Breast Cancer Classification using Convolutional Neural Networks (CNNs)

Racha Ganesh[1], Vedanvita.G[2], Vyshnavi. B[3] and Samreen.S[4]
[1]Assoc. Professor, CVR College of Engineering/ECE Department, Hyderabad, India.
Email: rachaganesh@gmail.com
[2]UG Student, CVR College of Engineering/ECE Department, Hyderabad, India.
Email: vedanvita.gudavalli@gmail.com
[3]UG Student, CVR College of Engineering/ECE Department, Hyderabad, India.
Email: vaishnavibanna12@gmail.com
[4]UG Student, CVR College of Engineering/ECE Department, Hyderabad, India.
Email: samreensulthana2803@gmail.com

Abstract: **In the present world, women are facing many health issues and medical problems. Breast cancer is one among them. It is a common cancer in women, and one of the major causes of death among women around the world. The Invasive Ductal Carcinoma (IDC) is the most widespread type of breast cancer with about 80% of all diagnosed cases.**

The IDC is characterized by hard lumps with asymmetrical borders. The Invasive breast cancers spread from the origin into the adjoining breast tissue. On a mammogram, IDC typically appears like a mass with spikes radiating from the edges. Early accurate diagnosis plays an important role in choosing the right treatment plan and improving survival rate among the patients. However, due to the small size and low contrast (of lumps?) compared to the background of images, it is challenging and time-consuming for radiologists to make an independent and accurate assessment. Hence, there is a necessity to develop helpful automated tools to overcome these obstacles in the diagnostic performance of breast cancer.

The proposed system is a breast cancer classifier on an IDC dataset that can accurately classify a histology image as benign or malignant using Artificial Intelligence. This design is implemented by using an image classification technique with the help of Deep Learning using six layered Convolutional Neural Network (CNN) architecture to identify the breast cancer. This design is tested by using different Machine Learning Algorithms like, Random Forest, Gradient Boosting, Extra Trees, and Logistic Regression for comparative analysis in terms of accuracy.

Index Terms: **Convolutional Neural Network, Cancer, Machine Learning Algorithm, Breast Cancer, Deep Learning**

I. Introduction

Cancer is an ensemble of diseases with gigantic molecular miscellany between tumors of afflicted patients. Breast cancer is one of the leading causes of death for women worldwide and it is occurring more frequently in both developed and developing countries [1]. As time is a major factor in saving lives in the case of breast cancer, human resources, and technology are _essential to deliver prompt patient services in terms of screening, diagnosis, and treatment. However, tumor diagnosis is time-consuming and often challenging for radiologists, while examining medical images due to the presence of noise, artefacts, and complex structure.

Additionally, a growing number of patients adds to the radiologist's burden that often results in misdiagnosis of tumors. At present, Mammography, Magnetic Resonance Imaging (MRI), and Ultrasound are the most common medical imaging modalities, available and used for early detection of cancerous breast tumors [2]. In this regard, mammogram-based diagnosis outperforms symptoms-based diagnosis among other modalities.

Artificial Intelligence (AI) is a computer performing tasks commonly associated with human intelligence [3]. Humans are coding or programming a computer to act, reason, and learn. An algorithm or model is the code that tells the computer how to act, reason, and learn.

Machine Learning (ML) is a type of AI that is not explicitly programmed to perform a specific task but rather can learn iteratively to make predictions or decisions. The more data an ML model is exposed to, the better it performs over time.

Deep Learning (DL) is a subset of ML which uses artificial neural networks to model how the human brain processes information to learn by using huge amounts of data processing [4]. A well-designed and well-trained DL model can perform classification tasks and make predictions with high accuracy, sometimes exceeding human expert–level performance [5].

Integration of AI technology in cancer care could improve the accuracy and speed of diagnosis, aid clinical decision-making, and would lead to better health outcomes. This integration of breast cancer diagnosis will also improve the accuracy of all convolutional designs [6] AI-guided clinical care has the potential to play an important role in reducing health disparities, particularly in low-resource settings.[7] The proposed system is a breast cancer classifier on an IDC dataset from Kaggle that can accurately classify a histology image as benign or malignant. This design work aims to classify the image using Machine learning algorithms and Deep Learning algorithms.[8] Under Machine Learning Algorithms, one can use Random Forest, Gradient Boosting, Extra Trees, and Logistic Regression. Under Deep Learning, which is the focused area, one can use six layered Convolutional Neural Network (CNN) architecture to classify the image. Comparison between various algorithms is done based on their accuracy.[9]

