CHAPTER

4 TRANSFER LEARNING AND SUPERVISED CLASSIFIER BASED PREDICTION MODEL FOR BREAST CANCER

Md. Nuruddin Qaisar Bhuiyan, Md. Shamsujjoha, Shamim H. Ripon, Farhin Haque Proma, Fuad Khan
Department of Computer Science and Engineering, East West University, Dhaka, Bangladesh

4.1 INTRODUCTION

There are several methods available for diagnosis of breast cancer such as breast exam, mammography, breast ultrasound, and biopsy [1, 2]. Of these, biopsy is the only way to detect the presence of breast cancer. The most common biopsy techniques are core needle biopsy, fine needle biopsy, and surgical open biopsy. In the biopsy procedure, breast tissue samples are collected and examined under the microscope by pathologists. The whole procedure is based on visual inspection by pathologists. This is a time consuming, costly task and it requires attention of the pathologists examining the tissue. Histopathology image analysis also depends on the experience of the pathologist, which is costly and requires a huge amount of time [3].

As a result, there is a pressing need for an automated system that can differentiate between the cancerous tissue (malignant tissue) and the noncancerous tissue (benign tissue) and help the pathologists to make the diagnosis process into an easy and time efficient task, and consequently the pathologist can focus on more difficult cases.

A significant amount of research work has already been undertaken to build a computer aided system to automate the classification of benign and malignant tumor tissue images using an image dataset consisting of different sizes of images. To build an automated system, we used one of the largest datasets of breast cancer images called *BreaKHis* containing 7909 images of benign and malignant tissue at four different magnification factors. Convolution neural networks (ConvNets) are a state-of-the-art technique for image classification. There are many available convolution networks released by renowned organizations and institutions. Some architecture examples are ResNet-50, Inception V3, Inception ResNet V2, Xception etc. These ConvNets are very deep and are trained on millions of images. Training these ConvNets from scratch requires a significant amount of time and can cause overfitting since the number of training images we have is not big enough. For this reason, in this paper, four pretrained ConvNets were used as fixed feature extractors to extract the feature from the benign and malignant tissue images. To reduce the dimension of extracted features from different ConvNet

Big Data Analytics for Intelligent Healthcare Management. https://doi.org/10.1016/B978-0-12-818146-1.00004-0

models, a dimensionality reduction algorithm PCA was applied. Then to perform classification, three different classifiers logistic regression (LR), support vector machine (SVM), and K-nearest neighbor (K-NN) were used. This model could help the pathologist have a preliminary idea of to what class a breast tissue image belongs to for example, to benign or malignant. After that step, it will be easy for the pathologists to diagnose the image to the predicted class or not.

4.2 RELATED WORK

A huge amount of research work has been done for the automatic prediction of the presence of breast cancer using different datasets. In paper [3], using the BreaKHis image dataset to classify the images into two classes (benign and malignant), the authors used six different feature extractors with four different classifiers and reported accuracy ranges from 80% to 85%. The authors of paper [4], using different pretrained ConvNet models running thousands of epochs, reported the highest accuracy of 99.8%. In paper [5], the authors reported accuracy ranges from 81% to 90% using DECAF features with other classifiers and compared their work to other works. In paper [6], the authors showed a comparative study of different machine learning techniques on breast cancer FNA biopsy data and the K-NN with Euclidean distance approach showed a prediction accuracy of 100% with the K value of 5, 10, and 11, and it also showed the same accuracy using cityblock distance with a K value of 13. In the paper [7], the authors applied different machine learning algorithms to the Wisconsin breast cancer dataset and analyzed the performance. They reported an accuracy close to 100% and SVM gaves an accuracy of 100%. In paper [8], the authors used ResNet-50 and VGG16 and reported an accuracy of 89% and 84%, respectively. In paper [9], the author used different machine learning algorithms and reported an accuracy of 98.8% and 96.33%m respectively, using SVM on two different datasets. In paper [10], the authors reported their best accuracy of 99.038% using MLP on the Wisconsin breast cancer dataset. A huge amount of research work has been done in the area of cancer with different supervised and semisupervised classification, clustering, and feature detection methods of biomedical images [11–20], optimization, and information security techniques of medical data [21–25] helping to make computer-aided medical systems.

4.3 DATASET AND METHODOLOGIES

Dataset: In this work, we have used the *BreaKHis* [3] breast cancer histopathological image dataset. This dataset contains about 7909 RGB images of benign and malignant tissue at four magnification factors (40 ×, 100 ×, 200 ×, 400 ×) (Fig. 4.1, Table 4.1).

4.3.1 CONVOLUTION NEURAL NETWORKS (CNNS/CONVNETS)

Convolution neural networks are the state-of-the-art models for image classification and they are very similar to ordinary neural networks except they have some extra layers. There are three main building blocks of convolution neural networks: *Convolution Layer, Pooling Layer, and Fully Connected Layer*. These terms are described further below.

