**Breast Cancer Detection from Mammograms using Deep Learning and Machine Learning**

Kolla Ananta Raj

**Abstract**

Breast cancer is one of the most frequent invasive and rapidly developing cancers in women globally, and it is a leading cause of death in women. This mostly occurs in females but in rare instances, it can also affect male patients. The risk of breast cancer can be reduced by spreading awareness about its symptoms, or an early-stage detection of cancer. Detection at very advanced stages implies the treatment is more difficult and uncertain. Many research works have been done in this field and many studies show that early detection of breast cancer has been linked to a higher mortality rate. Early identification of cancer, according to the World Health Organization (WHO), considerably increases the odds of making the best selection on a successful treatment strategy. In our present scenario, many Machine Learning and Deep Learning algorithms have been put to use for early-stage breast cancer detection. Standard Machine Learning algorithms such as RandomForest and XGBoost have shown promising results in breast cancer classification tasks. Deep Learning is a technique that has several advantages over Machine Learning algorithms. Deep Learning models such as the Convolutional Neural Networks (CNN) are specifically made for image data. These models analyze the data in multiple layers and extract features out of them just like the human brain has neurons. These multiple analysis layers make the model learn better features and make the model strong in comparison to the machine learning algorithms. Apart from conventional learning algorithms, we use the transfer learning technique which uses knowledge from its previous training in another related problem set. In our work, we have demonstrated the use of Deep Learning models through Transfer learning, Deep Feature Extraction, Machine learning models, and how they compare. Our findings have found that Deep learning models can achieve training accuracies of up to 99%, validation accuracies up to 97%, and test accuracies of up to 96%. In the end, we conclude by suggesting various improvements that could be made to existing architectures and how deep learning and machine learning approaches could assist enhance breast cancer screening and early detection.

**Keywords:** Artificial Intelligence; Deep learning; Convolutional Neural Networks; Transfer Learning.

1. **Introduction**

Breast cancer emerges within the lining cells of the conduits (85%) or lobules (15%) within the glandular tissue of the breast. At first, the cancerous development is restricted to the channel or lobule (“in situ”) where it for the most part causes no indications and has negligible potential for spread (metastasis). Over time, these in situ (stage 0) cancers may advance and attack the encompassing breast tissue (obtrusive breast cancer) at that point spread to the adjacent lymph nodes (territorial metastasis) or to other organs within the body (removed metastasis). Breast cancer occurs when cells in the breast begin to grow out of control. These cells normally grow into a tumor, which can be visible on x-rays or felt like a bump. If the cells in the tumor can develop into (invade) surrounding tissues or spread (metastasize) to other parts of the body, it is considered malignant (cancer).

Treatment for breast cancer can be quite effective, especially if the disease is detected early. To tackle this problem of early-stage detection, Deep Learning methods have come up, and hence in our study, we have proposed multilayered Convolutional Neural Network models for aiding manual analysis. Breast cancer will affect 2.3 million women globally in 2020, with 685 000 fatalities. Breast cancer had been diagnosed in 7.8 million women in the previous five years as of the end of 2020, making it the most common cancer in the world. Breast cancer strikes women at any age after puberty in every country on the planet, with rates rising as they become older. Deep learning techniques are revolutionizing the field of medical image analysis, and hence in this study, we proposed Convolutional Neural Networks (CNNs) for breast mass detection to minimize the overheads of manual analysis[[1]](https://www.zotero.org/google-docs/?19Frgp). Breast cancer was recorded in one out of every 800 women in the United States in 2017. Female Breast Cancer has surpassed lung cancer as the most often diagnosed cancer as of 2020, with a projected 2.3 million new cases (11.7%), followed by lung cancer (11.4%), colorectal cancer (10%), prostate cancer (7.3%), and stomach cancer (7.3%). (5,6 percent ). With 685,000 deaths worldwide, it is the seventh-highest cause of cancer death. Breast cancer is the most common cancer in women, accounting for one out of every four cases and one out of every six deaths, and it is the most common disease in the great majority of countries (159 of 185 countries)[[2]](https://www.zotero.org/google-docs/?nWrJSN). Because of the medical importance of breast cancer screening, CAD techniques for detecting anomalies such as calcifications, masses, architectural distortion, and bilateral asymmetry have been developed [[3]](https://www.zotero.org/google-docs/?pZRoAJ).

In our study, we have used mammography scan images from the popular MIAS and DDSM datasets. Transfer learning has been used extensively for this purpose, as shown in the work proposed by a paper in 2019[[4]](https://www.zotero.org/google-docs/?AgotVE). Transfer learning has been used extensively for this purpose, as shown in the work proposed by a paper in 2018[[5]](https://www.zotero.org/google-docs/?xulZZf). In our work, we aim to evaluate and assess the impact of deep learning-based techniques that can help health professionals screen and diagnose breast cancer early and help save lives[[6]](https://www.zotero.org/google-docs/?4Fp7Ov).

1. **Background/Literature Review**

A massive amount of research has been done in the field of medical disease detection over the past few decades, and thorough research is going on now. Deep Learning has been a key element in the successful results of Breast Cancer Classification and segmentation in the past few years. Convolutional Neural Networks have helped researchers map important features from the scan images of a breast and classify them into various kinds of abnormalities. The use of Convolutional neural networks enabled the use of transfer learning for breast cancer classification where the knowledge of previous Mammography Scan Data can be retained. The large pre-trained architectures have helped in detecting features on the map onto classifying the mass of cancer. In recent work [[7]](https://www.zotero.org/google-docs/?BODIvt), researchers have been successful in combining the deep learning model’s predictions with the pathologist’s diagnosis, decreasing the human error rate by approximately 85%. This shows the power of combining Deep learning knowledge to significantly enhance the diagnosis. A paper [[8]](https://www.zotero.org/google-docs/?A5MqOA) in 2018 applied deep learning techniques such as convolutional neural networks, sparse autoencoders, and stacked sparse autoencoders and the performance analysis of the three techniques showed that stacked sparse autoencoders performed better. A Faster Region convolutional neural networks-based model was proposed[[9]](https://www.zotero.org/google-docs/?vNXpif) in 2020 for detecting masses in the OMI-DB dataset. The framework achieved a TPR of 0.99 for malignant and 0.85 for benign masses, demonstrating the models' potential for usage as an automated component in an advanced CAD system for breast screening.

1. **Deep Learning for Disease Detection**
   1. **Artificial Neural Networks**

Artificial Neural Networks find their application in various tasks and are capable in pattern recognition [[10], [11]](https://www.zotero.org/google-docs/?d6t0Pa), classification, and modeling non-linear problems. ANN’s are flexible and can adapt itself various forms of data including imaging data. Deep Artificial Neural Networks (ANN) have been used in many fields for classification tasks, pattern recognition, predictive modeling and have proved to deliver excellent performance on such data. ANN's have been fast-growing for their ability to adapt to different kinds of data. Through artificial neural networks, a large amount of data can be manipulated. It can be done by combining deep neural networks, pre-trained networks, and hyper-parameter tuning. In our work, we have analyzed the use of artificial neural networks by tuning their hyper-parameters such as several hidden layers, learning rates, and activation function with varying depths. For a particular set of tuned hyper-parameters, the model performs better than other stages. ANN’s deliver good accuracy in breast cancer diagnosis but they tend to take longer training time with an expensive computational cost at each iteration. Our research involves training a classification model to increase the depth and then tuning the hyperparameters to achieve optimal results. ANN's, in particular, have been extensively used in disease classification. ANN’s have been used for the Classification and segmentation of diseases such as Alzheimer's, Breast Cancer, Lung Cancer, Brain Tumors, among other well-known diseases. ANN’s have been used extensively for disease detection and segmentation in imaging data. It has been used for the classification and segmentation of diseases like COVID-19[[12], [13]](https://www.zotero.org/google-docs/?tIy4Kg), Pest Detection[[14]](https://www.zotero.org/google-docs/?pkDONa), Alzheimer’s disease, Lung Cancer, Casava [[15]](https://www.zotero.org/google-docs/?l2b9t5)Leaf Disease, and other visually related diseases. Artificial Neural Networks have huge potential for classification, segmentation, and modeling and it’s on wide use is on the rise.

