pictures are obtained by a procedure known as a biopsy. After the photos are acquired, data augmentation is done. The deep learning community has recently paid a lot of attention to data augmentation. Neural networks need a lot of training data to function well, however the data sets currently accessible in the medical sector are from low-resource regions.

Therefore, to increase the variety of the fundamental dataset, a data augmentation approach must be applied. After that, the CNN Algorithm is trained to determine whether or not a picture includes cancer. After classification, we use the Patch Detection Algorithm to find the cancer cell.

5.3 Mammography Images

The patient's breast is first placed on a flat plate during a mammography, and a parallel plate is used to compress it. A short burst of radiation from the x-ray machine travels through the breast and is picked up on the other side. In contrast to a photographic film plate, which captures the x-ray image on film, a solid-state detector may be able to produce digital images by delivering electrical signals to a computer. Mammograms are the medical word for the images.
Dense tissue sections, such as connective and glandular tissue or tumours, appear whiter on a grey background on a film mammography, whereas low density tissues, such as fat, seem translucent (i.e., darker shades of grey).

6. Experimental Setup

In the following paragraphs, we will discuss the technique for setting up the assessment of frameworks, as well as the datasets and frameworks that were utilized.

6.1 Tools Used

• Google Colaboratory: As an open-source integrated development environment (IDE), google colaboratory is being utilized in this project. We are able to train our machine learning and deep learning models on CPUs, GPUs, and TPUs by using the Google Collaboratory Environment, a free online cloud-based Jupyter notebook. It offers us a reasonably excellent GPU that we can use for twelve hours straight without incurring any costs.This should be sufficient to fulfill the computation requirements of the vast majority of data science professionals.

Google colaboratory provides us with three distinct sorts of run-times for our notebooks, which are as follows:
• Central Processing Units.
• Graphics Processing Units.
• Tensor processing Units.

Because of colaboratory, we have unrestricted access to a period of execution time equal to 12 hours. After that, the entire virtual machine will be wiped clean, and we will have to start the process all over again from the beginning. We are able to simultaneously execute several instances of the CPU, GPU, and TPU; however, the resources associated with these components are shared among these instances.

Through the use of colaboratory notebooks, we are able to incorporate not only rich text but also executable code into a single document, in addition to graphics, HTML, LaTeX, and other formats. When we make our own colaboratory notebooks, copies of them are kept in the Google Drive account associated with that account. We are able to quickly share our colaboratory notebooks with other people, such as coworkers or acquaintances. This gives other people the ability to remark on or even modify your notebooks.

Using google colaboratory , we are able to carry out the tasks listed below as developers:
• Write Python code and execute it.
• Make notebooks, upload them, and share them.
• Google Drive and GitHub notebooks may be imported, saved, and published.
• Save, import, and export outside datasets.
• Sync up OpenCV, PyTorch, TensorFlow, and Keras.
• Free GPU resources are available on a cloud service.

7. Result and Analysis

7.1 Introduction

In this chapter, the practical outcomes that were acquired while the project was being put into action are discussed.

7.2 Ultrasound

Using the Ultrasound Dataset, we first perform some Image preprocessing techniques to resize and correct the images.

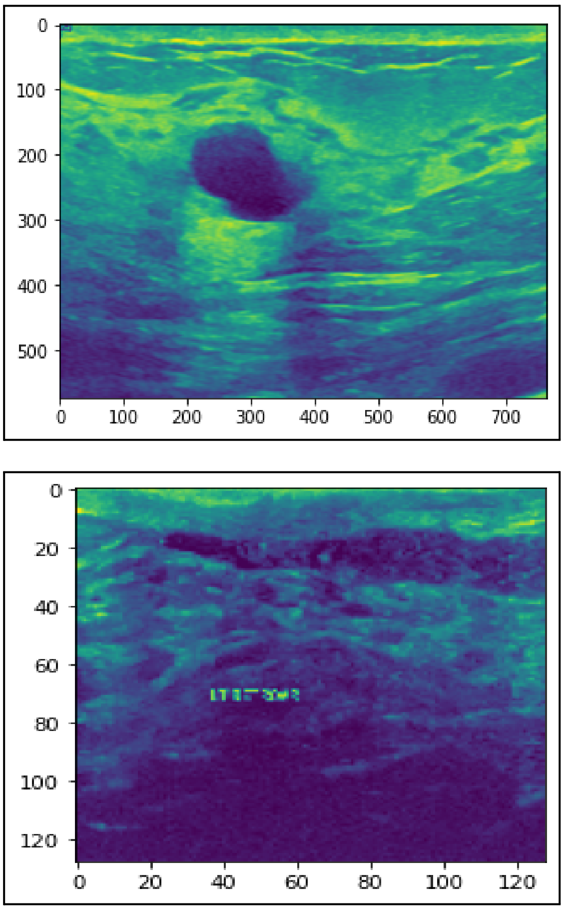


FIGURE 7.1 ORIGINAL IMAGE VS RESIZED IMAGE

7.2.1 CNN Using K-Fold Cross Validation

A CNN is created by carefully stacking a number of these layers. Here, we define a high-level CNN to determine the output classification probabilities after applying a convolution layer, an activation layer, a fully connected layer, and a convolution layer to an input to acquire the classification probabilities. The following diagram displays the defined CNN layers. Adam Optimizer [11] and Relu were used as the activation function.