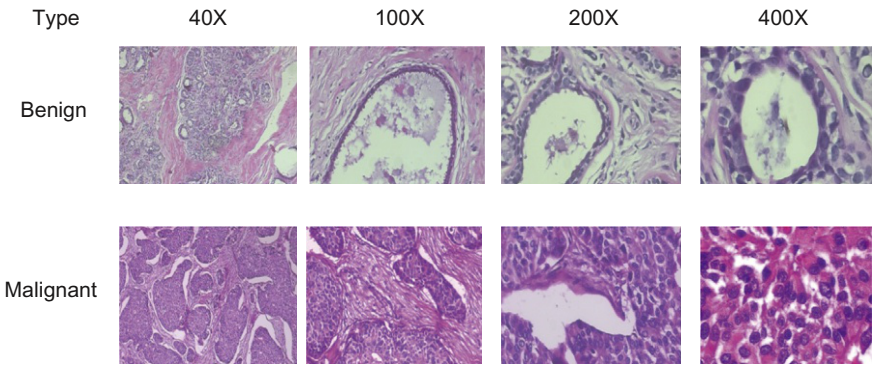

FIG. 4.1

Major two types of images with four magnification factors.

Table 4.1 Image Distribution by Magnification Factor and Class

Magnification Factor	# of Benign	# of Malignant	Total Images
40×	625	1370	1995
100×	644	1437	2081
200×	623	1390	2013
400×	588	1232	1820
Total images	2480	5429	7909
# of patients	24	58	82

Based on *F.A. Spanhol, L.S. Oliveira, C. Petitjean, L. Heutte, A dataset for breast cancer histopathological image classification, in IEEE Trans. Biomed. Eng., 63 (2016) 1455–1462, doi: 10.1109/TBME.2015.2496264.*

- *Convolution Layer*: In this layer, a set of filters slide in the direction of height and width of the input image, compute dot products, and produce two-dimensional activation maps for each of the filters. There are several activation functions that are applied to the activation maps to add nonlinearity such as RELU, Sigmoid etc.
- *Pooling Layer*: With a specified filter, strides, and pooling method, value from the activation maps is extracted. There are several pooling methods such as max pooling, min pooling, and average pooling. For example, if the filter size is 2×2 and strides is 2, and pooling method is max, then for each 2×2 matrix in the activation map, the output will be the maximum value within that 2×2 cell and then the filter will head to the next 2×2 cell as the specified strides is 2.
- *Fully Connected Layer*: In this layer, a number of hidden layers with specified neurons and activation functions are declared and the flattened feature of the previous stacked convolution and pooling layers are passed through this fully connected layer. More details about ConvNets can be found here [26] (Fig. 4.2).

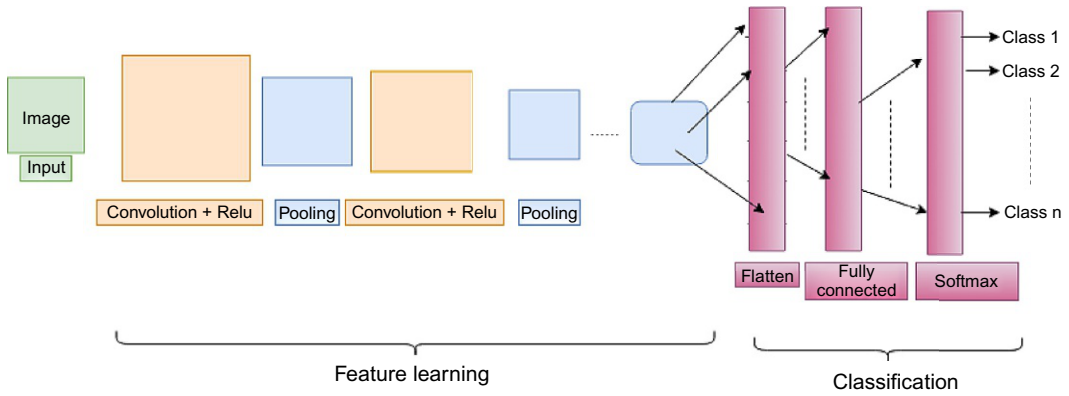

FIG. 4.2

Sample convolution network architecture.

Data from Mathworks.com, Convolutional Neural Network, 2018. Available from: https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html. Accessed 10 June 2018.

In this work, four pretrained ConvNets architectures were used: ResNet50 [27], Inception V3 [28], InceptionResnetV2[29], and Xception [30] with their default parameter settings with an average pooling implemented in *keras* [31] deep learning library.

4.3.1.1 Transfer learning and convolution networks

Transfer learning is a machine learning method that allows the use of a model trained on a task to perform another task. Since the convolution architectures released by different organizations trained on ImageNet databases containing 1.2 million images from 1000 categories is very large, training these types of architectures for custom datasets is not practical because datasets are not large enough in practice. So instead of training the whole network, these pretrained networks are used. There are ways of using a pretrained a convolution network that is, doing transfer learning using convolution network. One of them is using the pretrained convolution network as fixed feature extractors [32].