* 1. **Convolutional Neural Networks**

Convolutional Neural Networks were specifically designed to work with Image data. In the past decade, an immense amount of research has been done to improve Convolutional Neural Network models. With the rapid rise in the amount of image data being captured every day, with the ease of annotating, the significant improvement in the strength of graphical processing units has paced the research work on Convolutional Neural Networks. The research on Convolutional Neural Networks has been rapidly emerging and swiftly achieving state-of-art architectural results on various real-life problems. Among different types of deep neural networks, convolutional neural networks have been most extensively studied. Leveraging on the rapid growth in the amount of the annotated data and the significant improvements in the strengths of graphics processor units, the research on convolutional neural networks has been emerged swiftly and achieved state-of-the-art results on various tasks. Convolutional Networks do their job by analyzing the images through a kernel-like structure that traverses the whole image. This traversal of the kernel extracts important features from the image. The extracted information includes facial features, edge detection, object recognition, and other relevant features present in the image. These features help distinguish one class of images from another. This is what sets the Convolutional neural Networks different from other deep learning techniques. In our case, the Convolutional Neural Networks help in determining the classes i.e. Benign and Malignant. In our work, we have extensively used CNN architectures for extracting features to classify the mammography scan images in the MIAS and DDSM datasets. During a convolution, the size of the image is reduced. Various padding strategies are employed to keep the image size consistent. Feature extraction, dimensionality reduction, and classification are the three main phases of a convolutional neural network. The convolutional layers perform the task of feature extraction, pooling layers perform feature size reduction, and finally, the softmax layer classifies the image. Convolutional Neural Networks tend to outperform Dense Neural Networks(Feed Forward networks) as they can extract image-specific information through the use of multiple feature maps. In contrast, Densely connected Networks simply flatten the image and feed the flattened information to successive layers without any image-specific context.

1. **Disease detection using Artificial Intelligence**
   1. **Feature Extraction using deep learning**

A total of 10 pre-trained CNN architectures have been used for feature extraction apart from several other multi-layered CNN models. The pre-trained architectures were trained using the ImageNet dataset, which comprises roughly 20000+ pictures and the accompanying weights. This usage of models with pre-trained weights is known as transfer learning. This method uses a previously learned model to extract features from the present data with more efficiency. This provides a better method for generalizing and improving algorithms. The training images were passed on through the pre-trained models to extract features of varying dimensions. Feature extraction techniques have previously been used to extract specific features from Iris scans using Deep Convolutional Neural Networks such as ResNet [[16]](https://www.zotero.org/google-docs/?SOvX9r)and VGG[[17]](https://www.zotero.org/google-docs/?L56CDG) architectures. After experimenting with multiple-dimensional outputs, the best feature embedding/ feature vector was used for further processing and model building. To obtain a good feature embedding, dropout layers with appropriate dropout rates were used along with batch normalization. Transfer learning was used to retrain the trained models on image data and obtain feature embeddings that contain class and image-specific information.

* 1. **Dimension Reduction**

Before modeling, dimensionality reduction is a data preparation technique used on data. Dimension reduction is an important aspect of Feature extraction using deep learning. The features extracted through deep learning are often obtained by simply placing a dense layer of dimensions equal to that of the initial image. This may lead to good results but is computationally heavy to carry out This is why Feature embedding of appropriate Dimensions is to be used while extracting features from deep learning models. It might be done after cleaning and scaling the data and before training a prediction model. It is an important expect aspect when it comes to features extraction. The image data is analyzed through the Convolutional Neural Networks and then the extracted features are obtained by traversing several dense layers at the end. These are generally of very high dimensions[[18]](https://www.zotero.org/google-docs/?hJjDM2), which might produce significantly better output but have very high computational costs. This is the reason why dimension reduction is carried out while analyzing the images. The dimension of the images from the MIAS and DDSM datasets were reduced to (224,224) and (100,100) respectively. The Dense layer embedding was also dimensionally reduced to lessen the computational cost. The different architectures that were used were VGG16, VGG19, ResNet50, ResNet101, DenseNet121, DenseNet169 , InceptionV3, InceptionResNetV2, MobileNet and MobileNetV2 . In deep learning problems reducing the dimension where it is not needed, reduces the overall performance of the architecture whereas where higher dimensions create a problem, dimension reduction comes to save the performance and helps eliminate the curse of dimensionality. This method has helped train a huge amount of data in a comparatively lesser amount of time because of its flexibility. However, in other methods where higher dimensions lead to problems, a decrease in the dimensions would help eliminate the curse of dimensionality and lead to an overall increase in performance. Dimension reduction made it possible to train large amounts of data and test it effectively on unseen data.

* 1. **Prediction and Classification**

Our research was largely focused on two datasets: MIAS and DDSM mammography. Our major goal in both datasets was to categorize the mammography scan pictures into benign and malignant cancer abnormalities based on their grade of abnormality. The sigmoid layer was used to predict the end of the neural network as it was a binary classification task. There was a massive class imbalance in the DDSM dataset. To deal with such situations techniques such as Early stopping, stratifying the data, reducing learning-rate on the plateau were used. In the MIAS dataset, the images were too few to train and get good results. So we came up with a technique to produce six more images from a single image by rotating each version of the image by 60o. This increased dataset size by 6 times and now the data was good to train. We mainly used Adam optimizer for optimizing the classification process. This optimization technique showed good convergence in both datasets. As mentioned earlier various Convolutional Neural Networks and Transfer Learning techniques were used, apart from that some additional Dense layers were embedded for classification enhancement. The network was fed the input photos and labels, and the model was fitted with the data. A second validation dataset was used to validate the training data. The model's performance was nearly identical to that of state-of-the-art models.

In the next part of our experiment, various classifiers were used to train on features extracted from the above Deep Learning models. The Region of interest captured in the extracted features is fed into various classifiers, to classify the images into benign and malignant. The classifiers used were Artificial Neural networks(ANN), K-nearest neighbors (KNN) classifier, Support vector machines(SVM) classifier[[19], [20]](https://www.zotero.org/google-docs/?tuPDMz), Random Forest classifier, Adaboost classifier, and XGBoost classifier.

* 1. **Experimental Data**

The CBIS-DDSM dataset is the most widely used in machine learning and deep learning approaches for breast cancer detection. The dataset we utilized comprises pictures from the DDSM and CBIS-DDSM[[21]](https://www.zotero.org/google-docs/?dTmccs) databases and is freely available. By extracting the Region of Interest from the pictures, they were preprocessed and transformed to 299x299 images (ROIs). TensorFlow stores the data as tfrecords files. The dataset is separated into five tfrecords files and comprises 55,890 training instances, 14 percent of which are positive and 86 percent of which are negative.

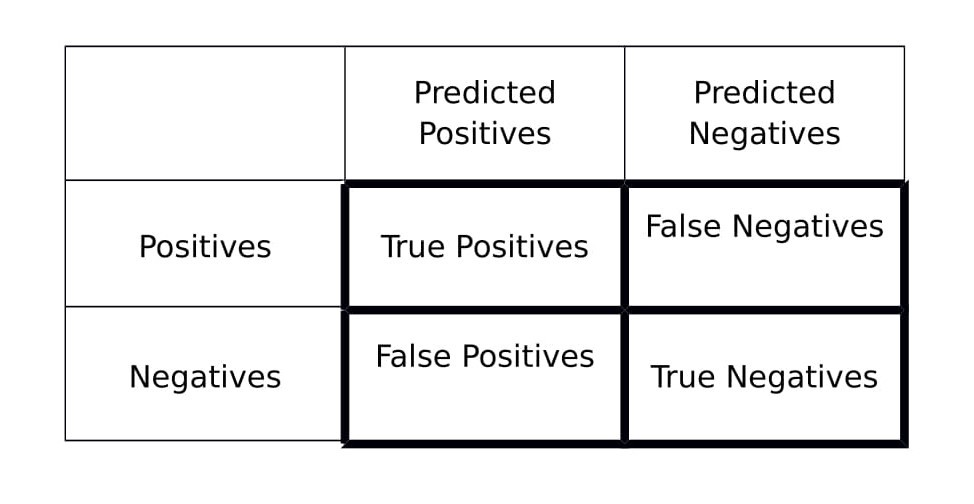
Because the tumors themselves comprise only a small percentage of the image of the entire breast [[5]](https://www.zotero.org/google-docs/?shzE89), early diagnosis of subclinical breast cancer on screening mammography is difficult as an image classification task. We have used image patches from CBIS-DDSM for Classification and tests on total mammograms for localization. We merge the training and testing dataset in CBIS-DDSM, and the data was further split into training, validation, and testing.