The Early detection of breast cancer and classification of mammogram images with the help of different Deep Learning Classifiers is a major area of research. Mammographic mass detection is one of the most important areas of Computer Aided Diagnosis (CAD) and can be achieved by using DCNN as a feature extractor. The detection and classification of

Received on 31-10-2022, Revised on 08-11-2022, Accepted on 17-11-2022.

lesions in mammograms with deep learning and the comparison between various algorithms is done based on their accuracy. [10]

The section II gives the design methodology workflow of Machine Learning (ML) and Deep Learning (DL) algorithms. The section III gives the information about different Design algorithms used based on Machine Learning and Deep Learning concepts. The section IV gives the information about the result analysis and comparison of accuracy of different algorithms. The section V gives the information about the Conclusion followed by References.

II. Design Methodology

The proposed system is designed by using the design methodology workflow of Machine Learning (ML) and Deep Learning (DL) algorithms. The workflow design methodology is explained in Figure 1. The Figure 1 is used for understanding the architecture of the research done and it also gives the information about understanding the modules used in the designed system.[11]

The Raw image data is collected for Image data acquisition. The image data is analyzed, and the same data is applied to the Machine Learning algorithms workflow and the Deep Learning algorithms workflows. Both these algorithms use image data pre-processing, image partitioning, creating the models as per the algorithms and training these algorithms. Then these workflows will create the testing and evaluation of performance models.[12] After the testing, verification, and performance evaluation of these algorithms on the input image data, the results are compared and visualized for creating performance design model for ML and DL algorithms. [13]

Data Acquisition:

Data Acquisition is loading/importing the necessary Data into python workspace. Converting the normal image data like JPG or PNG or JEPG files etc. into python understandable data such as “nd-array” object of “numpy” module.

Data Analysis:

Data Analysis means understanding the basics of the data being loaded. To have knowledge of the number of images in each set i.e., training and testing sets, their statistics and graphical or structural differences. Hence, the Data pre-processing step can be easily utilized.

Data pre-processing:

Data preprocessing is preparing the data for giving it as input to the algorithm. Here re-scaling, resizing and reshaping are used to prepare the image for training the model.

Creating and training algorithms

Creating:

Instantiating the multiple algorithms which can accept input and produce output and supplying them with the train data to start the training.

Training:

Making the algorithm understand the training data and become intelligent in that concept.

Testing:

The process used to predict the outputs for the inputs in the test set. And understand the performance of trained models.

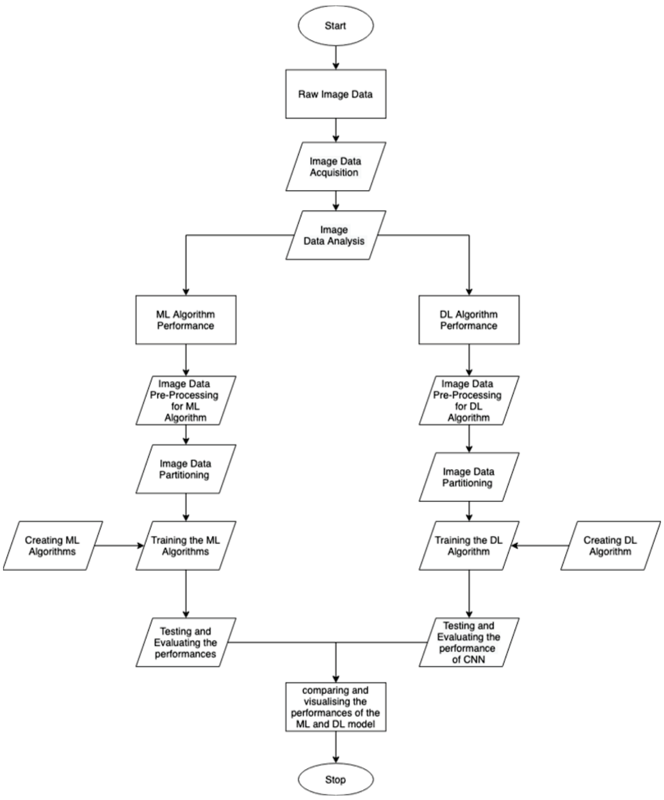

Figure 1. Proposed system Flow Chart.