4.3.1.2 Convolution networks as fixed feature extractors

To use a convolution network as a feature extractor, remove the last fully connected layer of a pretrained convolution network and then use the rest of the architecture as a fixed feature extractor for the custom dataset. Then the extracted features can be used for other purposes [32] (Fig. 4.3).

4.3.1.3 Dimensionality reduction and principle component analysis (PCA)

It is difficult to train a learning algorithm with a higher dimensional data. Here comes the importance of dimension reduction. Dimensionality reduction is a method of reducing the original dimension of data to a lower dimension without much loss of information. Dimension reduction techniques have two components. One is feature selection and the other is feature extraction. Feature selection is responsible for selecting the subset of original attributes with specified parameters and feature extraction is responsible for projecting the data into a lower dimensional space that is forming a new dataset with selected attributes [33]. PCA is one of the popular dimension reduction algorithms that uses the orthogonal

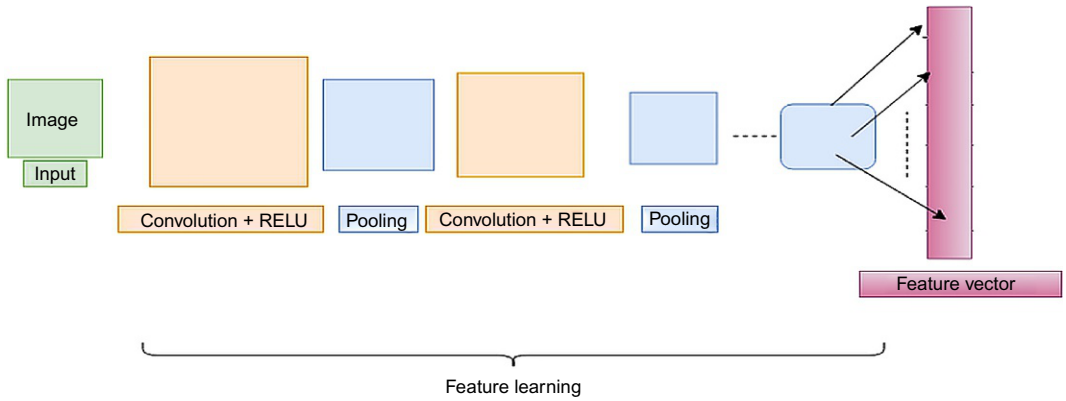

FIG. 4.3

Convolution network as feature extractor.

Data from Mathworks.com, Convolutional Neural Network, 2018. Available from: https://www.mathworks.com/solutions/deep-learning/convolutional-neural-network.html. Accessed 10 June 2018.

linear transformation to project the dataset to a new coordinate system such that the highest variance by some projection of data lies on the first coordinate called the first principle component and second largest variance lies on the second coordinate called the second principle component and so on.

PCA steps [34]:

1. Calculate covariance matrix.
2. Calculate eigenvalue and eigenvectors from covariance matrix.
3. Select K largest eigenvalues where K is the dimension of new subspace.
4. Calculate projection matrix from K selected eigenvalues.
5. Transform the dataset through projection matrix to form new dataset of K dimension.

4.3.1.4 Supervised machine learning

In supervised machine learning, both input and output pairs are used to train the learning algorithm and it is the task of the algorithm to learn the mapping function from input to output very well so that when a new input comes, the function can map the input to the output [35]. Three supervised machine learning algorithms were used in this work for classification purpose: LR, SVM, and K-NN. These algorithms are described further below.

- LR is borrowed from the field of statistics and named after the logistic function. It is the most popular method for binary classification problems. Logistic function is also known as sigmoid function, which is an s-shaped curve that transforms any input to a value between 0 and 1.

$$y = 1/(1+e^{-x}) = e^{x}/(1+e^{x})$$

Here, e is the base of the natural logarithm and x is the input and y is the output.

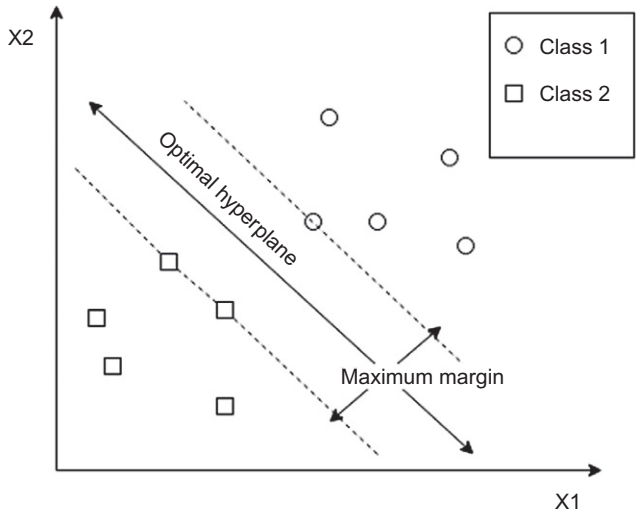

FIG. 4.4

SVM separates two classes, keeping the maximum margin.