The Mammographic Image Analysis Society (MIAS) is a consortium of UK-based research groups dedicated to better understanding mammograms, and they've created a digital mammography database. The MIAS[[22]](https://www.zotero.org/google-docs/?PxJy3n) dataset consists of 322 images of size 1024\*1024. A Joyce-Loebl scanning microdensitometer, which is linear in the optical density range of 0-3.2 and represents each pixel with an 8-bit word, was used to digitize these images to a 50-micron pixel edge.

Our research work primarily focuses on implementing and tuning state-of-the-art models to attain higher Accuracy. Transfer learning has played a significant role in this field. Pretrained weights have led to better results, and techniques such as feature extraction make it possible to extract representation vectors or feature embeddings from extensive data and maintain high accuracy when combined with Machine Learning classification models.

* 1. **Performance Evaluation Measures**

With a 72:13:15 split, the MIAS dataset was divided into training, validation, and test data. Similarly, the DDSM dataset was divided into 64:16:20 training, validation, and test data, accordingly. The splitting was stratified to maintain class ratios uniformly through the process. Evaluation metrics are used to check the accuracy of our model before actually doing the work of prediction. In our experimentation work, we used the F1 score, Cohen-Kappa Score, AUC-ROC, precision, recall, and accuracy metrics. These are widely used metrics to interpret the performance of the model on multi-class data.

****

The above shown is a sample confusion matrix for binary Classification. To evaluate the results of our classification model. We require some values such as TP, FN, FP, and TN. True positive and true negatives are the observations that are correctly predicted and therefore shown in green. We want to minimize false positives and false negatives, so they are shown in red color.

True Positives (TP) - These are the correctly predicted positive values that mean that the actual class's value is yes and the value of the predicted class is also yes.

True Negatives (TN) - These are the correctly predicted negative values that mean that the actual class's value is no and value of the predicted class is also no.

False positives and false negatives. These values occur when your actual class contradicts the predicted class.

False Positives (FP) – When the actual class is no, and the predicted class is yes.

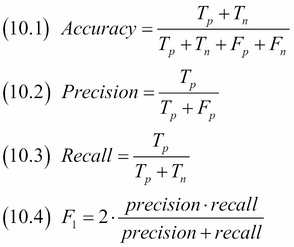
False Negatives (FN) – When actual class is yes, but predicted class is no.

1. Accuracy: It is the ratio of correctly predicted observations to the total number of observations.

2. Precision: Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

3. Recall (Sensitivity): Recall is the ratio of correctly predicted positive observations to all observations in the actual class.

4. F1:  F1 Score is the weighted average of Precision and Recall. Therefore, this Score takes both false positives and false negatives into account.



**5. Cohen’s Kappa coefficient:**

It is a statistic that is used to measure inter-rater reliability (and also intra-rater reliability) for qualitative (categorical) items. Cohen’s Kappa builds on the idea of measuring the concordance between the Predicted and the True Labels, which are regarded as two random categorical variables[[23]](https://www.zotero.org/google-docs/?FgVFt5). The kappa statistic uses the fact that by chance, the classifier will simply agree or disagree. The kappa statistics is the most widely used metric for categorical data measurement where there is no objective way of determining the probability of chance agreement between two or more observers. If the values are closer to 1, it is considered good, and values closer to 0 are uncertain. Cohen (Cohen, 1960) defined the kappa statistic as an agreement index and defined as the following,

k = Po — Pe

1 — Pe

**6. ROC -AUC Score:**

(Receiver Operator Characteristic) the curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the 'signal' from the 'noise.' The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve[[24]](https://www.zotero.org/google-docs/?MBR6uA). The higher the value of this metric, the better the performance of the classification model. It is evaluated by calculating the area under the ROC curve[[25]](https://www.zotero.org/google-docs/?gDkSa1).

Accuracy tends to hide vital classification errors for classes with few units since those classes are less relevant than the biggest ones. Therefore, Accuracy is most suited when we just care about single individuals instead of multiple classes. Hence, several metrics have to be used to interpret the results of a multi-class classification model[[26]](https://www.zotero.org/google-docs/?cZT73a).

* 1. **Experimental Results**

**This section contains the experimental results. We have worked on two datasets, namely: MIAS and DDSM. These are publicly available datasets and can be obtained from here. [MIAS, DDSM]. The experimental results have been presented below:**

1. **MIAS Dataset**
2. **PreTrained Networks (Transfer Learning)**

| **CLASSIFIER** | **TRAINING ACCURACY** | **VALIDATION ACCURACY** | **TEST ACCURACY** | **F1 Measure** | **KAPPA** | **ROC area** |
| --- | --- | --- | --- | --- | --- | --- |
| **ResNet50** | 0.988 | 0.962 | 0.9665 | 0.9666 | 0.9328 | 0.9924 |
| **VGG16** | 0.992 | 0.9803 | 0.9678 | 0.9679 | 0.9355 | 0.9945 |
| **VGG19** | 0.9912 | 0.9576 | 0.973 | 0.973 | 0.9457 | 0.997 |
| **Inception\_v3** | 0.98 | 0.84 | 0.8263 | 0.8262 | 0.65 | 0.9065 |
| **MobileNet** | 0.9938 | 0.903 | 0.9459 | 0.946 | 0.8914 | 0.9863 |
| **DenseNet169** | 0.9925 | 0.8848 | 0.8983 | 0.8984 | 0.7959 | 0.9606 |
| **DenseNet121** | 0.99 | 0.9 | 0.9034 | 0.9032 | 0.8042 | 0.9631 |
| **InceptionResNetV2** | 0.98 | 0.934 | 0.8937 | 0.8935 | 0.7846 | 0.9536 |
| **MobileNetV2** | 0.9995 | 0.9167 | 0.8919 | 0.8915 | 0.7814 | 0.9544 |
| **ResNet101** | 1 | 0.9682 | 0.9691 | 0.9691 | 0.9378 | 0.9943 |

**From the above experimentation, the method of transfer learning using pre-trained models showed impressive results. VGG-19 showed the best performance as compared to other models.**

1. **CNN Networks with different number of Layers**

| **CLASSIFIER** | **TRAINING ACCURACY** | **VALIDATION ACCURACY** | **TEST ACCURACY** | **F1 Measure** | **KAPPA** | **ROC area** |
| --- | --- | --- | --- | --- | --- | --- |
| **CNN 2 Layer** | 0.9699 | 0.9731 | 0.9678 | 0.9677 | 0.9344 | 1.0 |
| **CNN 3 Layer** | 0.9731 | 0.9719 | 0.971 | 0.9709 | 0.9411 | 0.9967 |
| **CNN 4 Layer** | 0.9712 | 0.9849 | 0.9836 | 0.9836 | 0.9667 | 0.9978 |
| **CNN 5 Layer** | 0.9851 | 0.9844 | 0.9932 | 0.9932 | 0.9863 | 0.999 |
| **CNN 6 Layer** | 0.9719 | 0.9918 | 0.9945 | 0.9945 | 0.9889 | 0.9998 |
| **CNN 7 Layer** | 0.99 | 1.00 | 0.999 | 0.999 | 0.998 | 1 |
| **CNN 8 Layer** | 0.993 | 0.9989 | 0.9974 | 0.9974 | 0.9948 | 1 |