III. Design Algorithms

The proposed system is designed by using the Machine Learning (ML) and Deep Learning (DL) algorithms [14]. The Workflow and algorithms for Machine learning are shown in Figure 2. The following are the different types of Machine Learning Algorithms [15] used for the proposed system design.

1) Logistic Regression Algorithm
2) Extra Tree Algorithm
3) Gradient Boosting Algorithm
4) Random Forest Algorithm

1). Logistic Regression

Logistic regression is one of the most popular Machine Learning algorithms, which comes under the Supervised Learning technique. It is used for predicting the categorical dependent variable using a given set of independent variables. Logistic regression predicts the output of a categorical dependent variable. Therefore, the outcome must be a categorical or discrete value. It can be either Yes or No, 0 or 1, True or False, etc. but instead of giving the exact value as 0 and 1, it gives the probabilistic values which lie between 0 and 1.

Logistic Function (Sigmoid Function)

The sigmoid function is a mathematical function used to map the predicted values to probabilities. It maps any real value into another value within a range of 0 and 1. The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.

Machine Learning Algorithms Workflow

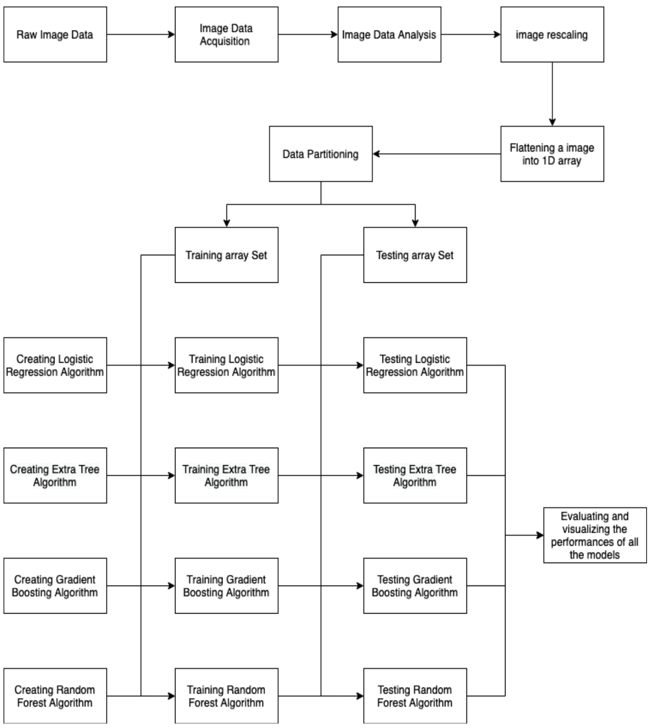

Figure 2. Machine Learning Algorithms Workflow

2). Random Forest

A Random Forest is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees and a technique called Bootstrap and Aggregation, commonly known as bagging. The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees. Random Forest has multiple decision trees as base learning models. The design randomly performs row sampling and feature sampling from the dataset forming sample datasets for every model. This part is called Bootstrap. Boosting combines weak learners into strong learners by creating sequential models such that the final model has the highest accuracy

3). Gradient Boosting

The term gradient boosting consists of two sub-terms, gradient and boosting. Gradient boosting re-defines boosting as a numerical optimization problem where the objective is to minimize the loss function of the model by adding weak learners using gradient descent. Gradient Descent is a first-order iterative optimization algorithm for finding a local minimum of a differentiable function. As gradient boosting is based on minimizing a loss function, different types of loss functions can be used resulting in a flexible technique that can be applied to regression, multi-class classification, etc.

Intuitively, gradient boosting is a stage-wise additive model that generates learners during the learning process (i.e., trees are added one at a time, and existing trees in the model are not changed). The contribution of the weak learner to the ensemble is based on the gradient descent optimization process. The calculated contribution of each tree is based on minimizing the overall error of the strong learner.

4. Extra Trees

Extra Trees Classifier is an ensemble learning method fundamentally based on decision trees. Extra Trees Classifier, like Random Forest, randomizes certain decisions and subsets of data to minimize over-learning from the data and overfitting.

Deep Learning Algorithm Workflow

The Workflow and algorithms for Deep Learning are shown in Figure 3. In this Deep Learning algorithm workflow, a 6-Layered Convolutional Neural Network (CNN) Architecture for the given IDC Dataset is used [16].