Data from Docs.opencv.org, Introduction to Support Vector Machines—OpenCV 2.4.13.6 Documentation, 2018. Available from: https://docs.opencv.org/2.4/doc/tutorials/ml/introduction_to_svm/introduction_to_svm.html, Accessed 10 June 2018.

An example logistic regression equation,

$$Y = e^{x}(b0 + b1*x)/(1 + e^{x}(b0 + b1*x))$$

Here Y is the predicted output, $b0$ is the bias, and $b1$ is the coefficient of input x.
Thus, for every input, the logistic equation learns the coefficient and uses these learned coefficients for prediction when an unknown input arrives [36].

- SVM is a supervised machine learning algorithm and it can be used for classification problems. It separates the data points of different classes by hyperplane that maximizes the distance (also called margin) of the nearest point of each class from the hyperplane, shown in Fig. 4.4. SVM is also called maximal margin classifier [37].

- K-NN algorithm requires no learning. It simply stores the whole dataset and when a new instance comes, it measures the distance of k–data points around it and labels the new instance as the same label of the closest instance, illustrated in Fig. 4.5. K-NN is also called instance based learning [38].

4.4 PROPOSED MODEL

Fig. 4.6 demonstrates the overall architecture of the proposed model. In the proposed model, the images at each of the magnification factors are passed through four pretrained ConvNets (ResNet-50, Inception V2, Inception ResNet V2, and Xception). The outputs of these ConvNets are the image features. Then on the flattened image features, PCA is applied to reduce the dimension of the feature vector. Then

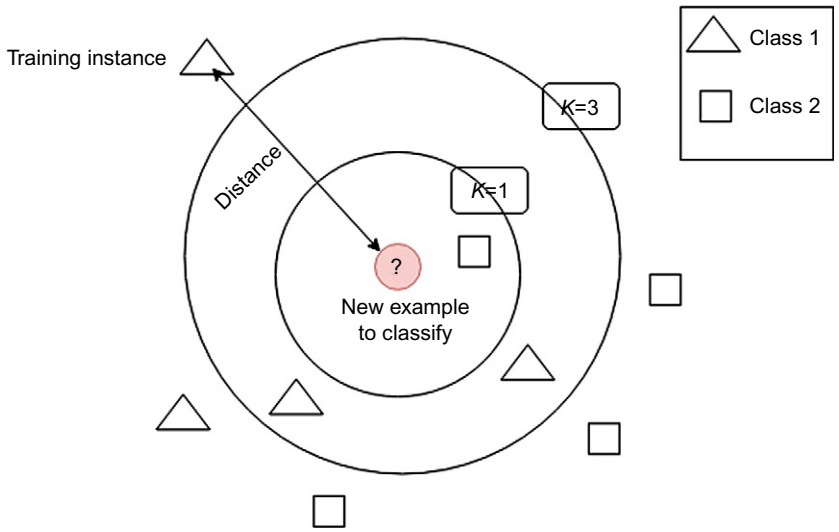

FIG. 4.5

K-NN to classify new instance.

Data from Medium, A Quick Introduction to K-Nearest Neighbors Algorithm, 2018. Available from: https://medium.com/@adi.bronshtein/a-quick-introduction-to-k-nearest-neighbors-algorithm-62214cea29c7, Accessed 10 June 2018.

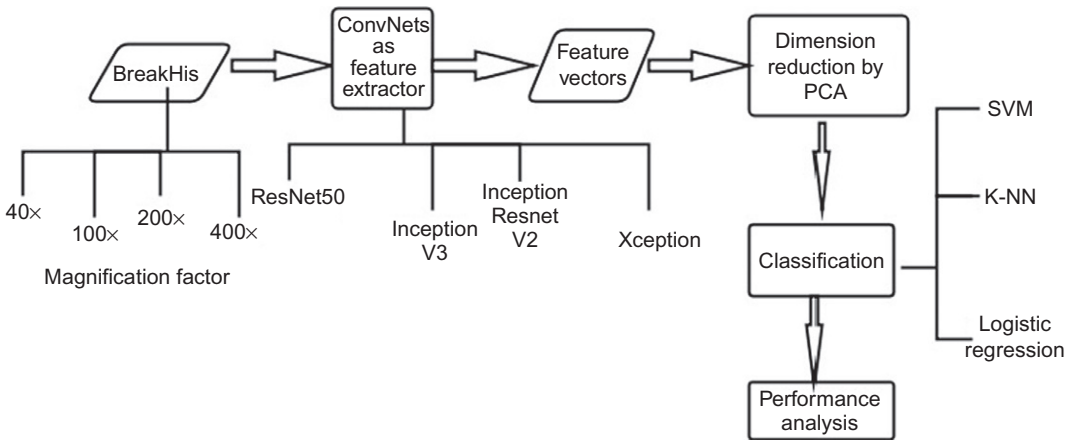

FIG. 4.6

Proposed model.

depending on the explained variance ratio, the dimension of the feature vector can be reduced. Then the reduced feature set is passed to the classifiers to perform binary classification to automate the classification of benign and malignant images. Classification is performed by three different classifiers are LR, SVM, K-NN. All the feature extraction, dimension reduction, and classification is done per magnification.