We apply the K-Fold cross validation approach after the CNN Layers are defined. To evaluate machine learning models on a tiny data sample, a resampling approach known as cross-validation is utilised. The procedure has a single parameter, k, which specifies how many groups should be formed from a given data sample. As a result, the procedure is commonly known as k-fold cross-validation. When a specific k value is chosen, it may be substituted for k in the model's reference, as when k=10 was used to represent 10-fold cross-validation. Cross-validation is commonly used in applied machine learning to assess how well a machine learning model performs on untrained data. The Result Obtained is shown in the Figure below:

```
------------------------------------------------------------------------
Score per fold
------------------------------------------------------------------------
> Fold 1 - Loss: 1.1053917407989502 - Accuracy: 77.21518874168396%
------------------------------------------------------------------------
> Fold 2 - Loss: 0.9974886775016785 - Accuracy: 81.64557218551636%
------------------------------------------------------------------------
> Fold 3 - Loss: 0.6004195213317871 - Accuracy: 84.8101258277893%
------------------------------------------------------------------------
> Fold 4 - Loss: 1.1476929187774658 - Accuracy: 79.7468364238739%
------------------------------------------------------------------------
> Fold 5 - Loss: 0.8113642930984497 - Accuracy: 81.64557218551636%
------------------------------------------------------------------------
> Fold 6 - Loss: 0.5672875046730042 - Accuracy: 82.27847814559937%
------------------------------------------------------------------------
> Fold 7 - Loss: 1.284202218055725 - Accuracy: 85.44303774833679%
------------------------------------------------------------------------
> Fold 8 - Loss: 1.277942180633545 - Accuracy: 79.11392450332642%
------------------------------------------------------------------------
> Fold 9 - Loss: 0.6631337404251099 - Accuracy: 76.43312215805054%
------------------------------------------------------------------------
> Fold 10 - Loss: 0.5421966314315796 - Accuracy: 82.16560482978821%
------------------------------------------------------------------------
Average scores for all folds:
> Accuracy: 81.04974627494812 (+- 2.7994362545250553)
> Loss: 0.8997119426727295
------------------------------------------------------------------------
```

FIGURE 7.2 RESULT OF CNN USING K-FOLD USING CROSS VALIDATION

As we can see, the average accuracy attained is 91.04%, while the average loss is 0.899.

7.2.2 RESNET 101

Here we define a Resnet101 model. The defined model is shown in the image below:

Model: "sequential_1"

Layer (type)	Output Shape	Param #
resnet101 (Functional)	(None, 4, 4, 2048)	42658176
max_pooling2d_1 (MaxPooling2	(None, 2, 2, 2048)	0
flatten_1 (Flatten)	(None, 8192)	0
batch_normalization_1 (Batch	(None, 8192)	32768
dense_5 (Dense)	(None, 512)	4194816
dropout_3 (Dropout)	(None, 512)	0
dense_6 (Dense)	(None, 256)	131328
dropout_4 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 128)	32896
dropout_5 (Dropout)	(None, 128)	0
dense_8 (Dense)	(None, 64)	8256
dense_9 (Dense)	(None, 3)	195

Total params: 47,058,435
Trainable params: 4,383,875
Non-trainable params: 42,674,560

FIGURE 7.3 RESNET101

By training our dataset in the defined model we get the following results:
Train accuracy = 0.9798657894134521
Validation accuracy = 0.8656716346740723
Test accuracy = 0.8888888955116272
f1_measure = 0.8893466348937087
KAPPA = 0.8100412139378044
roc_area = 0.9742590133570231

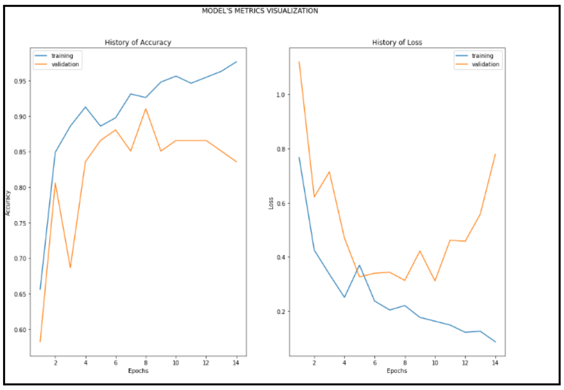


FIGURE 7.4 VISUALIZATION OF RESNET101

7.2.3 VGG 16

The VGG 16 model is defined as shown in the image below:

Model: "sequential_1"

Layer (type)	Output Shape	Param #
vgg16 (Functional)	(None, 4, 4, 512)	14714688
flatten_1 (Flatten)	(None, 8192)	0
batch_normalization_1 (Batch	(None, 8192)	32768
dense_5 (Dense)	(None, 512)	4194816
dropout_4 (Dropout)	(None, 512)	0
dense_6 (Dense)	(None, 256)	131328
dropout_5 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 128)	32896
dropout_6 (Dropout)	(None, 128)	0
dense_8 (Dense)	(None, 64)	8256
dropout_7 (Dropout)	(None, 64)	0
dense_9 (Dense)	(None, 3)	195

Total params: 19,114,947
Trainable params: 4,383,875
Non-trainable params: 14,731,072

FIGURE 7.5 VGG16

By training our dataset in the defined model we get the following results:
Train accuracy = 0.986577153205871
Validation accuracy = 0.9552238583564758
Test accuracy = 0.9743589758872986
f1_measure = 0.9744646080523943
KAPPA = 0.9561633570625703
roc_area = 0.9957788020980289