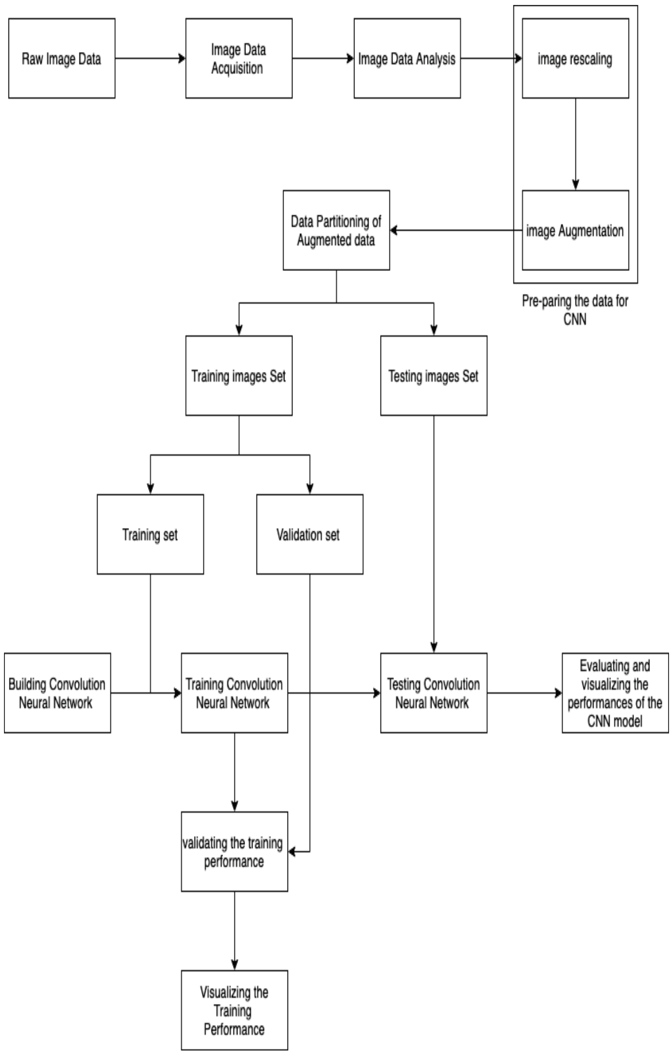

Figure 3. Deep Learning Algorithm Workflow

The image data acquisition and data analysis are done on the input raw data image as the first step. The data is prepared for Convolutional Neural Network (CNN) image using image resizing and image augmentation process [17]. This Augmented data information is partitioned into training image set and testing images set [18]. The training image set is used for building, training, and validating the performance of CNN. The testing image set is used for testing CNN along with the training set testing results [19]. At the end of this process, an evaluation of performance of CNN model is visualized [20].

CVR Journal of Science and Technology, Volume 23, December 2022
DOI: 10.32377/cvrjst2309

IV. Result Analysis

This section gives information about the results of various algorithms used for the classification. Figure 4 gives accurate information about Logistic Regression.

Logistic Regression:

```
In [22]:
# Instantiating Algorithm
logreg = LogisticRegression(solver='lbfgs',multi_class = 'multinomial',max_iter=800)

# Training
logreg.fit(x_train,y_train)

# Testing
log_pred = logreg.predict(X_test)

# Evaluate
print("Accuracy Score: ",accuracy_score(y_test,log_pred)*100,"%")
print("Classification Report: \n",classification_report(y_test,log_pred))
```

```
Accuracy Score:  66.3963963963964 %
Classification Report:
              precision    recall  f1-score   support

           0       0.66      0.67      0.67       559
           1       0.66      0.65      0.66       551

    accuracy                           0.66      1110
   macro avg       0.66      0.66      0.66      1110
weighted avg       0.66      0.66      0.66      1110
```

Figure 4. Logistic Regression Accuracy

The Figure 5 gives information about the Confusion Matrix of Logistic Regression.

```
In [23]:
sns.heatmap(confusion_matrix(y_true=y_test,y_pred=log_pred),annot=True,fmt="d")
plt.xlabel("acutal values")
plt.ylabel("predicted values")
plt.title("Confusion Matrix")
plt.show()
```