4.5 IMPLEMENTATION

System Description: Proposed model is implemented on a system with 8 GB RAM core i7 processor.

Tools: Python programming language with python image processing package *scikit-mage* [39], machine learning package *scikit-learn* [40], and for ConvNets, a deep learning framework, *Keras*, is used.

4.5.1 FEATURE EXTRACTION

I. Every image is resized by the required maximum input size for the four ConvNet models. After that, image pixels are rescaled to [−1, +1] (Table 4.2).
II. Then the resized and rescaled images are passed through the ConvNets Models and 1D feature vector is collected (Table 4.3).

All the ConvNets models are available in *keras* deep learning framework.

4.5.2 DIMENSIONALITY REDUCTION

PCA is applied on the feature vectors to reduce feature dimension. Before applying PCA, the features are standardized. PCA was trained only on a training set and projection to lower dimensional space was applied to both the training and test set. PCA is applied with an explained variance ratio of 0.95. PCA is available in the *scikit-learn* package (Table 4.4).

Table 4.2 Input Image Size Required for Four ConvNet Models

Feature Extractor	Input Size
ResNet50	224 × 224
InceptionV3	299 × 299
Inception-ResNet-v2	299 × 299
Xception	299 × 299

Table 4.3 Feature Dimension by Four Feature Extractors

Feature Extractor	Feature Dimension
ResNet50	2048
InceptionV3	2048
Inception ResnetV2	1536
Xception	2048

Table 4.4 Feature Dimension Reduced After Applying PCA

	40×	100×	200×	400×
ResNet50	650	659	658	622
InceptionV3	511	521	507	487
Inception ResnetV2	286	297	291	283
Xception	636	652	637	613

4.5.3 CLASSIFICATION

Classifiers: For classification purposes, three classifiers are used (LR, SVM, and K-NN). All these classifiers are available in the *scikit-learn* package. The reduced image feature vectors are passed to the classifiers for classification. Every classifier is first trained on the training set and tested on the test set.

4.5.4 TUNING HYPERPARAMETERS OF THE CLASSIFIERS

Some of the parameters of the classifiers, such as for LR, the parameters C, for SVM the parameters C, and *gamma* both with *rbf* kernel and for K–NN the parameters *n_neighbors* are tuned.

To tune the hyperparameters, the 10-fold cross validation approach is used. For each of the classifiers, 10-fold cross validation is performed on the training set with the combination of the above-mentioned hyperparameters and best cross validation giving hyperparameters are used as tuned parameters. The performance of each of the classifiers is described in Section 4.5.

4.6 RESULT AND ANALYSIS

To evaluate the performance of the three classifiers along with the four feature extractors, 10-fold cross validation approach applied on the training set and then a test result was generated on the test set.

4.6.1 10-FOLD CROSS VALIDATION RESULT

The below Table 4.5 describes the 10-fold cross validation accuracy on the training set for the different combinations of feature extractor and classifiers along with different magnification factors of the images.

4.6.2 MAGNIFICATION FACTOR WISE ANALYSIS ON VALIDATION ACCURACY

4.6.2.1 Validation accuracy of 40×

Interpretation: On the 40 × data, almost all of the combinations of feature extractors and classifiers gave a validation accuracy above 90% and among them the ResNet-50 and LR classifier gave the best cross validation score of 94.17% (Fig. 4.7).

Table 4.5 10-fold Cross Validation Results

Feature Extractors	Classifiers	Magnification Factor Wise 10-Fold Cross Validation Accuracy (%)			
		40×	100×	200×	400×
ResNet-50	LR	94.17	94.41	94.09	92.03
	SVM	90.72	90.32	90.06	89.28
	K-NN	90.85	88.45	91.06	89.35
Inception V3	LR	92.60	92.18	91.80	88.73
	SVM	92.54	90.04	90.55	89.00
	K-NN	91.97	89.06	89.93	87.42
Inception ResNet V2	LR	92.98	92.13	92.86	88.80
	SVM	94.04	92.30	94.22	89.42
	K-NN	89.90	88.52	89.62	85.30
Xception	LR	92.60	91.64	92.91	89.90
	SVM	90.40	88.88	90.80	87.15
	K-NN	89.71	88.88	88.26	85.44