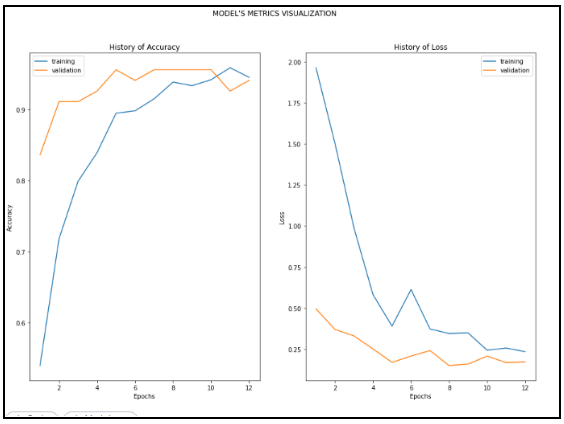


FIGURE 7.6 VISUALIZATION OF VGG16

7.2.4 VGG 19

The VGG 19 model is defined as shown in the image below.

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
vgg19 (Functional)           (None, 4, 4, 512)         20024384
_________________________________________________________________
flatten_3 (Flatten)          (None, 8192)              0
_________________________________________________________________
batch_normalization_3 (Batch (None, 8192)              32768
_________________________________________________________________
dense_13 (Dense)             (None, 128)               1048704
_________________________________________________________________
dropout_10 (Dropout)         (None, 128)               0
_________________________________________________________________
dense_14 (Dense)             (None, 32)                4128
_________________________________________________________________
dropout_11 (Dropout)         (None, 32)                0
_________________________________________________________________
dense_15 (Dense)             (None, 3)                 99
=================================================================
Total params: 21,110,083
Trainable params: 1,069,315
Non-trainable params: 20,040,768
```

FIGURE 7.7 VGG19

By training our dataset in the defined model we get the following results:
Train accuracy = 0.9882550239562988
Validation accuracy = 0.9402984976768494
Test accuracy = 0.9658119678497314
f1_measure = 0.9660831689677843
KAPPA = 0.941805521014673
roc_area = 0.9984315746197393

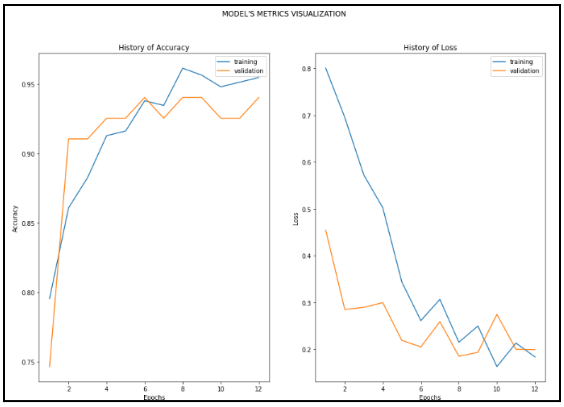


FIGURE 7.8 VISUALIZATION OF VGG19

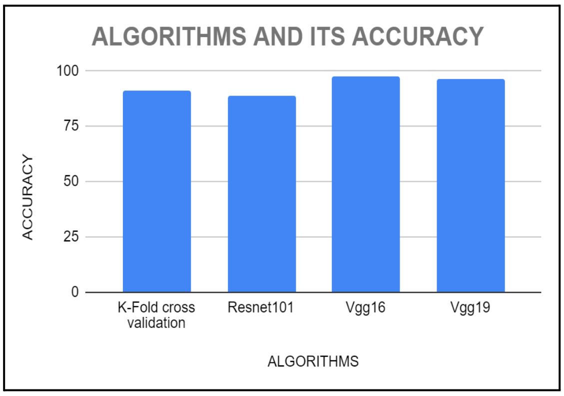


FIGURE 7.9 COMPARISON OF ULTRASOUND ALGORITHMS

7.2.5 DENSENET 121

Using Transfer learning, we have used Dense 121 after training the images on the following algorithms:
SVM, Random Forest, AdaBoost, KNN, XGBoost, Bagging, ANN, LSTM, Bi-LSTM. The Results obtained are shown in the image below:

	classifier	train_accuracy	val_accuracy	test_accuracy	f1_measure
0	SVM	0.9832	0.9104	0.9829	0.9829
1	Random Forest	1.0000	0.9104	0.9915	0.9914
2	AdaBoost	0.9480	0.8507	0.9316	0.9326
3	KNN	0.9715	0.8806	0.9744	0.9742
4	XGBoost	1.0000	0.9104	1.0000	1.0000
5	Bagging	1.0000	0.8955	0.9915	0.9914
6	ANN	1.0000	0.9552	0.9744	0.9747
7	LSTM	0.9916	0.9552	0.9402	0.9415
8	Bi-LSTM	0.9849	0.9403	0.9573	0.9580

	kappa_score	recall	Precision
0	0.9706	0.9829	0.9829
1	0.9853	0.9915	0.9916
2	0.8846	0.9316	0.9364