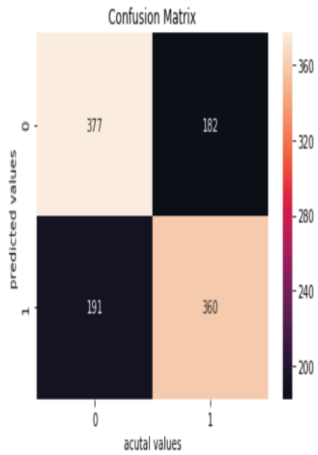

Figure 5. Confusion Matrix of Logistic Regression

Random Forest:

The Figure 6 gives the accuracy information about Random Forest Logistic Regression.

```
In [25]:
# Instantiating Algorithm
random_forest = RandomForestClassifier(random_state=1,class_weight="balanced")

# Training
random_forest.fit(x_train,y_train)

# Testing
RF_pred = random_forest.predict(X_test)

# Evaluate
print("Accuracy Score: ",accuracy_score(y_test,RF_pred)*100,"%")
print("Classification Report: ",classification_report(y_test,RF_pred))
```

```
Accuracy Score:  71.8018018018018 %
Classification Report:               precision    recall  f1-score   support

           0       0.70      0.76      0.73       559
           1       0.74      0.68      0.70       551

    accuracy                           0.72      1110
   macro avg       0.72      0.72      0.72      1110
weighted avg       0.72      0.72      0.72      1110
```

Figure 6. Random Forest Accuracy.

The Figure 7 gives information about the Confusion Matrix of Logistic Regression.

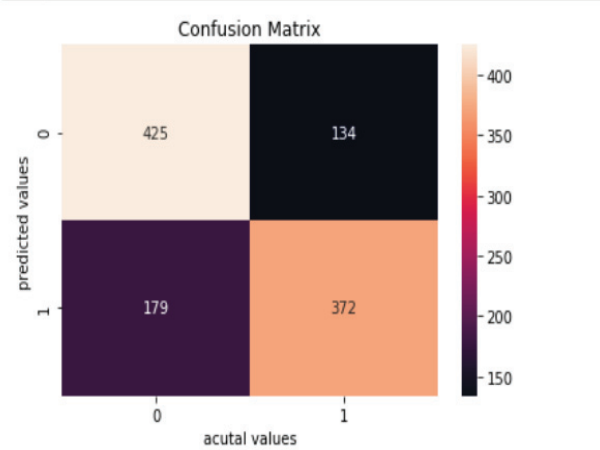

Figure 7. Confusion Matrix of Random Forest.

Gradient Boosting:

The Figure 8 gives accurate information about Gradient Boosting.

```
In [28]:
# Instantiating Algorithm
gradient_boost = GradientBoostingClassifier()

# Training
gradient_boost.fit(x_train,y_train)

# Testing
GBST_pred = gradient_boost.predict(X_test)

# Evaluate
print("Accuracy Score: ",accuracy_score(y_test,GBST_pred)*100,"%")
print("Classification Report: \n",classification_report(y_test,GBST_pred))
```

```
Accuracy Score:  77.29729729729729 %
Classification Report:
              precision    recall  f1-score   support

           0       0.78      0.76      0.77       559
           1       0.76      0.79      0.78       551

    accuracy                           0.77      1110
   macro avg       0.77      0.77      0.77      1110
weighted avg       0.77      0.77      0.77      1110
```

Figure 8. Gradient Boosting Accuracy

The Figure 9 gives information about the Confusion Matrix of Gradient Boosting.

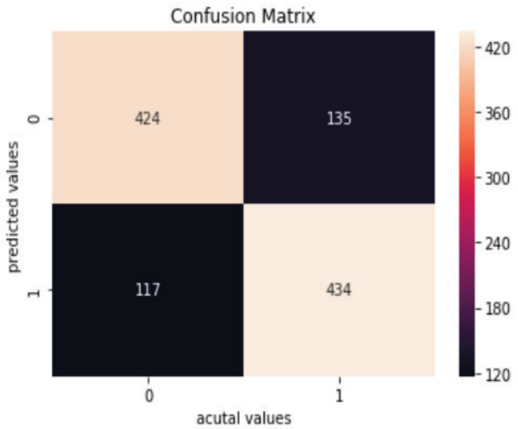

Figure 9. Confusion Matrix of Gradient Boosting.