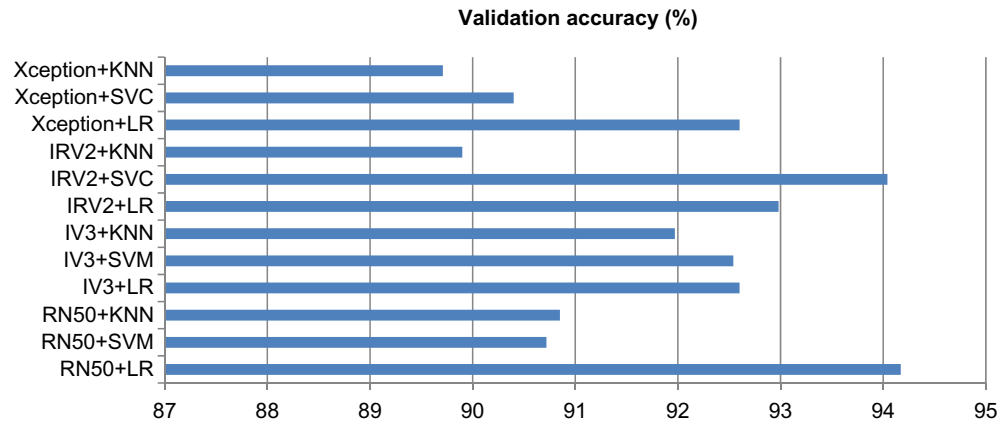

FIG. 4.7

Validation accuracy graph for 40×.

4.6.2.2 Validation accuracy of 100×

Interpretation: On the 100× data, all of the combinations of feature extractors and classifiers gave validation accuracy above 88% and the ResNet50 and LR classifier gave the best cross validation score of 94.41% (Fig. 4.8).

4.6.2.3 Validation accuracy of 200×

Interpretation: On the 200× data, all of the combinations of feature extractors and classifiers gave a validation accuracy above 88% and the Inception ResNet V2 with Support Vector Classifier gave the best cross validation score of 94.22% (Fig. 4.9).

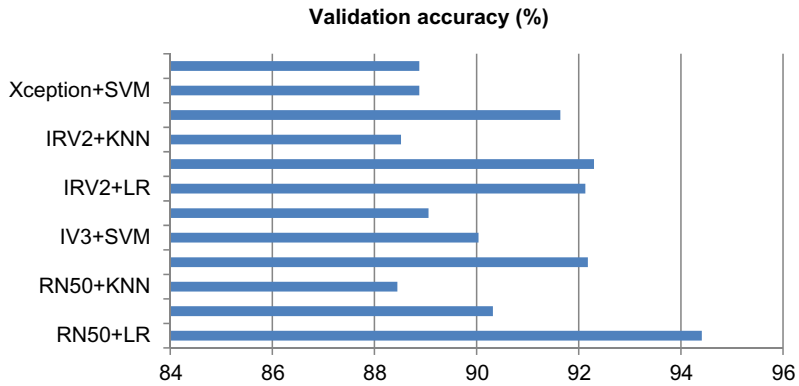

FIG. 4.8

Validation accuracy graph for 100×.

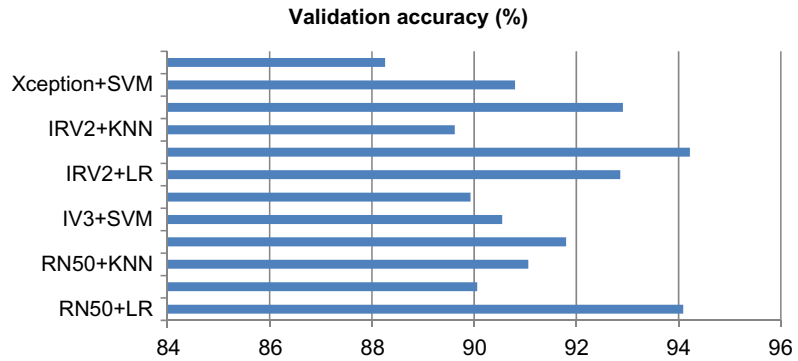

FIG. 4.9

Validation accuracy graph for 200×.

4.6.2.4 Validation accuracy of 400×

Interpretation: On the 400× data, most of the combinations of feature extractors and classifiers gave a validation accuracy above 86% and the ResNet50 and LR classifier gave the best cross validation score of 92.03% (Fig. 4.10).

4.6.2.5 Best validation accuracy

Table 4.6 summarizes the best validation accuracy achieved. It is noticeable that for 40×, 100×, and 400×, the ResNet-50 with LR classifier performed better than any others.

4.6.2.6 Performance on the test set

To evaluate the performance of the combinations of feature extractors and classifiers, some parameters are described below with the help of a sample confusion matrix. In this work, the positive class is *Malignant*, which means cancer is present and the negative class is *Benign*, which means cancer is not present.

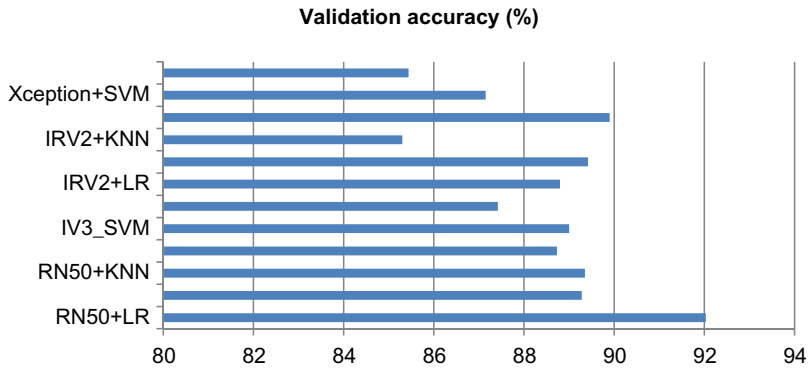

FIG. 4.10

Validation accuracy graph for 400×.