3	0.9558	0.9744	0.9743
4	1.0000	1.0000	1.0000
5	0.9853	0.9915	0.9916
6	0.9565	0.9744	0.9766
7	0.9003	0.9402	0.9512
8	0.9282	0.9573	0.9632

FIGURE 7.10 DENSENET121

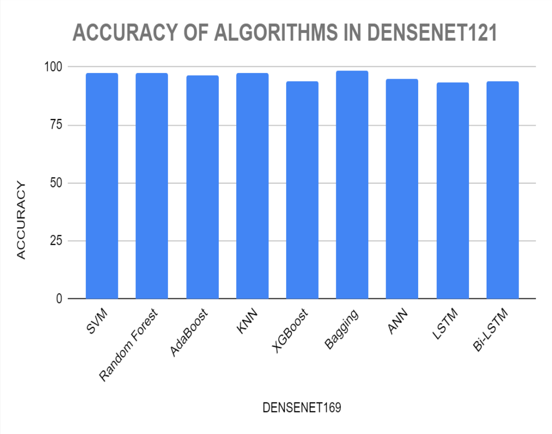


FIGURE 7.11 ACCURACY OF ALGORITHMS IN DENSENET121

7.2.6 DENSE NET 169

Using Transfer learning, we have used Dense 169 after training the images on the following algorithms:
SVM, Random Forest, AdaBoost, KNN, XGBoost, Bagging, ANN, LSTM, Bi-LSTM. The Results obtained are shown in the image below:

	classifier	train_accuracy	val_accuracy	test_accuracy	f1_measure	kappa_score	recall	Precision
0	SVM	0.9883	0.9254	0.9744	0.9745	0.9562	0.9744	0.9748
1	Random Forest	1.0000	0.9403	0.9744	0.9745	0.9562	0.9744	0.9748
2	AdaBoost	0.9597	0.8806	0.9658	0.9655	0.9408	0.9658	0.9660
3	KNN	0.9782	0.9104	0.9744	0.9745	0.9562	0.9744	0.9748
4	XGBoost	1.0000	0.9403	0.9402	0.9409	0.8986	0.9402	0.9431
5	Bagging	0.9966	0.9254	0.9829	0.9830	0.9709	0.9829	0.9839
6	ANN	1.0000	0.9552	0.9487	0.9495	0.9135	0.9487	0.9531
7	LSTM	0.9866	0.9701	0.9316	0.9333	0.8866	0.9316	0.9456
8	Bi-LSTM	0.9933	0.9701	0.9402	0.9412	0.8995	0.9402	0.9466

FIGURE 7.12 DENSENET169

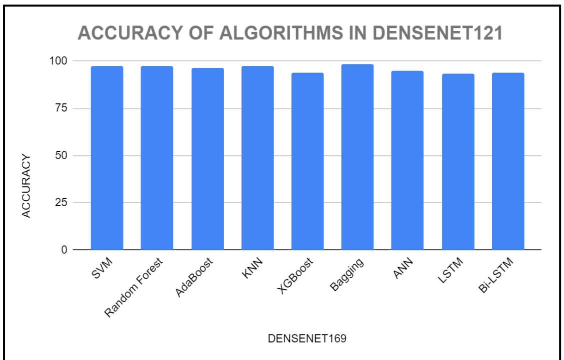


FIGURE 7.13 ACCURACY OF ALGORITHMS IN DENSENET169

We can plainly observe from the algorithms utilized that DenseNet Models have outperformed all other deep learning models.

We employ the MASK - RCNN algorithm to locate the tumor once the images have been classified as malignant or benign. So we install the model and import the MASK RCNN Model from torch. We train the model for 35 epochs and the results obtained are shown below.

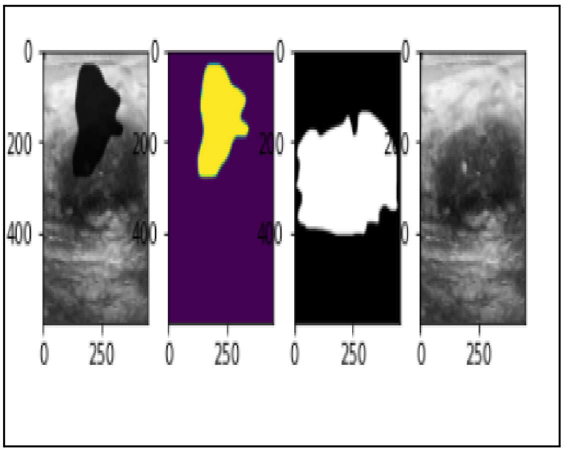


FIGURE 7.14 MASK R-CNN

The figure above shows that the patches are masked. In order to identify the malignant tumor, we next use the predicted output to draw a square over the mask.

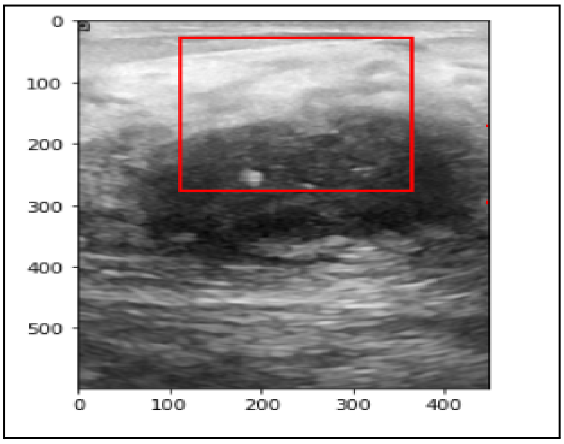


FIGURE 7.15 MARKED PATCHES

We can see the comparison between the actual mask and the predicted mask from the image below:

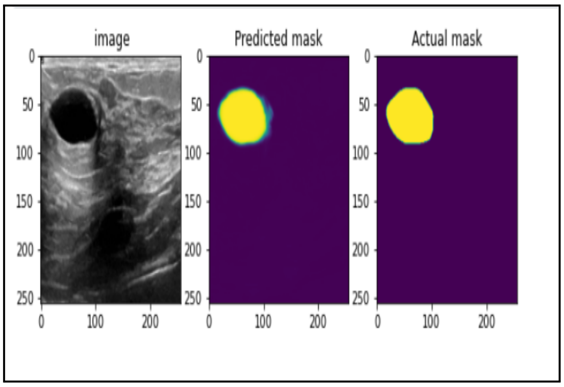


FIGURE 7.16 ACTUAL VS PREDICTED MASK

7.3 Histopathology

We use the Histopathology images dataset as shown below:

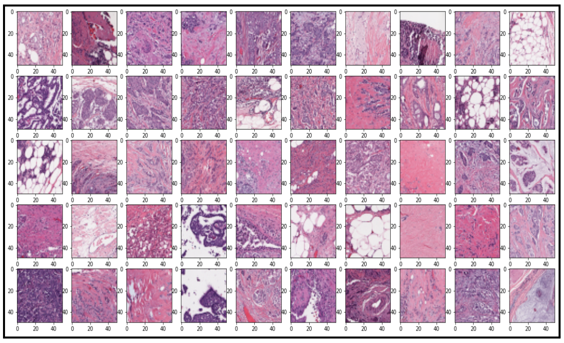


FIGURE 7.17 HISTOPATHOLOGY DATASET IMAGES OF CANCER PATCHES

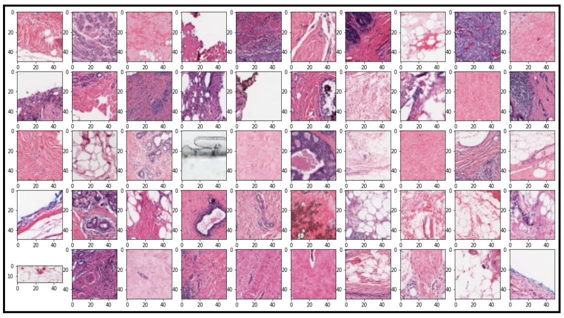


FIGURE 7.18 HISTOPATHOLOGY DATASET IMAGES OF HEALTHY PATCHES

Using this labeled dataset we first build a classification algorithm using CNN. The Model is defined as given below:

