# A Deep Learning Approach for the Automatic Analysis and Prediction of Breast Cancer for Histopathological Images using A Webapp

Dr.Mohammed Abdul Raheem[1]
[1]Assistant Professor
Dept. of Electronics and Communication Engineering,
Muffakham Jah College of Engineering and Technology,
Osmania University, Hyderabad, Telangana, India

Shaik Tabassum[2], Syeda Kulsoom Nahid[2],
Syeda Areej Anzer[2]
[2]Students of B.E.
Dept. of Electronics and Communication Engineering,
Muffakham Jah College of Engineering and Technology,
Osmania University, Hyderabad, Telangana, India

***Abstract*— Since time immemorial, women are the victims of breast cancer which is a predominant disease worldwide with high dreariness and mortality. The absence of careful visualization models brings about trouble for specialists to set up a treatment plan that may lengthen outpatient endurance time. More often than not, Breast Malignancy is distinguished by utilizing a biopsy strategy where the tissue is taken out and examined under a magnifying lens. In the event that a histopathologist isn't very much trained person, at that point this may lead to some wrong findings. And due to this one may have to undergo wrong diagnosis. To encourage better diagnosis, the programmed examination of histopathology images can assist pathologists with recognizing harmful tumors and malignancy subtypes. Convolutional Neural Networks have become favored Deep Learning approaches for computer vision tasks which involves feature extraction and image classification. This proposed work focuses on building a machine learning model that can classify histopathology images into two classes namely cancer (malignant) and non-cancer (benign) using transfer learning approach. To make our work beneficial to everyone, we made a website that is being built and deployed using Streamlit library, where we integrated the model with highest accuracy for classifying the histopathology images. Our work will help doctors and medical practitioners for early diagnosis of breast cancer.**

***Keywords*— Breast Cancer; Histopathology; Machine learning; Deep Learning;Convolutional Neural Networks ;VGG16; Xception ; NASNet ; ResNet50.***

## I. INTRODUCTION

As per the World Health Organization, Breast Cancer now is one of the most frequently analyzed and fatal cancers among women which affects 2.1 million women per year. It is assessed that the normal number of breast cancer patients will be in excess of 28 million out of 2030 [2], which prompts a significant examination subject in clinical sciences [1].

Breast Cancer mainly develops from Breast tissue distinguished by a lump in the breast and some unusual changes observed in typical conditions [3]. The reason for Breast cancerous growth hails from changes and mutations in DNA. If the changes are not detected at an early stage, it may lead to patient's death [4]. This accounts for the constant studies being carried out in this domain [12][13][14].

The Deep learning architectures especially the Convolutional neural networks have shown significant performance over the image classification tasks [5][6].The proposed work aims to use pre-trained CNN classifiers which can differentiate between healthy tissues and cancerous samples. For this purpose, pre-trained CNNs such as VGG16, ResNet50, Xception, NASNetMobile are used. After analyzing individual model performances ,we concatenated these models one over the other which resulted in two new models i.e. .VGG16+ResNet50 (VGG16 concatenated with ResNet50) and Xception+NASNet (Xception concatenated with NASNetMobile . We compared the performance of the models to arrive at a model with highest accuracy. To make better use of the study carried out, we then integrated the same model in a website that is capable of making predictions on an uploaded histopathology image and returning the probability of having cancer. Then we deployed the same in order to make it accessible to everyone.

## II. RELATED WORKS

Myung Jaelim et.al [7] from Eulji University, Korea carried out their studies on histopathological images classification using deep learning algorithms such as VGG16 and InceptionV3 .They applied these algorithms on BreakHis dataset and achieved an accuracy of 98% in their work.

Neslihan Bayromoghu et.al [8] in their work, designed a custom CNN network. They basically proposed two different CNN architectures. Of which one is a single task CNN, used for predicting the malignancy which has achieved an accuracy of 77.3±5.91% to 83±8.54%.And the second one is a multi-task CNN, which is used for predicting both the state of malignancy and the image magnification levels at the same time. This model has achieved an accuracy of 82.1±4.4% to 83.1±3.5%.They have used BreakHis dataset for this purpose.