Table 4.6 Best Validation Accuracy

Magnification Factor	Feature Extractor	Classifier	Validation Accuracy (%)
40×	ResNet-50	LR	94.17
100×	ResNet-50	LR	94.41
200×	Inception ResNet V2	SVM	94.22
400×	ResNet-50	LR	92.03

Confusion Matrix

Class	Predicted Yes	Predicted No
Actual Yes	TP	FN
Actual No	FP	TN

Accuracy: Accuracy refers to how often the classifiers predict the correct label and is calculated as:

$$\text{Accuracy} = (\text{TP}+\text{TN})/(\text{TP}+\text{TN}+\text{FP}+\text{FN})$$

Precision: Precision refers to the correctness of predicting yes of a classifier and is calculated as:

$$\text{Precision} = \text{TP}/(\text{TP}+\text{TN})$$

Recall: Recall refers to the true positive rate and is calculated as:

$$\text{Recall} = \text{TP}/(\text{TP}+\text{FN})$$

F1-score: This is the weighted average of precision and recall and is calculated as:

$$\text{F1}-\text{score} = (2^{*}\,\text{precision}^{*}\,\text{recall})/(\text{precision}+\text{recall})$$

False Positive Rate (FPR): the ratio of FP and the summation of FP and TN.

False Negative Rate (FNR): the ratio of N and the summation of FN and TP.

4.6.3 RESULT AND ANALYSIS OF TEST PERFORMANCE

4.6.3.1 Test performance on 40×

Interpretation: With ResNet50, LR gave the highest precision while Support Vector claimed the highest F1 Score and recall though they both have the highest accuracy because the LR model can correctly classify the positive class better than the SVM model (Table 4.7, Fig. 4.11).

Interpretation: With InceptionV3, LR and SVM both have the highest recall value but the Support Vector classifier had the maximum accuracy and precision, leading to the maximum f1score (Table 4.8, Fig. 4.12).

Interpretation:

With Inception ResNet V2, SVM had the best accuracy, precision, recall, and f1score (Table 4.9, Fig. 4.13).

Table 4.7 Result of ResNet-50 With LR, SVM, and K-NN on 40×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ResNet-50	LR	96.24	96.25	98.60	97.41
	SVM	96.24	95.02	100	97.44
	K-NN	93.23	92.71	97.80	95.19

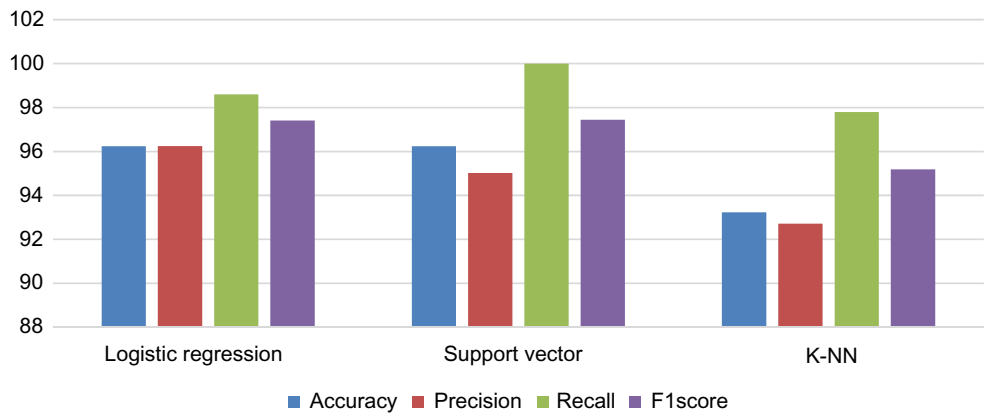

FIG. 4.11

Performance of Resnet50 with three difference classifiers on 40×.

Table 4.8 Result of Inception V3 with LR, SVM, and K-NN on 40×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Inception V3	LR	94.49	94.30	98.25	96.23
	SVM	94.74	94.61	98.25	96.40
	K-NN	91.23	94.16	93.14	93.65