```
early_stop=EarlyStopping(monitor='val_loss',patience=5)
model = Sequential()
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same', input_shape=(50, 50, 3)))
model.add(BatchNormalization())
model.add(Conv2D(32, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(MaxPooling2D((2, 2)))
model.add(BatchNormalization())
model.add(Dropout(0.3))
model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(BatchNormalization())
model.add(Conv2D(64, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(BatchNormalization())
model.add(MaxPooling2D((2, 2)))
model.add(Dropout(0.3))
model.add(Conv2D(128, (3, 3), activation='relu', kernel_initializer='he_uniform', padding='same'))
model.add(Flatten())
model.add(Dense(128, activation='relu', kernel_initializer='he_uniform'))
model.add(BatchNormalization())
model.add(Dense(64, activation='relu', kernel_initializer='he_uniform'))
model.add(BatchNormalization())
model.add(Dense(64, activation='relu', kernel_initializer='he_uniform'))
model.add(Dropout(0.3))
model.add(Dense(24, activation='relu', kernel_initializer='he_uniform'))
model.add(Dense(2, activation='softmax'))
```

FIGURE 7.19 CNN LAYERS

Using this model we obtain an accuracy of 94.86%.

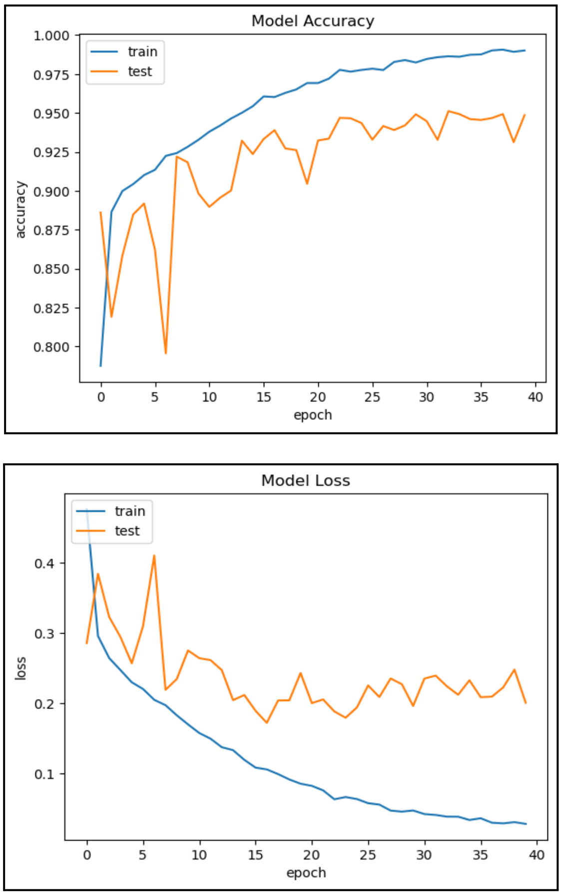
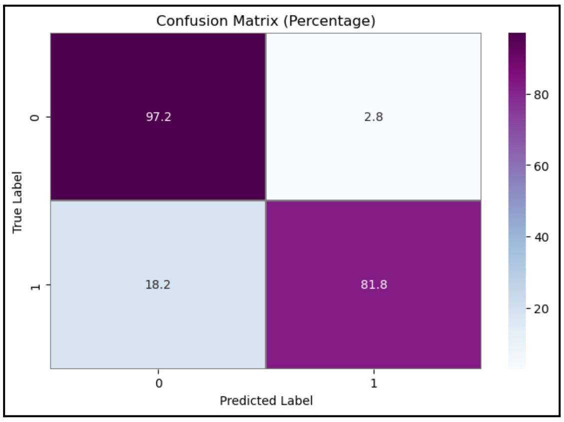


FIGURE 7.20 VISUALIZATION OF HISTOPATHOLOGY ACCURACY