**From the above experimentation, it can be seen that multi-layered CNN architectures out-performed the pre-trained models. The models appear to overfit on the train data, but the model performs equally well on the validation and test data. As the number of layers increases, the accuracy of the model increases.**

1. **Feature Extraction with PreTrained Models followed by Classification using Machine Learning models**
2. **VGG16**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 55.94 | 54.09 | 53.15 | 0.4117 | -0.0048 | 0.53 | 0.4929 |
| KNN | 69.69 | 51.48 | 51.99 | 0.579 | 0.0288 | 0.5199 | 0.5173 |
| SVM | 55.489 | 54.09 | 52.9 | 0.3953 | -0.013 | 0.529 | 0.4629 |
| Random Forest | 99.77 | 51.59 | 52.51 | 0.5109 | 0.0246 | 0.5251 | 0.5161 |
| AdaBoost | 59.09 | 52.5 | 53.54 | 0.5 | 0.0321 | 0.5354 | 0.5233 |
| XGBoost | 98.07 | 53.86 | 51.34 | 0.5082 | 0.01 | 0.5135 | 0.5 |

**The RandomForest and XGBoost classifier were overfitting on the training data. The accuracy scores were consistently in the range of 52-55 ; similar scores were seen in recall and precision.**

1. **VGG19**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 51.2 | 50 | 50 | 0.493 | 0.02 | 0.5 | 0.51 |
| KNN | 70.15 | 51.59 | 49.94 | 0.4961 | -0.01 | 0.499 | 0.4954 |
| SVM | 54.94 | 55 | 53.8 | 0.3764 | 0 | 0.538 | 0.2894 |
| Random Forest | 99.94 | 55.45 | 53.28 | 0.5189 | 0.04 | 0.5328 | 0.5246 |
| AdaBoost | 59.27 | 52.61 | 53.93 | 0.4921 | 0.034 | 0.5393 | 0.5274 |
| XGBoost | 99.29 | 56.02 | 51.22 | 0.5 | 0.012 | 0.5122 | 0.5 |

**The RandomForest and XGBoost classifier were overfitting on the training data. The accuracy scores were consistently in the range of 50-54 ; similar scores were seen in recall and precision.**

1. **ResNet 50**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 57.1 | 53.18 | 56.62 | 0.5 | 0.08 | 0.566 | 0.57 |
| KNN | 69.19 | 53.07 | 54.31 | 0.54 | 0.07 | 0.54 | 0.54 |
| SVM | 56.68 | 55 | 55.21 | 0.45 | 0.04 | 0.55 | 0.57 |
| Random Forest | 100 | 54.32 | 54.44 | 0.53 | 0.06 | 0.54 | 0.54 |
| AdaBoost | 59.66 | 54.32 | 55.86 | 0.53 | 0.08 | 0.56 | 0.55 |
| XGBoost | 98 | 54.2 | 53.41 | 0.53 | 0.05 | 0.53 | 0.53 |

**The RandomForest and XGBoost classifier were overfitting on the training data. The accuracy scores were consistently in the range of 53-56. The accuracy scores are better here as compared to previous results; similar scores were seen in recall and precision.**

1. **ResNet 101**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 58.75 | 53.86 | 52.63 | 0.51 | 0.02 | 0.526 | 0.52 |
| KNN | 70.2 | 50 | 50.06 | 0.5 | -0.01 | 0.5 | 0.5 |
| SVM | 57.9 | 55.11 | 53.67 | 0.47 | 0.02 | 0.54 | 0.52 |
| Random Forest | 100 | 58.52 | 55.7 | 0.54 | 0.09 | 0.56 | 0.55 |
| AdaBoost | 62 | 54.43 | 53.93 | 0.52 | 0.05 | 0.54 | 0.53 |
| XGBoost | 98.72 | 53.75 | 55.6 | 0.55 | 0.09 | 0.56 | 0.55 |

**The RandomForest and XGBoost classifier were overfitting the training data. The accuracy scores were consistently in the range of 50-56; similar scores were seen in recall and precision. SVM classifier doesn’t seem to overfit the training data.**

1. **MobileNetV2**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 57.89 | 53.4 | 56.11 | 0.544 | 0.1 | 0.56 | 0.56 |
| KNN | 69.93 | 52.88 | 51.22 | 0.51 | 0.02 | 0.51 | 0.51 |
| SVM | 59.1 | 53.94 | 54.57 | 0.45 | 0.03 | 0.55 | 0.55 |
| Random Forest | 100 | 56.36 | 55.47 | 0.54 | 0.08 | 0.55 | 0.55 |
| AdaBoost | 59.63 | 54.85 | 53.8 | 0.51 | 0.04 | 0.54 | 0.53 |
| XGBoost | 99.81 | 52.58 | 53.8 | 0.53 | 0.06 | 0.54 | 0.53 |

**The RandomForest and XGBoost classifier were overfitting on the training data. The accuracy scores were consistently in the range of 52-56 ; similar scores were seen in recall and precision. The ANN performs the best here with a 56% test accuracy.**

1. **MobileNet**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 57.76 | 54.69 | 53.66 | 0.45 | 0.02 | 0.52 | 0.53 |
| KNN | 70.2 | 52.88 | 49.81 | 0.49 | -0.02 | 0.5 | 0.49 |
| SVM | 57.34 | 54.39 | 53.28 | 0.42 | 0 | 0.53 | 0.5 |
| Random Forest | 100 | 55.6 | 52.9 | 0.51 | 0.03 | 0.53 | 0.52 |
| AdaBoost | 61.82 | 53.94 | 51.35 | 0.49 | -0.01 | 0.51 | 0.5 |
| XGBoost | 99.76 | 53.94 | 54.56 | 0.54 | 0.08 | 0.55 | 0.54 |

**The RandomForest, KNN, and XGBoost classifier were overfitting on the training data. The accuracy scores were consistently in the range of 52-56; similar scores were seen in recall and precision. The XGBoost performs the best here with a 54.5% test accuracy.**

1. **Inception V3**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 56.15 | 54.39 | 54.31 | 0.4 | 0.02 | 0.5431 | 0.564 |
| KNN | 69.26 | 52.12 | 50.97 | 0.5 | 0 | 0.51 | 0.5 |
| SVM | 55.97 | 54.09 | 53.8 | 0.41 | 0.01 | 0.54 | 0.52 |
| Random Forest | 91.01 | 50.14 | 52.9 | 0.51 | 0.03 | 0.53 | 0.52 |
| AdaBoost | 57.65 | 51.21 | 52.51 | 0.45 | -0.01 | 0.53 | 0.5 |
| XGBoost | 99.87 | 56.97 | 58.69 | 0.58 | 0.16 | 0.59 | 0.58 |

**The RandomForest and XGBoost classifier were overfitting the training data. The accuracy scores were consistently in the range of 52-56; similar scores were seen in recall and precision. The XGBoost classifier performs the best here with a 59% test accuracy. The accuracy attained is by far the best.**

1. **InceptionResNet v2**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 55.75 | 54.84 | 56.76 | 0.49 | 0.08 | 0.56 | 0.59 |
| KNN | 65.44 | 50.76 | 54.53 | 0.55 | 0.01 | 0.55 | 0.55 |
| SVM | 55.7 | 54.85 | 56.63 | 0.49 | 0.08 | 0.57 | 0.58 |
| Random Forest | 64.77 | 54.39 | 55.21 | 0.5 | 0.06 | 0.55 | 0.56 |
| AdaBoost | 56.47 | 54.7 | 56.49 | 0.49 | 0.08 | 0.56 | 0.59 |
| XGBoost | 81.81 | 49.24 | 55.73 | 0.55 | 0.09 | 0.56 | 0.55 |

**The RandomForest and XGBoost classifier were overfitting on the training data. The accuracy scores were consistently in the range of 55-56 ; similar scores were seen in recall and precision. The SVM classifier performs the best here with a 56% test accuracy.**