Majid Nawaz et.al [9] of Assiut University, proposed a deep learning approach for multi-class breast cancer classification Meaning, not only benign and malignant but also predicting cancer sub-classes such as Ductal carcinoma, Fibroadenoma, Lobular carcinoma, etc. They have used deep learning models like DenseNet CNN and have achieved an accuracy of 95.4%. They carried out their studies on BreakHis dataset.

ZhongyiHan et.al [10] from Shandong University of Traditional Chinese Medicine, China have compared the performances of newly proposed CNN network such as

AlexNet, LeNet, and CSDCNN with both un-augmented data and augmented Data and have achieved an accuracy of 93.2% on BreakHis dataset.

Abdullah-Al Nahid et.al [11] from Macquarie University, Australia have proposed a novel DNN model. The proposed model consist of a combination of CNN and LSTM for feature extraction and softmax, Support Vector Machine (SVM) are used at the decision making stage for binary classification. The model has achieved an accuracy of 91.00% over BreakHis dataset.

## III. ALGORITHMS USED

### *A. Pre-trained Convolutional Neural Network Models*

There are some traditional classifiers provided by the Keras library. The architectures used in our proposed work have been trained on the images from ImageNet dataset which consist of over 14 million number of images. These images in the dataset belong to 1000 different classes with image size as 224-by-224. That means, all these pre-trained networks are capable of classifying the images into 1000 different categories. However, one can give images of different dimensions other than this based on their requirement. The models used in this proposed work are as follows:

1. *VGG16*
   This model stood at 1st and 2nd place in ILSVRC-2014 image classification competition. This model has achieved test accuracy of about 92.7% over the ImageNet dataset .And it is available as an API in the Keras library and enables one to just load the model with the same weight with which model has been trained upon and use it for custom classification task.

2. *ResNet50*
   It is the short form for Residual networks. This network has shown outstanding performance in most of the computer vision problems. It has achieved Rank-1 in the ImageNet classification challenge of 2015. It is also available as an API and can be downloaded from the Keras library.

3. *Xception*
   Xception has got 36 convolutional layers and the model has 71 layers deep architecture which forms the basis for feature extraction capability. Its deep layer architecture enables the model to perform well in the image classification tasks.

4. *NASNet*
   NasNet on the other hand, has got 87.56% of top-1 accuracy over the ImageNet dataset. This model has shown excellent performance in image classification task.

### *B. Description of Dataset*

For our work, we have used a dataset from kaggle competitions which is a modified version of PatchCamelyon (PCam) dataset[1]. It contains the images of metastatic cancer in the form of small image patches taken from larger digital pathology scans in .tif format. The train set in this dataset consist 220025 number of images and the test set consist of 57458 number of images of size (50-by-50). This is a pretty huge data .If we train our pre-trained CNN model on the entire dataset, it will take lot of time and computational power for training. The entire process starting from Data acquisition to model testing has been carried out on Google colab with 12GB RAM Tesla T4 GPU.

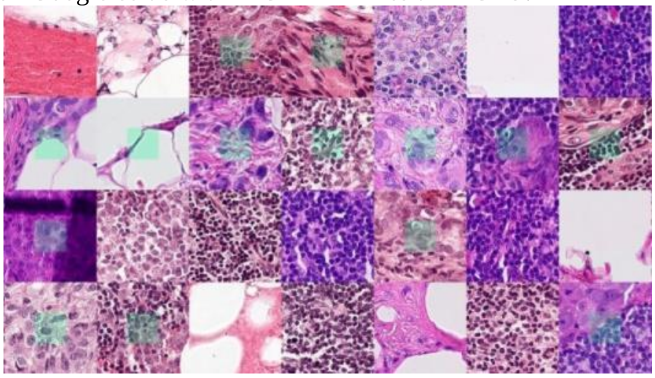


Fig. 1.Image patches from the PCam dataset

### *C. Visualization of Dataset*

Visualization of the dataset is carried out to understand the style and pattern of the data ie. percentage of images belonging to each class and is plotted in the form of a pie-chart using the matplot library. In the PCam dataset 40.5% of images belong to cancerous class and 59.5% of images are Non-cancerous.