```
1/1 [==============================] - 0s 24ms/step
Predicted Value using  cnn model 1
True Value 1
```

FIGURE 7.21 CONFUSION MATRIX

Finally, a Patch detection method is used to locate the cancer.

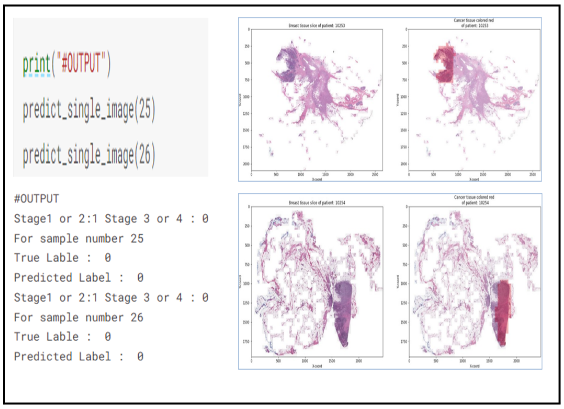


FIGURE 7.22 RESULT OF CANCER LOCATED USING PATCH DETECTION

7.4 Mammography

First we load the dataset to perform some analysis.

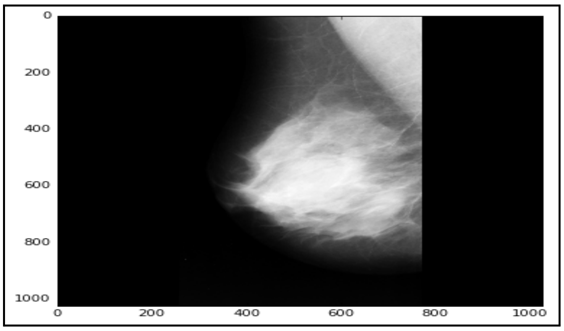


FIGURE 7.23 MAMMOGRAPHY DATASET

From the image we see the Region of Interest (ROI) by drawing a box around it.

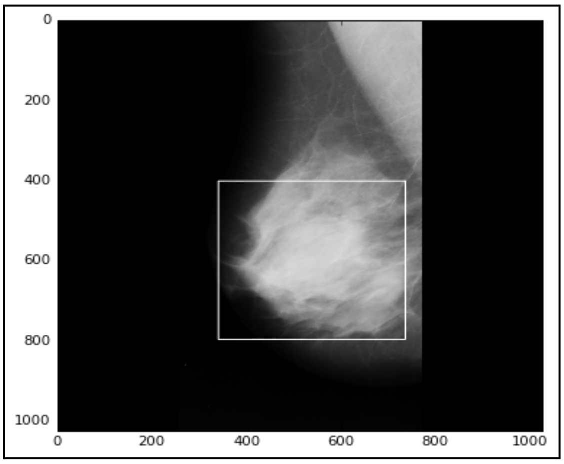


FIGURE 7.24 REGION OF INTEREST

We show the 9 patch image for the large ROI (Region of Interest).

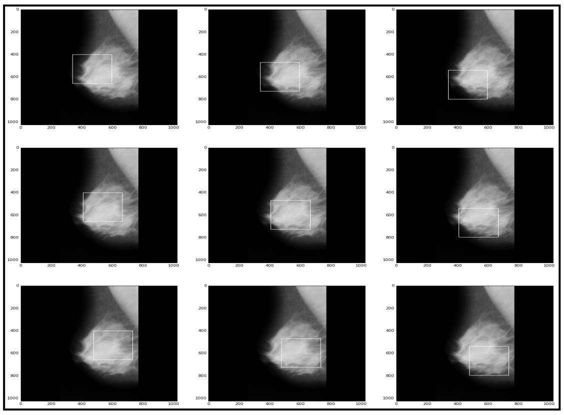


FIGURE 7.25 9 PATCH IMAGES FOR ROI

Then we use find contours for the image segmentation.

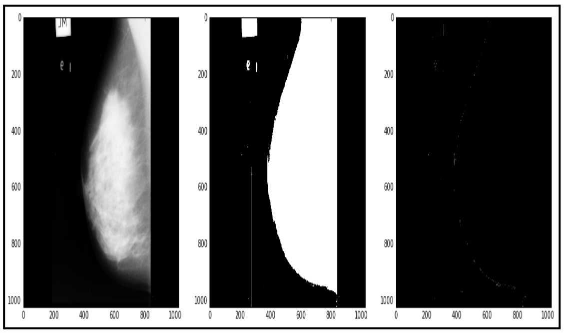


FIGURE 7.26 CONTOURS FOR IMAGE SEGMENTATION

We use Blob detection to find the candidate's Region of interest.

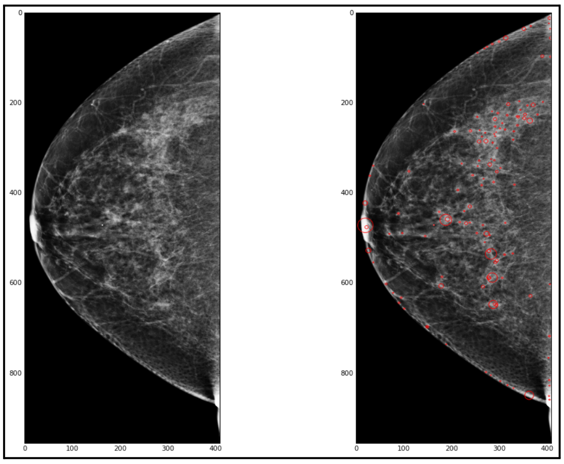


FIGURE 7.27 BLOB DETECTION FOR ROI

Then we test the ROI classifier on key points from mammograms.

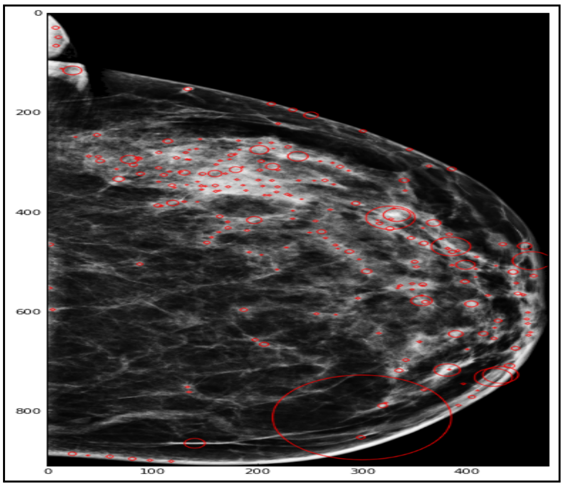


FIGURE 7.28 ROI CLASSIFIER

Then a convolutional neural network (CNN) is used to categorise microcalcifications and masses in a mammogram as benign or malignant. A mammography is categorised as normal if there are no masses in the breast tissue.

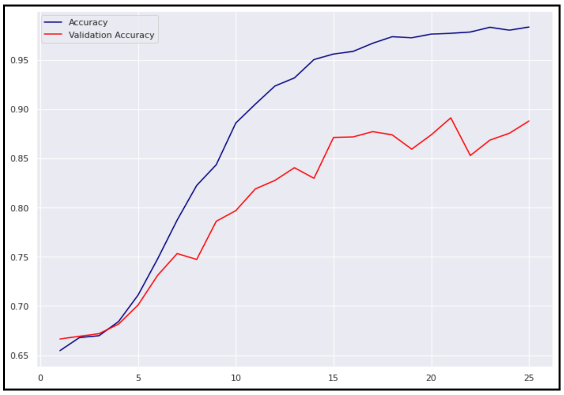


FIGURE 7.29 VISUALIZATION OF THE RESULT

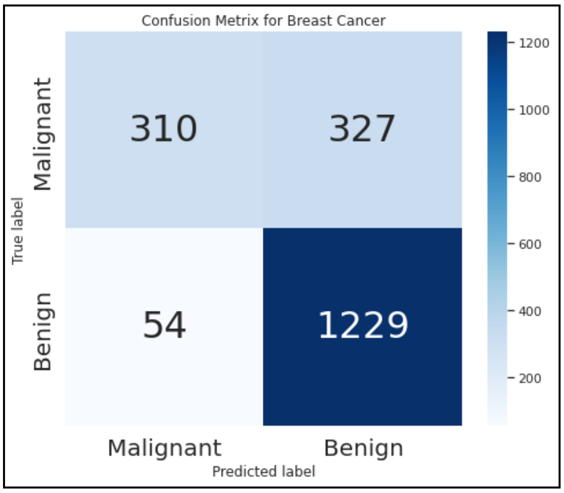


FIGURE 7.30 CONFUSION MATRIX

And then we use Region Proposal algorithms to train Support Vector Machines and Convolutional Neural Networks for mass detection in mammograms. We use morphological enhancement to enhance the image.

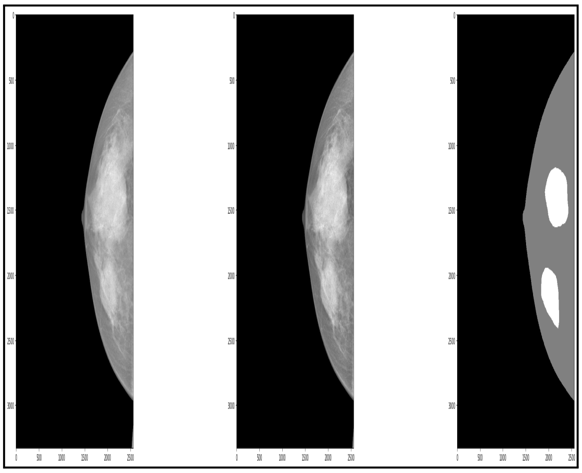


FIGURE 7.31 MULTISCALE MORPHOLOGICAL SIFTING