1. **DenseNet 169**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 55.35 | 55 | 53.79 | 0.39 | 0.002 | 0.53 | 0.52 |
| KNN | 68.89 | 51.21 | 51.73 | 0.51 | 0.02 | 0.52 | 0.51 |
| SVM | 55.3 | 55 | 53.8 | 0.39 | 0 | 0.54 | 0.52 |
| Random Forest | 99.97 | 53.03 | 57.79 | 0.56 | 0.13 | 0.58 | 0.57 |
| AdaBoost | 59.58 | 52.42 | 50.71 | 0.48 | -0.02 | 0.51 | 0.49 |
| XGBoost | 99.55 | 55.3 | 58.3 | 0.58 | 0.15 | 0.58 | 0.58 |

**The RandomForest and XGBoost classifier were overfitting the training data. The accuracy scores were consistently in the range of 50-58 ; similar scores were seen in recall and precision. The RandomForest classifier performs the best here with a 58% test accuracy.**

1. **DenseNet 121**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 55.99 | 53.63 | 53.79 | 0.4 | 0.004 | 0.54 | 0.52 |
| KNN | 70.84 | 49.09 | 51.48 | 0.51 | 0.01 | 0.51 | 0.51 |
| SVM | 55.22 | 54.85 | 53.93 | 0.38 | 0 | 0.54 | 0.57 |
| Random Forest | 99.83 | 55.61 | 55.21 | 0.54 | 0.08 | 0.55 | 0.55 |
| AdaBoost | 59.26 | 53.03 | 54.31 | 0.53 | 0.07 | 0.54 | 0.54 |
| XGBoost | 99.87 | 59.09 | 55.98 | 0.56 | 0.11 | 0.56 | 0.56 |

**The RandomForest and XGBoost classifier were overfitting the training data. The accuracy scores were consistently in the range of 52-56; similar scores were seen in recall and precision. The ANN performs the best here with a 56% test accuracy.**

1. **Xception**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 55.53 | 56.06 | 54.44 | 0.4 | 0.02 | 0.54 | 0.59 |
| KNN | 70.92 | 55.76 | 51.61 | 0.51 | 0.02 | 0.52 | 0.51 |
| SVM | 55.08 | 55 | 54.83 | 0.41 | 0.03 | 0.55 | 0.61 |
| Random Forest | 94.22 | 58.18 | 53.41 | 0.53 | 0.05 | 0.53 | 0.53 |
| AdaBoost | 57.92 | 55 | 53.93 | 0.5 | 0.04 | 0.54 | 0.53 |
| XGBoost | 99.81 | 62.88 | 62.16 | 0.62 | 0.23 | 0.62 | 0.62 |

The RandomForest and XGBoost classifier were overfitting the training data. The accuracy scores were consistently in the range of 52-62; similar scores were seen in recall and precision. The ANN performs the best here with a 62% test accuracy. The classification model obtained after feature extraction from the Xception pre-trained network gave the best test accuracy of the experiment .

**CBIS-DDSM Dataset**

1. **CNN Networks with different number of Layers**

| **CLASSIFIER** | **TRAINING ACCURACY** | **VALIDATION ACCURACY** | **TEST ACCURACY** | **F1 Measure** | **KAPPA** | **ROC area** |
| --- | --- | --- | --- | --- | --- | --- |
| **CNN 2 Layer** | 0.8896 | 0.8675 | 0.877 | 0.8277 | 0.117 | 0.6 |
| **CNN 3 Layer** | 0.9213 | 0.9026 | 0.9253 | 0.9196 | 0.6231 | 0.9463 |
| **CNN 4 Layer** | 0.9274 | 0.9144 | 0.921 | 0.909 | 0.5562 | 0.9448 |
| **CNN 5 Layer** | 0.9349 | 0.9194 | 0.9218 | 0.9118 | 0.5741 | 0.9573 |
| **CNN 6 Layer** | 0.9172 | 0.8902 | 0.8964 | 0.8672 | 0.3237 | 0.9371 |
| **CNN 7 Layer** | 0.93 | 0.90 | 0.9065 | 0.8842 | 0.4165 | 0.9568 |
| **CNN 8 Layer** | 0.92 | 0.8874 | 0.9042 | 0.882 | 0.4075 | 0.9388 |

**It can be observed from the above table that simple CNN models were able to achieve high performance overall. However, CNN 2 layer was not able to give good performance and was overfitting on specific class data, which resulted in a significant drop in ROC and Kappa values. CNN 5 Layer has the best performance overall in other networks.**

1. **PreTrained Networks (Transfer Learning)**

| **CLASSIFIER** | **TRAINING ACCURACY** | **VALIDATION ACCURACY** | **TEST ACCURACY** | **F1 Measure** | **KAPPA** | **ROC area** |
| --- | --- | --- | --- | --- | --- | --- |
| **ResNet50** | 0.937 | 0.9224 | 0.9292 | 0.9256 | 0.6576 | 0.9553 |
| **VGG16** | 0.9412 | 0.91 | 0.9058 | 0.8985 | 0.5238 | 0.9136 |
| **VGG19** | 0.9462 | 0.904 | 0.8966 | 0.896 | 0.5382 | 0.9018 |
| **Inception\_v3** | 0.9101 | 0.9053 | 0.9034 | 0.8973 | 0.5222 | 0.9117 |
| **MobileNet** | 0.927 | 0.9046 | 0.9113 | 0.9016 | 0.5298 | 0.925 |
| **DenseNet169** | 0.94 | 0.9265 | 0.9258 | 0.9219 | 0.64 | 0.9452 |
| **DenseNet121** | 0.9361 | 0.9238 | 0.924 | 0.9184 | 0.6185 | 0.9466 |
| **InceptionResNetV2** | 0.9862 | 0.9578 | 0.9585 | 0.9574 | 0.8078 | 0.9821 |
| **MobileNetV2** | 0.9451 | 0.9071 | 0.91 | 0.9006 | 0.5255 | 0.9152 |
| **ResNet101** | 0.9803 | 0.928 | 0.9224 | 0.9264 | 0.6699 | 0.9582 |

**It can be observed from the above table that InceptionResNetV2 and Resnet101 clearly overfit the training data. The InceptionResNetV2 model has the best performance with a test accuracy of 95.8% and Kappa score of 0.81 which highest among all the pre-trained models. Other models perform well with accuracy scores in the range of 89-92. A similar trend is seen in precision and recall scores.**

1. **Feature Extraction with PreTrained Models followed by Classification using Machine Learning models.**
2. **VGG16**

| **Model** | **Train\_accuracy** | **Val\_accuracy** | **Test\_accuracy** | **F1\_score** | **Kappa** | **Recall** | **Precision** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 98.51 | 87.19 | 87.92 | 0.8784 | 0.4875 | 0.8792 | 0.8776 |
| KNN | 88.75 | 87.81 | 86.85 | 0.8153 | 0.0939 | 0.8685 | 0.8726 |
| SVM | 93.34 | 89.83 | 88.86 | 0.8604 | 0.3322 | 0.8886 | 0.8847 |
| Random Forest | 100 | 88.37 | 87.39 | 0.8317 | 0.1819 | 0.8739 | 0.864 |
| AdaBoost | 89.99 | 88.649 | 87.92 | 0.8663 | 0.3944 | 0.8792 | 0.863 |
| XGBoost | 100 | 88.71 | 89.4 | 0.8781 | 0.4365 | 0.894 | 0.8823 |

**The RandomForest classifier, Artificial Neural Network, and XGBoost classifier were overfitting the training data. The test accuracy scores were consistently in the range of 87-90 . A similar trend was seen with precision and recall scores. The XGBoost classifier performs the best here with an 89.4% test accuracy although the Artificial neural network has the highest Kappa score of 0.49.**