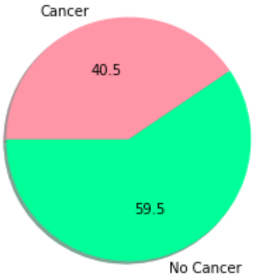


Fig. 2.Visualization of PCam dataset

## IV. PROPOSED METHODOLOGY

Our proposed methodology consists of building a machine learning model using the concept of transfer learning which facilitates one to use pre-trained models that have already been trained on a different but related problem [15].For eg, the knowledge acquired from the process of learning to recognize cars can be applied for the task of recognizing trucks. In this proposed system, we have used VGG16, ResNet50, Xception, NasnetMobile pre-trained models for classification. After analyzing their performances, we have gone for concatenating two pre-trained CNN models in order to increase validation accuracies. We carried out this methodology of concatenation of models over VGG16 & ResNet50 represented as VGG16+ResNet50 and also over

Xception & NASNetMobile represented as Xception+NASNet. As these pre-trained CNN models have been trained over million number of images in ImageNet challenge and have the capability to generalize the input images into thousand different categories, we need to define our own custom top layers which can classify the input images into only two categories with the help of a series of flatten, dense and dropout layers. This work consists of the following phases as shown in the block diagram.

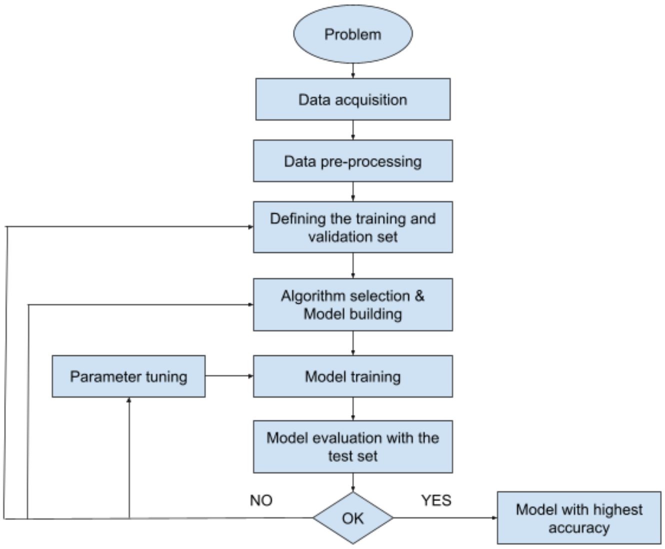

Problem
Data acquisition
Data pre-processing
Defining the training and validation set
Algorithm selection & Model building
Parameter tuning
Model training
Model evaluation with the test set
NO
OK
YES
Model with highest accuracy

Fig. 3.Supervised machine learning pipeline that is used to create a successful classifier which gives highest accuracy

### A. Data acquisition

The very first phase in our project starts with collection of data required for undertaking the research. As the problem statement mainly focuses on building a machine learning model which can classify histopathological images into two categories i.e. cancer or no-cancer. For this purpose we have used a dataset from kaggle competitions which contains the images in .tif format. The train set in the original dataset consist 220025 number of images and the test set consist of 57458 number of images. This is too large and hence we took a sample no. of images of about 8500 belonging to each class i.e. 0 and 1. Therefore, total number of images used has been reduced to 17000.

### B. Data pre-processing

Data pre-processing step involves labeling the images with 0 and 1 and also splitting the dataset into training and validation sets. The training ratio that has been employed is 0.9 i.e, 90% of data is used for training (15300 images belonging to two classes) and 10% for validation (1700 images belonging to two classes).The dataset from kaggle consists of a label.csv file which contains specific patient id with corresponding label value. Here label =0 indicates that the patient is healthy and label=1 indicates that the patient is having Cancer. On the basis of the information provided from this file, we have labeled the images with 0’s and 1’s.

### C. Data augmentation

Image augmentation is a technique of applying different transformations to original images which results in multiple transformed copies of the same image by applying shifting, rotating, flipping etc. on the original image. For this purpose, we have used Keras ImageDataGenerator class which provides a quick and easy way to augment the images.