After Multiscale Morphological Sifting, we perform segmentation of regions of interest.

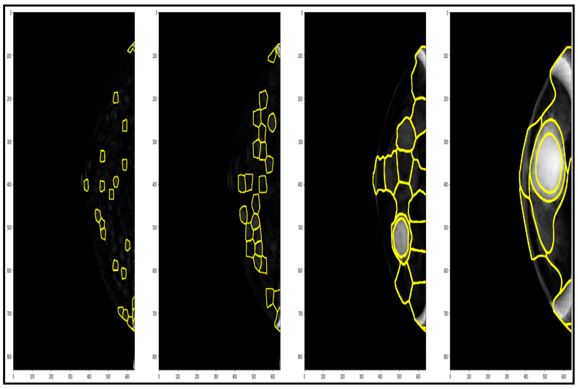


FIGURE 7.32 SEGMENTATION OF ROI

Then we extract the patches using patch extraction.

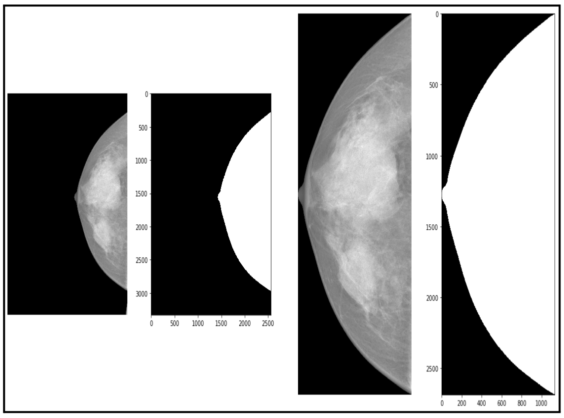


FIGURE 7.33 PATCH EXTRACTION

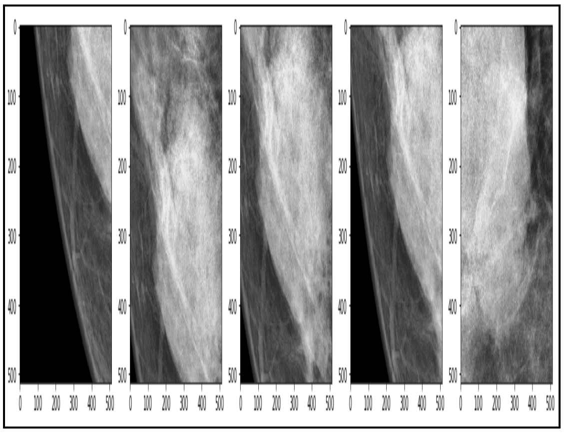


FIGURE 7.34 FINAL RESULT OF MAMMOGRAPHIC IMAGE

8. Conclusion and Future Work

We employed deep learning algorithms to categorise breast cancer pictures as malignant or benign in this study. We made use of ultrasound images, histopathological images, and mammography images. After predicting the malignancy, we apply location detection techniques to pinpoint its exact site.To assess performance, a range of matrices such as accuracy, precision, sensitivity, and specificity were employed.

DenseNet 169 was determined to be the most accurate and effective classifier on the ultrasound pictures dataset, with average accuracies ranging from 94.00% to 100% for binary and multi-class classification. The MASK R-CNN Algorithm is then used to find the malignant tumour.

We used the CNN model to categorise the input on the histopathology picture dataset, and then we used the patch detection approach to indicate the location of the cancer. In this case, multiclass classification was used to determine the patient's stage of cancer.

We tested multiple deep learning models on the mammography picture dataset, with the VGG Model outperforming the others. We categorised the input using that, and then the patch detection technique was applied.

Additionally, we plan to broaden our framework to enable it to handle more complicated datasets in order to diagnose breast cancer photos using a single application.

9. References