1. **VGG19**

| Model | Train\_accuracy | Val\_accuracy | Test\_accuracy | F1\_score | Kappa | Recall | Precision |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 99.11 | 88.038 | 88.595 | 0.883 | 0.4708 | 0.886 | 0.88 |
| KNN | 89.01 | 87.7 | 87.66 | 0.8313 | 0.125 | 0.8766 | 0.8591 |
| SVM | 92.97 | 89.32 | 89.62 | 0.8708 | 0.3471 | 0.8962 | 0.8908 |
| Random Forest | 100 | 88.37 | 88.33 | 0.8476 | 0.2183 | 0.8833 | 0.8699 |
| AdaBoost | 90.2 | 88.04 | 88.86 | 0.8776 | 0.417 | 0.8886 | 0.8743 |
| XGBoost | 100 | 90.16 | 89.8 | 0.8831 | 0.4296 | 0.898 | 0.8854 |

**The RandomForest classifier, Artificial Neural Network, and XGBoost classifier were overfitting the training data. The test accuracy scores were consistently in the range of 87-90. A similar trend was seen with precision and recall scores. The XGBoost classifier performs the best here with an 89.8% test accuracy although the Artificial neural network has the highest Kappa score of 0.47.**

1. **ResNet50**

| Model | Train\_accuracy | Val\_accuracy | Test\_accuracy | F1\_score | Kappa | Recall | Precision |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 99.67 | 91.98 | 91.35 | 0.9073 | 0.578 | 0.9135 | 0.9066 |
| KNN | 91.02 | 88.62 | 88.25 | 0.867 | 0.3644 | 0.8825 | 0.868 |
| SVM | 93.1 | 90.12 | 90.4 | 0.8852 | 0.4452 | 0.904 | 0.9 |
| Random Forest | 100 | 88.71 | 88.49 | 0.8523 | 0.2686 | 0.8849 | 0.8769 |
| AdaBoost | 91.71 | 88.71 | 88.73 | 0.8798 | 0.4545 | 0.8873 | 0.8763 |
| XGBoost | 100 | 90.54 | 90.82 | 0.8969 | 0.5168 | 0.9082 | 0.9 |

**The RandomForest classifier, Artificial Neural Network, and XGBoost classifier were overfitting the training data. The test accuracy scores were consistently in the range of 88-92. A similar trend was seen with precision and recall scores. The Artificial Neural Network performs the best here with a 91.4% test accuracy which also has the highest Kappa score of 0.578.**

1. **ResNet 101**

| Model | Train\_accuracy | Val\_accuracy | Test\_accuracy | F1\_score | Kappa | Recall | Precision |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 99.76 | 92.58 | 92.13 | 0.9172 | 0.6169 | 0.9213 | 0.916 |
| KNN | 90.96 | 88.58 | 88.55 | 0.8643 | 0.3541 | 0.8858 | 0.8682 |
| SVM | 93.96 | 91.27 | 91.35 | 0.8984 | 0.4969 | 0.9165 | 0.9079 |
| Random Forest | 100 | 88.62 | 89.12 | 0.8585 | 0.27 | 0.8862 | 0.8909 |
| AdaBoost | 92.28 | 89.35 | 88.94 | 0.8838 | 0.4621 | 0.8894 | 0.88 |
| XGBoost | 100 | 91.56 | 90.58 | 0.8941 | 0.4874 | 0.9058 | 0.8968 |

**The RandomForest classifier, Artificial Neural Network, and XGBoost classifier were overfitting the training data. The test accuracy scores were consistently in the range of 88-92. A similar trend was seen with precision and recall scores. The Artificial Neural Network performs the best here with a 92.13% test accuracy which also has the highest Kappa score of 0.617. The SVM classifier also has comparable test accuracy of 91.35%.**

1. **MobileNet V2**

| Model | Train\_accuracy | Val\_accuracy | Test\_accuracy | F1\_score | Kappa | Recall | Precision |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 98.95 | 88.87 | 89.02 | 0.8829 | 0.4593 | 0.8903 | 0.8795 |
| KNN | 90.25 | 87.42 | 87.92 | 0.8516 | 0.262 | 0.8792 | 0.856 |
| SVM | 90.1 | 88.75 | 89.18 | 0.862 | 0.305 | 0.8918 | 0.8856 |
| Random Forest | 100 | 88.07 | 87.95 | 0.8417 | 0.1958 | 0.8795 | 0.8633 |
| AdaBoost | 90.32 | 86.75 | 87.21 | 0.858 | 0.3267 | 0.8721 | 0.8522 |
| XGBoost | 100 | 89.25 | 89.42 | 0.8733 | 0.3765 | 0.8942 | 0.8821 |

**The RandomForest classifier, Artificial Neural Network, and XGBoost classifier were overfitting the training data. The test accuracy scores were consistently in the range of 88-89. A similar trend was seen with precision and recall scores. All classifier models perform with comparable accuracy scores. The Artificial Neural network performs best with a Kappa score of 0.46.**

1. **MobileNet**

| Model | Train\_accuracy | Val\_accuracy | Test\_accuracy | F1\_score | Kappa | Recall | Precision |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 93.07 | 88.79 | 90.04 | 0.8864 | 0.461 | 0.9004 | 0.8898 |
| KNN | 91.71 | 87.35 | 88.07 | 0.8654 | 0.3649 | 0.88 | 0.8624 |
| SVM | 89.3 | 87.94 | 88.64 | 0.8505 | 0.2515 | 0.8864 | 0.8896 |
| Random Forest | 100 | 87.69 | 88.79 | 0.8606 | 0.3137 | 0.8879 | 0.8764 |
| AdaBoost | 90.5 | 85.56 | 87.36 | 0.8646 | 0.3815 | 0.8736 | 0.8597 |
| XGBoost | 100 | 88.07 | 89 | 0.8744 | 0.4038 | 0.89 | 0.8749 |

**The RandomForest classifier and XGBoost classifier were overfitting the training data. The test accuracy scores were consistently in the range of 88-90. A similar trend was seen with precision and recall scores. The Artificial Neural Network performs the best here with a 90.04% test accuracy which also has the highest Kappa score of 0.46.**

1. **InceptionV3**

| Model | Train\_accuracy | Val\_accuracy | Test\_accuracy | F1\_score | Kappa | Recall | Precision |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 89.48 | 89.81 | 88.93 | 0.8834 | 0.4674 | 0.8894 | 0.8801 |
| KNN | 91.64 | 88.92 | 89.18 | 0.8809 | 0.4366 | 0.8918 | 0.8784 |
| SVM | 89.72 | 90.12 | 90.01 | 0.8814 | 0.4159 | 0.9001 | 0.8911 |
| Random Forest | 100 | 89.86 | 90.19 | 0.8904 | 0.4762 | 0.9019 | 0.8909 |
| AdaBoost | 91.07 | 89.05 | 89.74 | 0.8904 | 0.4912 | 0.8974 | 0.8876 |
| XGBoost | 100 | 89.94 | 89.53 | 0.8878 | 0.478 | 0.8953 | 0.8849 |

**The RandomForest classifier and XGBoost classifier were overfitting the training data. The test accuracy scores were consistently in the range of 88-90. A similar trend was seen with precision and recall scores. All classifier models perform with comparable accuracy scores. The AdaBoost classifier performs best with a Kappa score of 0.49.**

1. **InceptionResNet V2**

| Model | Train\_accuracy | Val\_accuracy | Test\_accuracy | F1\_score | Kappa | Recall | Precision |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 87.06 | 87.03 | 87.83 | 0.8215 | 0 | 0.8784 | 0.7715 |
| KNN | 88.77 | 85.52 | 86.54 | 0.8363 | 0.1285 | 0.8654 | 0.8233 |
| SVM | 87.07 | 87.03 | 87.84 | 0.8251 | 0 | 0.8754 | 0.7715 |
| Random Forest | 100 | 85.13 | 86.58 | 0.8361 | 0.1257 | 0.8658 | 0.8231 |
| AdaBoost | 87.54 | 86.31 | 87.34 | 0.8366 | 0.11 | 0.8734 | 0.8285 |
| XGBoost | 99.9 | 85.47 | 86.45 | 0.8427 | 0.1802 | 0.8645 | 0.8316 |

**The RandomForest classifier and XGBoost classifier were overfitting the training data. The test accuracy scores were consistently in the range of 88-89. A similar trend was seen with precision and recall scores. All classifier models perform with comparable accuracy scores. The classifiers exhibit very low kappa scores indicating the extracted features were not very reliable.**