The primary benefit of using the Keras ImageDataGenerator class is that it is designed to provide real-time data augmentation. Which means it is generating augmented images on the fly while the model is being trained. ImageDataGenerator class produces variations of the images for each epoch. And it also reduces the memory usage required for storing the images.

### D. Model building

Building the model consists of the following steps:

1. *Downloading the model*
   Keras gives access to a number of top-performing pre-trained models that were created for image recognition tasks. These models are made available as Application APIs. We just have to download the model and start using the model for custom classification problem.

2. *Defining top layers*
   The top layers are meant for classifying the images into thousand different categories. Our problem statement deals with binary classification i.e., Cancer or No-Cancer. Hence, we defined custom top layers which can classify the input images into only two categories with the help of a series of flatten, dense and dropout layers.

3. *Compiling the model*
   Before training, the model has to be compiled. When the model is compiled for training, it is provided with loss function and optimizer. The loss function is meant for measuring how far off the predictions are from the actual outcomes. And the optimizer is used for adjusting the internal values to reduce the loss. We have used binary cross entropy as the loss function and Adam optimizer for compiling the model.

### E. Model training

The fit_generator() is a method provided by Keras library which can be used for train the deep learning models. During this training process, the model takes in the probability values, performs a calculation using the weights and generates outputs. For the first iteration over the training set the predicted outcomes are not as close as the actual values. The difference between the actual value and the desired value is calculated using the loss function, and the optimizer function directs how the weights ought to be changed. This process of calculating, comparing, adjusting is controlled by the “fit_generator” and it continues in accordance with the number of epochs specified. Each epoch corresponds to one forward and backward propagation over the entire dataset (here 17000 images). And at each epoch, the model will be fed with unique set of 17000 augmented images. These augmented images will not be supplied at once to the model, but in the form batches. Each batch contains some ‘x’ number of images, where ‘x’ specifies the value given to the argument batch size in fit_generator function. If we say the number of epochs is 20, then by the end of training the model is trained over 17000x20= 3,40,000 number of augmented images. Therefore, with the help of only 17000 images we made the model to be learned over 3,40,000 unique images which enables the model to be beautifully fit with the given input data. It also prevents the model from over-fitting.

### *F. Model evaluation*

The validation set is then given to the model for evaluation. The model can be tested for its performance using predict_generator() function. The results were then plotted in the form of a graph called the ROC (receiver operating characteristic curve) curve and model performance is evaluated using AUC (Area under Curve) values.

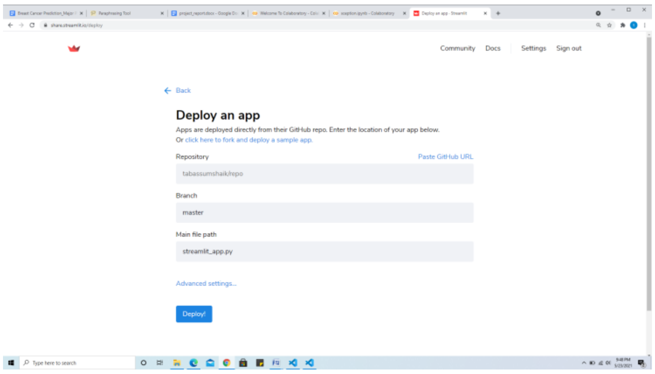
## V. Model Deployment On Streamlit Share

Streamlit is an open-source python library for building beautiful custom webapps for machine learning in an easy way. After creating a virtual environment, all the required libraries needs to be installed. The code for prediction resides in a .py file. Inside this python file consists of a function that takes in an image uploaded by the user and makes prediction on the uploaded image and displays the result with the corresponding probability value on to the browser. The streamlit library provides an API for file uploading functionality called file_uploader[13].

The model that has been saved in .h5 file is then loaded and given as a parameter to the prediction function. The uploaded image is first resized to (224, 224, 3) as our pre-trained models have been trained on these dimensions. This image is then converted into a numpy array. This array has values ranging from 0 to 255. And this will make the difficult to learn on and generalize between the images. Hence, this array is then normalized to have values between 0 to1. This normalized image array was given to the model for prediction. And the model then returns probability values ranging from 0 to 1 in correspondence with the status of uploaded image being cancer or no-cancer.