[1]A novel deep learning based framework for the detection and classification of breast cancer using transfer learning - Khan S, Islam N, Jan Z, Din IU, Rodrigues JJPC.

[2]Deep convolutional neural networks for breast cancer screening - Chougrad H, Zouaki H, Alheyane O.

[3]Breast cancer classification from histopathological images using patch-based deep learning modeling - Hirra, I.; Ahmad, M.; Hussain, A.; Ashraf, M.U.; Saeed, I.A.; Qadri, S.F.; Alghamdi, A.M.; Alfakeeh, A.S.

[4]A new transfer learning based approach to magnification dependent and independent classification of breast cancer in histopathological images- Boumaraf S, Liu X, Zheng Z, Ma X, Ferkous C.

[5]Improved breast cancer classification through combining graph convolutional network and convolutional neural network - Zhang Y-D, Satapathy SC, Guttery DS, Go´rriz JM, Wang S-H.

[6]Breast cancer detection using deep convolutional neural networks and support vector machines - Dina A Ragab 1 2, Maha Sharkas 1, Stephen Marshall 2, Jinchang Ren 2.

[7]Integrating spatial fuzzy clustering with level set methods for automated medical image segmentation - Bing Lan Li et al.

[8]M. Heenaye-Mamode Khan, N. Boodoo-Jahangeer, W. Dullull et al., “Multi- class classification of breast cancer abnormalities using deep convolutional neural network (CNN),” PLoS One, vol. 16, no. 8, article e0256500, 2021.

[9]Alexey Natekin and Alois Knoll, Gradient boosting machines, a tutorial, Frontiers in neurorobotics, 7, 21, 2013

[10]Christopher W. Elston and Ian O. Ellis, Pathological prognostic factors in breast cancer. I. The value of histological grade in breast cancer: experience from a large study with long-term follow-up, Histopathology, 19 (5), 403–410, 1991.

[11]Diederik P. Kingma and Jimmy Ba. “Adam: A Method for Stochastic Optimization”. In: (Dec. 22, 2014). arXiv: 1412.6980 [cs].