1. **DenseNet 169**

| Model | Train\_accuracy | Val\_accuracy | Test\_accuracy | F1\_score | Kappa | Recall | Precision |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 93.41 | 89.65 | 90.6 | 0.8997 | 0.5535 | 0.906 | 0.8983 |
| KNN | 90.87 | 88.32 | 88.73 | 0.874 | 0.3747 | 0.8873 | 0.8703 |
| SVM | 90.39 | 89.1 | 90.38 | 0.886 | 0.4179 | 0.9038 | 0.8943 |
| Random Forest | 100 | 89.49 | 90.07 | 0.8863 | 0.4279 | 0.9007 | 0.8879 |
| AdaBoost | 91.62 | 88.82 | 89.36 | 0.8902 | 0.4859 | 0.8936 | 0.8877 |
| XGBoost | 100 | 88.07 | 89 | 0.8744 | 0.4038 | 0.89 | 0.8749 |

**The RandomForest classifier and XGBoost classifier were overfitting the training data. The test accuracy scores were consistently in the range of 88-91. A similar trend was seen with precision and recall scores. All classifier models perform with comparable accuracy scores. The Artificial Neural network performs best with a Kappa score of 0.55 and a test accuracy of 90.6%.**

1. **DenseNet 121**

| Model | Train\_accuracy | Val\_accuracy | Test\_accuracy | F1\_score | Kappa | Recall | Precision |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 94.22 | 90.83 | 91.01 | 0.9075 | 0.5979 | 0.9101 | 0.9059 |
| KNN | 91.6 | 87.48 | 87.97 | 0.8635 | 0.3661 | 0.8797 | 0.8613 |
| SVM | 90.72 | 89.77 | 89.45 | 0.8758 | 0.4127 | 0.8945 | 0.8838 |
| Random Forest | 100 | 89.16 | 89.31 | 0.8756 | 0.4148 | 0.8931 | 0.8809 |
| AdaBoost | 92.04 | 89.38 | 89.18 | 0.8871 | 0.5037 | 0.8918 | 0.8843 |
| XGBoost | 100 | 90.78 | 90.61 | 0.8977 | 0.5361 | 0.9061 | 0.8975 |

**The RandomForest classifier and XGBoost classifier were overfitting the training data. The test accuracy scores were consistently in the range of 88-91. A similar trend was seen with precision and recall scores. All classifier models perform with comparable accuracy scores. The Artificial Neural network performs best with a Kappa score of 0.6 and a test accuracy of 91%.**

1. **Xception**

| Model | Train\_accuracy | Val\_accuracy | Test\_accuracy | F1\_score | Kappa | Recall | Precision |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANN | 89.97 | 89.04 | 89.04 | 0.8717 | 0.4151 | 0.8904 | 0.8793 |
| KNN | 90.38 | 87.65 | 88.1 | 0.856 | 0.3343 | 0.881 | 0.8663 |
| SVM | 89.15 | 88.6 | 88.77 | 0.8589 | 0.3399 | 0.8877 | 0.8875 |
| Random Forest | 100 | 88.82 | 89.49 | 0.8767 | 0.4371 | 0.8949 | 0.8867 |
| AdaBoost | 90.06 | 88.54 | 88.69 | 0.8748 | 0.4453 | 0.8869 | 0.8735 |
| XGBoost | 100 | 89.32 | 89.98 | 0.8874 | 0.497 | 0.8998 | 0.8906 |

**The RandomForest classifier and XGBoost classifier were overfitting the training data. The test accuracy scores were consistently in the range of 88-89. A similar trend was seen with precision and recall scores. All classifier models perform with comparable accuracy scores. The AdaBoost classifier performs best with a Kappa score of 0.445.**

1. **Discussion**

The goal of experimenting with various types of models, both pre-trained and custom-made, the classification of mammograms into Benign and Malignant was accomplished, and transfer learning was found to have the best overall performance. With the MIAS dataset the VGG-19 CNN architecture had the best performance - Train Accuracy : 99.12% , Validation Accuracy : 95.76% , Test Accuracy : 97.3% , F1 Score:0.973 , Cohen-Kappa Score : 0.9457 , AUC-ROC score: 0.997 . Among the custom n-layered models, the 7-layered CNN model worked best but with a very high overfitting trend due to over-augmentation - Train Accuracy: 99%, Validation Accuracy: 100%, Test Accuracy: 99.9%, F1 Score:0.999, Cohen-Kappa Score: 0.998, AUC-ROC score: 1.0. Standard classifiers (ANN, KNN, XGBoost, SVM, Random Forest, AdaBoost, and XGBoost ) gave satisfactory results on the test data with the XGBoost classifier trained on features extracted from the Xception model achieving 62.16% test accuracy.

With the CBIS-DDSM dataset the InceptionResNetV2 CNN architecture had the best performance - Train Accuracy : 98.62% , Validation Accuracy : 95.78% , Test Accuracy : 95.85% , F1 Score:0.9574 , Cohen-Kappa Score : 0.8078 , AUC-ROC score: 0.9821 . Among the custom n-layered models , 3-layered CNN model worked best - Train Accuracy : 92.13% , Validation Accuracy : 90.26% , Test Accuracy : 92.53% , F1 Score:0.9196 , Cohen-Kappa Score: 0.6231 , AUC-ROC score: 0.9463. Standard classifiers (ANN, KNN, XGBoost, SVM, Random Forest, AdaBoost and XGBoost ) gave satisfactory results on the test data with ANN classifier trained on features extracted from the ResNet101 model achieving 92.13% test accuracy .

The classifiers better performed with the DDSM dataset as compared to the MIAS dataset. The complex nature of the pre-trained models tends to learn complex features from the image that simple CNN architectures cannot achieve. The increasing depth of the layers in the models increases the networks’ efficiency and provides a research area to further explore the possibilities of patterns in models.

1. **Conclusion**

This paper highlights our work in various techniques on Breast Cancer classification using mammography scan images. We used transfer learning, features extraction through pre-trained models, hyperparameter tuning, and optimization. Different metrics were used to analyze the performances of the used models. The MIAS and DDSM breast cancer dataset was used for training and testing purposes. Our work includes a comparative study, with an in-depth analysis of various models. Our findings reveal that an overall 97% accuracy can be achieved in detecting the correct form of breast cancer with the help of mammograms. The future scope of our work includes tuning existing architectures which could lead to further development of finer models in the use of Breast Cancer detection from mammograms using Deep learning techniques.

**7. References**

[[1] S. Hadush, Y. Girmay, A. Sinamo, and G. Hagos, “Breast Cancer Detection Using Convolutional Neural Networks,” *ArXiv200307911 Cs*, Aug. 2020, Accessed: Jul. 23, 2021. [Online]. Available: http://arxiv.org/abs/2003.07911](https://www.zotero.org/google-docs/?UMgHRW)

[[2] F. Bray, J. Ferlay, I. Soerjomataram, R. L. Siegel, L. A. Torre, and A. Jemal, “Global cancer statistics 2018: GLOBOCAN estimates of incidence and mortality worldwide for 36 cancers in 185 countries,” *CA. Cancer J. Clin.*, vol. 68, no. 6, pp. 394–424, Nov. 2018, doi: 10.3322/caac.21492.](https://www.zotero.org/google-docs/?UMgHRW)

[[3] Y. Li and H. Chen, “A Survey of Computer-aided Detection of Breast Cancer with Mammography,” *J. Health Med. Inform.*, vol. 7, no. 4, 2016, doi: 10.4172/2157-7420.1000238.](https://www.zotero.org/google-docs/?UMgHRW)

[[4] K. Mendel, H. Li, D. Sheth, and M. Giger, “Transfer Learning From Convolutional Neural Networks for Computer-Aided Diagnosis: A Comparison of Digital Breast Tomosynthesis and Full-Field Digital Mammography,” *Acad. Radiol.*, vol. 26, no. 6, pp. 735–743, Jun. 2019, doi: 10.1016/j.acra.2018.06.019.](https://www.zotero.org/google-docs/?UMgHRW)