Extra Trees:

The Figure 10 gives accuracy information about Extra Trees. Figure 11 gives information about the Confusion Matrix of Extra Trees.

```
In [24]:
# Instantiating Algorithm
extra_trees = ExtraTreesClassifier()

# Training
extra_trees.fit(x_train,y_train)

# Testing
ETS_pred = extra_trees.predict(X_test)

# Evaluate
print("Accuracy Score: ",accuracy_score(y_test,ETS_pred)*100,"%")
print("Classification Report: \n",classification_report(y_test,ETS_pred))

Accuracy Score:  73.06306306306305 %
Classification Report:
              precision    recall  f1-score   support

           0       0.72      0.77      0.74       559
           1       0.75      0.69      0.72       551

    accuracy                           0.73      1110
   macro avg       0.73      0.73      0.73      1110
weighted avg       0.73      0.73      0.73      1110
```

Figure 10. Extra Trees Accuracy

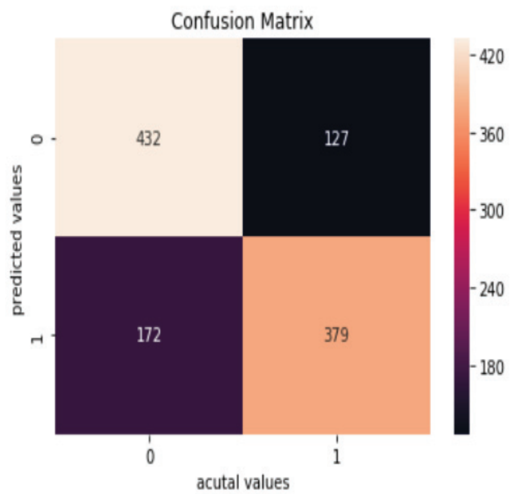

Figure 11. Confusion Matrix of Extra Trees.

Convolutional Neural Networks (CNNs):

The Figure 12 gives the information about building Convolutional Neural Networks (CNN) architecture.

Building CNN

```
In [11]:
import tensorflow as tf
class myCallback(tf.keras.callbacks.Callback):
    def on_epoch_end(self,epoch,logs={}):
        if(logs.get("val_accuracy") >= 0.80):
            print("Reached 80% accuracy so cancelling training!")
            self.model.stop_training = True
callbacks=myCallback()

# checkpoint
modelpath = "model.hdf5"
checkpoint = ModelCheckpoint(modelpath, monitor='val_accuracy', verbose=1, save_best_only=True, mode='max')

In [12]:
tf.keras.backend.clear_session()
K.clear_session() #clear session to make sure there is no overtraining
tf.random.set_seed(9)
np.random.seed(9)

input_shape = (xtrain.shape[1], xtrain.shape[2], 3)
batch_size = 64
num_classes = 2
epochs = 25
channelDim=-1
INIT_LR=1e-2 #initial learning rate

model = Sequential()

#first layer
model.add(SeparableConv2D(32, (3,3), padding="same",input_shape=input_shape))
model.add(Activation("relu"))
model.add(BatchNormalization(axis=channelDim))
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(SpatialDropout2D(0.2))
```

Figure 12. CNN Architecture.

The Table 1 gives comparison of accuracy of different algorithms for Breast Cancer Classification Analysis on different images.

TABLE I.
COMPARISON OF ALGORITHMS

S. No.	ALGORITHM	ACCURACY
1	Logistic Regression	66.39 %
2	Random Forest	71.80 %
3	Gradient Boosting	77.29 %
4	Extra Trees	73.06 %
5	Convolutional Neural Networks (CNNs)	80.36 %

V. CONCLUSIONS

The goal of novel study here was to comprehensively summarize and build the existing prior research and evaluate the performance of machine learning and deep learning methods in the task of distinguishing between benign or malignant lesions. This design shows the accuracy of 66.39 % for Logistic Regression, 71.80% accuracy for Random Forest, 77.29% accuracy for Gradient Boosting, 73.06% for Extra stress for the selected dataset. Although these results are promising, it is acknowledged that the test set used in our experiments is relatively small and our results require further clinical validation.

Typically, screening mammography is only the first step in a diagnostic pipeline, with the radiologist making a final determination and decision to biopsy only after recall for additional diagnostic mammogram images and possible ultrasound. However, in the study, a hybrid model including both a neural network and expert radiologists outperformed individually, suggesting that the use of such a model could improve radiologist sensitivity for breast cancer detection with an accuracy of 80.36%. Hence, this proposed approach may aid clinical specialists in diagnosing and treatment planning at an early stage.