The WebApp is then improvised over the front-end of the application by adding custom HTML and CSS using markdown API provided by the streamlit library.

The WebApp is then tested for its actual functionality by running it on the localhost: 8501. The code is then pushed to a GitHub[2] repository in order to make our work publicly available to everyone. The stack of files also consists of the requirements.txt file which contains information of all the libraries that have been used. And the website is deployed as a public accessible URL using the streamlit share.


Breast Cancer Prediction_Major
Paraphrasing Tool
project_report.docx - Google Do
Welcome To Colaboratory - Cola
xception.ipynb - Colaboratory
Deploy an app · Streamlit
share.streamlit.io/deploy
Community
Docs
Settings
Sign out
Back
Paste GitHub URL
Advanced settings...
Deploy!
Type here to search

Deploy an app

Apps are deployed directly from their GitHub repo. Enter the location of your app below.
Or click here to fork and deploy a sample app.

Repository: tabassumshaik/repo

Branch: master

Main file path: streamlit_app.py

Fig. 4.Figure shows the deployment of website on streamlit share

## VI. RESULTS AND DISCUSSION

### *A. Comparison of performance and arriving at the best model*

Each model was trained over 20 epochs and corresponding values of training and validation accuracies were obtained .The time taken for each model for training will be different because it depends on the model architecture. Complex architectures will take more time when compared to simple models. The best training and validation accuracy values over 20 epochs are as shown below:

TABLE I.

| *Classifier* | *Training accuracy* | *Validation accuracy* | *Time taken per epoch(sec)* |
|---|---|---|---|
| ResNet50 | 77.59% | 76.18% | 199 |
| VGG16 | 92.11% | 93.12% | 294 |
| Xception | 88.97% | 90.94% | 180 |
| NASNetMobile | 82.61% | 83.41% | 225 |

The validation accuracy will not be that appreciable for the very first epoch and it will take some time for the model to learn the features from the images. For the ResNet50 model, it took around 199 seconds for each epoch and the validation accuracy was 75.35% for the first epoch and it has been increased up to76.18% over 20 epochs. The training accuracy was 77.59% .There was not much difference between these two values (only 1%) and the model is neither under-fitting nor over-fitting. The same procedure was applied for VGG16, Xception and NASNetMobile pre-trained models .It took around 294s ,180s ,225s respectively for each epoch and their validation accuracies were 93.12%, 90.94% and 83.41% respectively.

After analyzing the performances of these models, two pre-trained models were concatenated one over the other to increase model accuracy. Below table shows the training and validation accuracies after concatenating the pre-trained models:

TABLE II.

| *Classifier* | *Training accuracy* | *Validation accuracy* | *Time taken per epoch(sec)* |
|---|---|---|---|
| VGG16+ResNet50 | 94.54% | 94.12% | 401 |
| Xception+NASNet | 96.06% | 95.65% | 558 |

The process of concatenation increased the time required for training the model because the model is complex. And it took around 401s, 5558s for training the models

VGG16+ResNet50 and Xception+NASNet respectively. It also increased the model performance as the model with highest validation accuracy i.e. 95.65% was obtained for Xception+NASNet model.

Area under the curve (AUC) is another performance evaluation criteria for CNN which is obtained from ROC.