[[5] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh, “Deep Learning to Improve Breast Cancer Detection on Screening Mammography,” *Sci. Rep.*, vol. 9, no. 1, p. 12495, Dec. 2019, doi: 10.1038/s41598-019-48995-4.](https://www.zotero.org/google-docs/?UMgHRW)

[[6] J. R. Burt *et al.*, “Deep learning beyond cats and dogs: recent advances in diagnosing breast cancer with deep neural networks,” *Br. J. Radiol.*, p. 20170545, Apr. 2018, doi: 10.1259/bjr.20170545.](https://www.zotero.org/google-docs/?UMgHRW)

[[7] D. Wang, A. Khosla, R. Gargeya, H. Irshad, and A. H. Beck, “Deep Learning for Identifying Metastatic Breast Cancer,” *ArXiv160605718 Cs Q-Bio*, Jun. 2016, Accessed: May 17, 2021. [Online]. Available: http://arxiv.org/abs/1606.05718](https://www.zotero.org/google-docs/?UMgHRW)

[[8] D. Selvathi and A. Aarthy Poornila, “Deep Learning Techniques for Breast Cancer Detection Using Medical Image Analysis,” in *Biologically Rationalized Computing Techniques For Image Processing Applications*, vol. 25, J. Hemanth and V. E. Balas, Eds. Cham: Springer International Publishing, 2018, pp. 159–186. doi: 10.1007/978-3-319-61316-1\_8.](https://www.zotero.org/google-docs/?UMgHRW)

[[9] R. Agarwal, O. Díaz, M. H. Yap, X. Lladó, and R. Martí, “Deep learning for mass detection in Full Field Digital Mammograms,” *Comput. Biol. Med.*, vol. 121, p. 103774, Jun. 2020, doi: 10.1016/j.compbiomed.2020.103774.](https://www.zotero.org/google-docs/?UMgHRW)

[[10] K. Suzuki, Ed., *Artificial Neural Networks - Methodological Advances and Biomedical Applications*. InTech, 2011. doi: 10.5772/644.](https://www.zotero.org/google-docs/?UMgHRW)

[[11] S. Khan, N. Islam, Z. Jan, I. Ud Din, and J. J. P. C. Rodrigues, “A novel deep learning based framework for the detection and classification of breast cancer using transfer learning,” *Pattern Recognit. Lett.*, vol. 125, pp. 1–6, Jul. 2019, doi: 10.1016/j.patrec.2019.03.022.](https://www.zotero.org/google-docs/?UMgHRW)

[[12] H. Panwar, P. K. Gupta, M. K. Siddiqui, R. Morales-Menendez, and V. Singh, “Application of deep learning for fast detection of COVID-19 in X-Rays using nCOVnet,” *Chaos Solitons Fractals*, vol. 138, p. 109944, Sep. 2020, doi: 10.1016/j.chaos.2020.109944.](https://www.zotero.org/google-docs/?UMgHRW)

[[13] A. Subasi, A. Mitra, F. Ozyurt, and T. Tuncer, “Automated COVID-19 Detection from CT Images Using Deep Learning,” in *Computer-aided Design and Diagnosis Methods for Biomedical Applications*, 1st ed., V. Bajaj and G. R. Sinha, Eds. First edition. | Boca Raton : CRC Press, 2021.: CRC Press, 2021, pp. 153–176. doi: 10.1201/9781003121152-7.](https://www.zotero.org/google-docs/?UMgHRW)

[[14] M. Türkoğlu and D. Hanbay, “Plant disease and pest detection using deep learning-based features,” p. 16.](https://www.zotero.org/google-docs/?UMgHRW)

[[15] A. Ramcharan, K. Baranowski, P. McCloskey, B. Ahmed, J. Legg, and D. P. Hughes, “Deep Learning for Image-Based Cassava Disease Detection,” *Front. Plant Sci.*, vol. 8, p. 1852, Oct. 2017, doi: 10.3389/fpls.2017.01852.](https://www.zotero.org/google-docs/?UMgHRW)

[[16] K. He, X. Zhang, S. Ren, and J. Sun, “Deep Residual Learning for Image Recognition,” *ArXiv151203385 Cs*, Dec. 2015, Accessed: May 29, 2021. [Online]. Available: http://arxiv.org/abs/1512.03385](https://www.zotero.org/google-docs/?UMgHRW)

[[17] K. Simonyan and A. Zisserman, “Very Deep Convolutional Networks for Large-Scale Image Recognition,” *ArXiv14091556 Cs*, Apr. 2015, Accessed: May 29, 2021. [Online]. Available: http://arxiv.org/abs/1409.1556](https://www.zotero.org/google-docs/?UMgHRW)

[[18] D. Zhang, L. Zou, X. Zhou, and F. He, “Integrating Feature Selection and Feature Extraction Methods With Deep Learning to Predict Clinical Outcome of Breast Cancer,” *IEEE Access*, vol. 6, pp. 28936–28944, 2018, doi: 10.1109/ACCESS.2018.2837654.](https://www.zotero.org/google-docs/?UMgHRW)

[[19] The author is with the Department of Electrical Engineering and Computer Science, University of Toledo, OH 43606 USA and A. A. Bataineh, “A Comparative Analysis of Nonlinear Machine Learning Algorithms for Breast Cancer Detection,” *Int. J. Mach. Learn. Comput.*, vol. 9, no. 3, pp. 248–254, Jun. 2019, doi: 10.18178/ijmlc.2019.9.3.794.](https://www.zotero.org/google-docs/?UMgHRW)

[[20] E. H. Houssein, M. M. Emam, A. A. Ali, and P. N. Suganthan, “Deep and machine learning techniques for medical imaging-based breast cancer: A comprehensive review,” *Expert Syst. Appl.*, vol. 167, p. 114161, Apr. 2021, doi: 10.1016/j.eswa.2020.114161.](https://www.zotero.org/google-docs/?UMgHRW)

[[21] J. Wen *et al.*, “Convolutional neural networks for classification of Alzheimer’s disease: Overview and reproducible evaluation,” *Med. Image Anal.*, vol. 63, p. 101694, Jul. 2020, doi: 10.1016/j.media.2020.101694.](https://www.zotero.org/google-docs/?UMgHRW)

[[22] O. Ozdemir, R. L. Russell, and A. A. Berlin, “A 3D Probabilistic Deep Learning System for Detection and Diagnosis of Lung Cancer Using Low-Dose CT Scans,” *ArXiv190203233 Cs*, Jan. 2020, Accessed: Jul. 23, 2021. [Online]. Available: http://arxiv.org/abs/1902.03233](https://www.zotero.org/google-docs/?UMgHRW)

[[23] P. Ranganathan, C. Pramesh, and R. Aggarwal, “Common pitfalls in statistical analysis: Measures of agreement,” *Perspect. Clin. Res.*, vol. 8, no. 4, p. 187, 2017, doi: 10.4103/picr.PICR\_123\_17.](https://www.zotero.org/google-docs/?UMgHRW)

[[24] K. Feng, H. Hong, K. Tang, and J. Wang, “Decision Making with Machine Learning and ROC Curves,” *ArXiv190502810 Cs Econ Q-Fin Stat*, May 2019, Accessed: Jul. 23, 2021. [Online]. Available: http://arxiv.org/abs/1905.02810](https://www.zotero.org/google-docs/?UMgHRW)

[[25] J. B. Brown, “Classifiers and their Metrics Quantified,” *Mol. Inform.*, vol. 37, no. 1–2, p. 1700127, Jan. 2018, doi: 10.1002/minf.201700127.](https://www.zotero.org/google-docs/?UMgHRW)

[[26] M. Grandini, E. Bagli, and G. Visani, “Metrics for Multi-Class Classification: an Overview,” *ArXiv200805756 Cs Stat*, Aug. 2020, Accessed: Jul. 23, 2021. [Online]. Available: http://arxiv.org/abs/2008.05756](https://www.zotero.org/google-docs/?UMgHRW)