REFERENCES

[1] R. L. Siegel, K. D. Miller, and A. Jemal, "Cancer statistics, 2019," CA, Cancer J. Clinicians, vol. 69, no. 1, pp. 7–34, 2019.

[2] C. J. D'Orsi et al., ACR BI-RADS Atlas, Breast Imaging and Data System. Reston, VA, USA: American College of Radiology, 2013.

[3] K. G. A. Gilhuijs, M. L. Giger, and U. Bick, "Computerized analysis of breast lesions in three dimensions using dynamic magnetic-resonance imaging," Med. Phys., vol. 25, no. 9, pp. 1647–1654, 1998.

[4] T. Kyono, F. J. Gilbert, and M. van der Schaar, "MAMMO: A deep learning solution for facilitating radiologist-machine collaboration in breast cancer diagnosis," 2018, arXiv:1811.02661. [Online].

[5] T. Kooi et al., "Large scale deep learning for computer aided detection of mammographic lesions," Med. Image Anal., vol. 35, pp. 303–312, Jan. 2017.

[6] L. Shen, "End-to-end training for whole image breast cancer diagnosis using an all convolutional design," 2017, arXiv:1711.05775.

[7] K. J. Geras et al., "High-resolution breast cancer screening with multiview deep convolutional neural networks," 2017

[8] N. Wu et al., "Breast density classification with deep convolutional neural networks," in Proc. ICASSP, Apr. 2018, pp. 6682–6686.

[9] W. Lotter, G. Sorensen, and D. Cox, "A multi-scale CNN and curriculum learning strategy for mammogram classification," in Proc. DLMIA, 2017, pp. 169–177.

[10] D. Ribli, A. Horváth, Z. Unger, P. Pollner, and I. Csabai, "Detecting and classifying lesions in mammograms with deep learning," Sci. Rep., vol. 8, Mar. 2018, Art. no. 4165.

[11] M. L. Giger, N. Karssemeijer, and J. A. Schnabel, "Breast image analysis for risk assessment, detection, diagnosis, and treatment of cancer," Annu. Rev. Biomed. Eng., vol. 15, pp. 327–357, Jul. 2013.

[12] J. Wu et al., "Identifying relations between imaging phenotypes and molecular subtypes of breast cancer: Model discovery and external validation," J. Magn. Reson. Imag., vol. 46, no. 4, pp. 1017–1027, 2017.

[13] B. Sahiner et al., "Deep learning in medical imaging and radiation therapy," Med. Phys., vol. 46, no. 1, pp. e1–e36, Jan. 2019.

[14] W. L. Bi et al., "Artificial intelligence in cancer imaging: Clinical challenges and applications," CA, Cancer J. Clinicians, vol. 69, no. 2, pp. 127–157, Mar./Apr. 2019.

[15] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in Proc. Adv. Neural Inf. Process. Syst., 2012, pp. 1097–1105.

[16] H. Harvey et al., "The role of deep learning in breast screening," Current Breast Cancer Reports, vol. 11, no. 1, pp. 17–22, 2019.

[17] A. S. Becker, M. Marcon, S. Ghafoor, M. C. Wurnig, T. Frauenfelder, and A. Boss, "Deep learning in mammography: Diagnostic accuracy of a multipurpose image analysis software in the detection of breast cancer," Investigative Radiol., vol. 52, no. 7, pp. 434–440, 2017.

[18] N. Antropova, B. Q. Huynh, and M. L. Giger, "A deep feature fusion methodology for breast cancer diagnosis demonstrated on three imaging modality datasets," Med. Phys., vol. 44, no. 10, pp. 5162–5171, 2017.

[19] A. Gastounioti, A. Oustimov, M.-K. Hsieh, L. Pantalone, E. F. Conant, and D. Kontos, "Using convolutional neural networks for enhanced capture of breast parenchymal complexity patterns associated with breast cancer risk," Acad. Radiol., vol. 25, no. 8, pp. 977–984, Aug. 2018.

[20] D. Truhn, S. Schrading, C. Haarburger, H. Schneider, D. Merhof, and C. Kuhl, "Radiomic versus convolutional neural networks analysis for classification of contrast-enhancing lesions at multiparametric breast MRI," Radiology, vol. 290, no. 3, pp. 290–297, 2019.