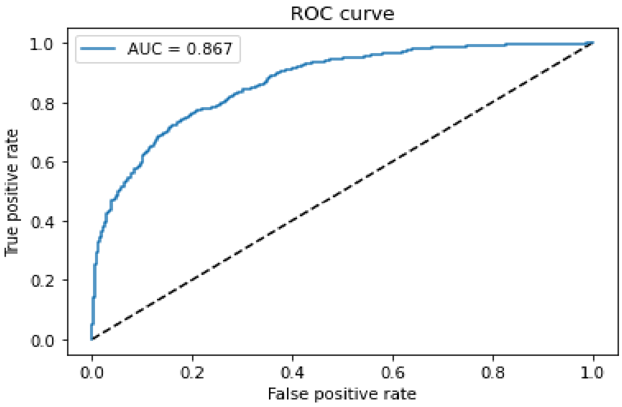


Fig. 4.ResNet50 model

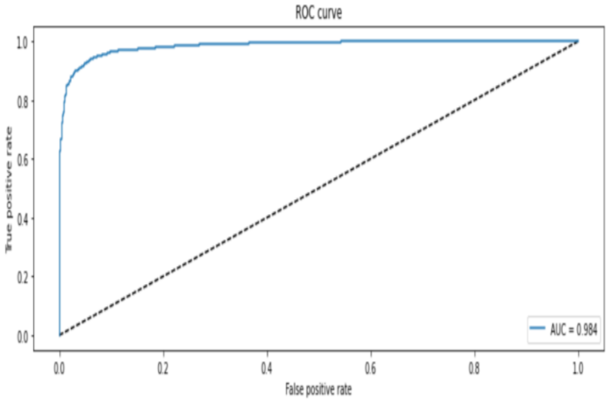


Fig. 5.VGG16 model

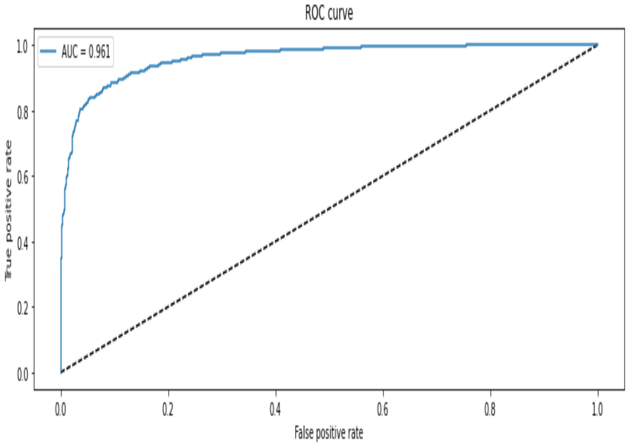


Fig. 6.Xception model

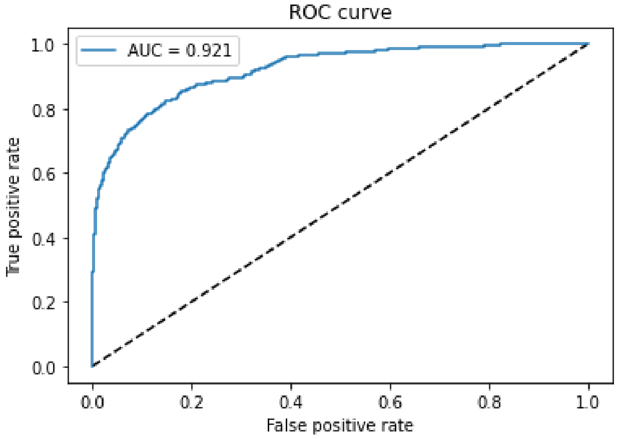


Fig. 7.NASNetMobile model

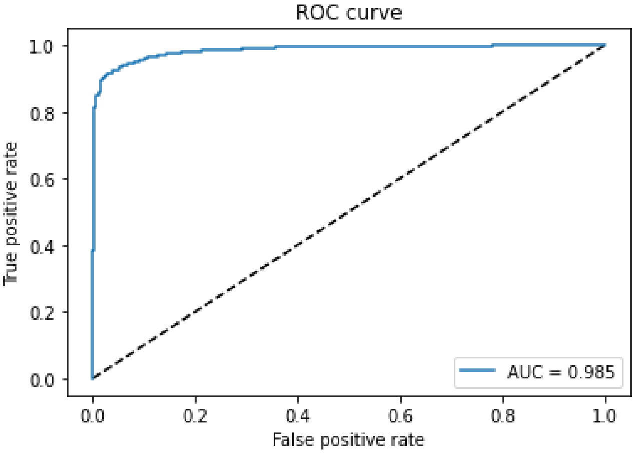


Fig. 8.VGG16 concatenated with ResNet50

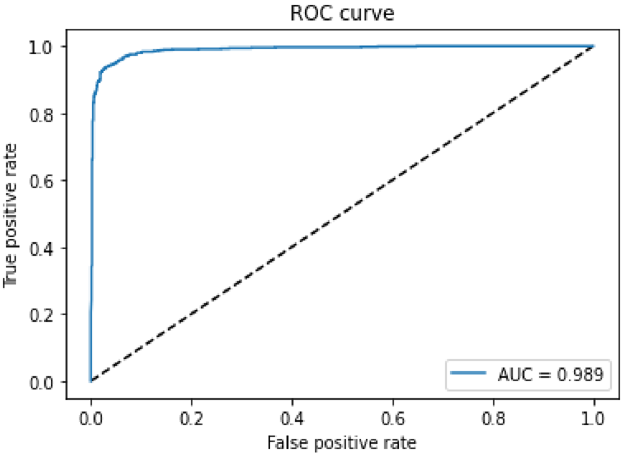


Fig. 9.Xception concatenated with NASNetMobile

The area under the curve values (AUC) for the respective model is listed in the table as shown below:

TABLE III.

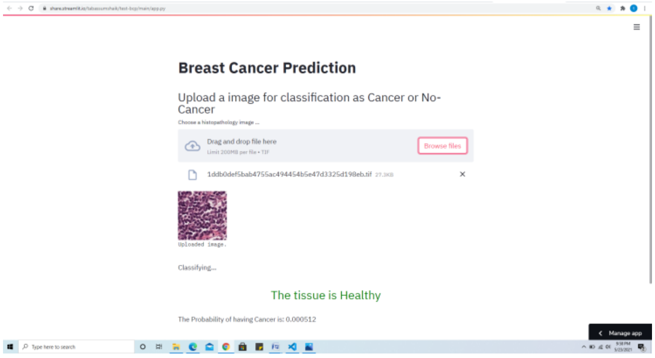
| ***Classifier*** | **AUC Value** |
|---|---|
| ResNet50 | 86.7 |
| VGG16 | 98.4 |
| Xception | 96.1 |
| NASNetMobile | 92.1 |
| VGG16+ResNet50 | 98.5 |
| Xception+NASNet | 98.9 |

a.

Among all the six models, Xception+NasNet model has yielded highest AUC value around 98.9. Later this model is used for WebApp development and deployment.

### B. Model deployment on Streamlit Share

The image after deploying the website as a public accessible URL[3] is as shown:



Breast Cancer Prediction

Upload a image for classification as Cancer or No-Cancer

Choose a histopathology image ...

Drag and drop file here
Limit 200MB per file • TIF

1ddb0def5bab4755ac494454b5e47d3325d198eb.tif 27.3KB

Uploaded image.

Classifying...

The tissue is Healthy

The Probability of having Cancer is: 0.000512

Fig. 11.Figure shows the website being able to successfully make predictions on the uploaded image with the appropriate probability

The user have to upload a histopathology image through the browse files button. The uploaded image is then the resized to (224x224), converted into an array, values will be normalized and then this normalized array will be fed to the model for prediction .The model then returns the probability of having cancer onto the browser. If the probability value is greater than 0.5, then the model predicts the tissue to be Cancerous otherwise Non-cancerous.

## VII. CONCLUSION AND FUTURE WORK

Taking everything into account, Convolutional neural Networks (CNNs) are the current state-of-the-art design for automatic classification of histopathological images. In our proposed methodology, we analyzed the performances of various combinations of pre-trained CNN models i.e., VGG16, Resnet50, Xception, NASNetMobile, VGG16+ResNet50 and XCeption+NASNet. From results, we observed that the Xception+Nasnet network have achieved the highest accuracy i.e., 95.65% in comparison with other models. However, there are some limitations in our work. Although the accuracy is high, the model is still generating few false predictions with the real-time data.

As we carried out our work on the Google colab platform which offered us Tesla T4 GPU for training the model, one can use higher configuration GPUs or even TPUs for training the model with increased number of epochs which will ultimately increase the model performance.

The functionality of the website can be further improved by adding features like scanning the pictures (website being able to directly taking the pictures of the scanned tissue that are generated after biopsy) and making predictions on the scanned image in various image formats (.jpg/.png/.jpeg/.tif) and also with different magnification factors (40x, 100x, 200x and 400x).We carried out our work on binary class classification i.e. Benign or Malignant. One can extend our work for multi-class classification for making predictions over cancer sub-types.

## ACKNOWLEDGMENT

We would like to thank Kaggle community for making the PCam dataset publicly available for everyone.

## REFERENCES

[1] B.J. Williams, A. Hanby, R. Millican-Slater, A. Nijhawan, E. Verghese, D. Treanor,”Digital pathology for the primary diagnosis of breast histopathological specimens: an innovative validation and concordance study on digital pathology validation and training”, Histopathology, 72 (4) (2018), pp. 662-671.

[2] C. Fitzmaurice, C. Allen, R. M. Barber, L. Dandona et al., "Global regional and national cancer incidence mortality years of life lost years lived with disability and disability-adjusted life-years for 32 cancer groups 1990 to 2015: a systematic analysis for the global burden of disease study", JAMA oncology, vol. 3, no. 4, pp. 524-548, 2017.

[3] G. Viale, N. Rotmensz, P. Maisonneuve, E. Orvieto, E. Maiorano, V. Galimberti, et al., "Lack of prognostic significance of classic lobular breast carcinoma: a matched single institution series," Breast cancer research and treatment, vol. 117, no. 1, pp. 211, 2009.

[4] Sushma L , Dr. K.P. Lakshmi , “An Analysis of Convolution Neural Network for Image Classification using Different Models,” International Journal of Engineering Research & Technology, Vol. 9 Issue 10,2020.

[5] Yadav, S.S., Jadhav, S.M. ,“Deep convolutional neural network based medical image classification for disease diagnosis,”. J Big Data 6**,** 113 (2019).

[6] Waseem Rawat, Zenghui Wang,”Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review,” Neural Comput 2017; 29 (9): 2352–2449.

[7] Lim, Myung & Kim, Da & Chung, Dong & Lim, Hoon & Kwon, Young,“ Deep Convolution Neural Networks for Medical Image Analysis”, International Journal of Engineering and Technology(UAE). 7. 115-119. 10.14419/ijet.v7i3.33.18588 (2018).

[8] N. Bayramoglu, J. Kannala and J. Heikkilä, "Deep learning for magnification independent breast cancer histopathology image classification,” 2016 23rd International Conference on Pattern Recognition (ICPR), Cancun, 2016, pp. 2440-2445.

[9] Majid Nawaz, Adel A. Sewissy and Taysir Hassan A. Soliman, “Multi-Class Breast Cancer Classification using Deep Learning Convolutional Neural Network,” International Journal of Advanced Computer Science and Applications (IJACSA), 9(6), 2018.

[10] Han, Zhongyi, Benzheng Wei, Yuanjie Zheng, Yilong Yin, Kejian Li and Shuo Li., “Breast Cancer Multi-classification from Histopathological Images with Structured Deep Learning Model,” Scientific Reports (2017).

[11] Abdullah-Al Nahid, Mohamad Ali Mehrabi, and Yinan Kong, “Histopathological Breast Cancer Image Classification by Deep Neural Network Techniques Guided by Local Clustering,” BioMed Research International, vol. 2018, Article ID 2362108, 20 pages, 2018.

[12] M. Jannesari et al., "Breast Cancer Histopathological Image Classification: A Deep Learning Approach," 2018 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), Madrid, Spain, 2018, pp. 2405-2412 .

[13] F. A. Spanhol, L. S. Oliveira, C. Petitjean and L. Heutte, "Breast cancer histopathological image classification using Convolutional Neural Networks," 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 2560-2567.

[14] V. Gupta and A. Bhavsar, "Breast Cancer Histopathological Image Classification: Is Magnification Important?," 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2017, pp. 769-776.

[15] R. Ribani and M. Marengoni, "A Survey of Transfer Learning for Convolutional Neural Networks," 2019 32nd SIBGRAPI Conference on Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2019, pp. 47-57.