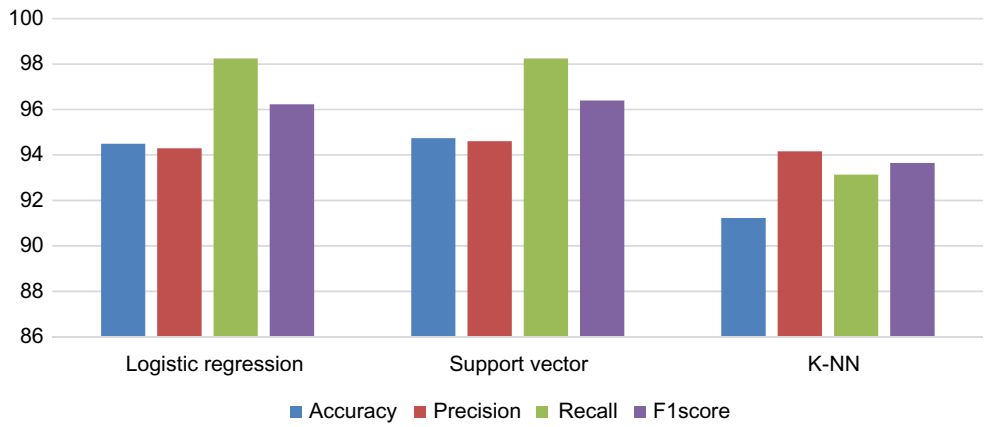

FIG. 4.12

Performance of InceptionV3 with three different classifiers on 40×.

Table 4.9 Result of Inception ResNet V2 with LR, SVM, and K-NN on 40×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1 Score (%)
Inception ResNet V2	LR	92.23	93.52	95.80	94.65
	SVM	95.99	96.55	97.90	97.22
	K-NN	89.47	92.19	92.19	92.19

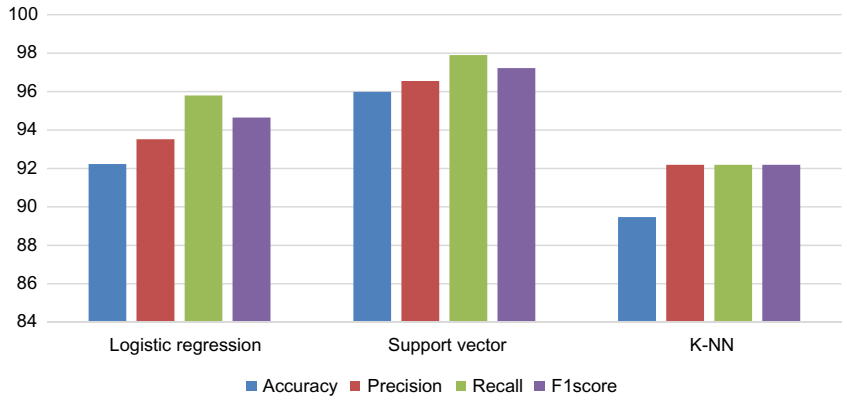

FIG. 4.13

Performance of Inception ResNet V2 with three different classifiers on 40×.

Table 4.10 Result of Xception with LR, SVM, and K-NN on 40×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Xception	LR	94.49	95.52	96.85	96.18
	SVM	93.73	93.65	97.90	95.73
	K-NN	91.48	92.08	94.94	93.49

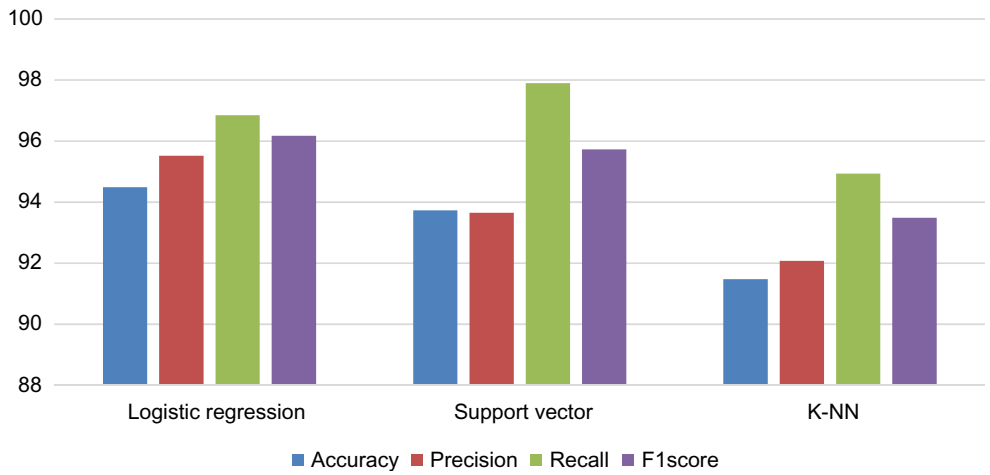

FIG. 4.14

Performance of Xception with three different classifiers on 40×.

Interpretation: With Xception, Support Vector had the maximum recall but LR gave the best accuracy, precision, and f1score (Table 4.10, Fig. 4.14).

4.6.3.2 Overall performance on 40×

Interpretation: The ResNet50 with both LR and Support Vector classifier had the maximum accuracy but the Support Vector classifier had best recall value. On the other hand, with Inception ResNet V2, Support Vector gave the highest precision (Fig. 4.15).

4.6.3.3 Test performance on 100×

Interpretation: With ResNet50, the Support Vector classifier gave the maximum recall value but K-NN had the maximum precision while the LR gave the best accuracy and f1score (Table 4.11, Fig. 4.16).

Interpretation: With InceptionV3, the Support Vector classifier had the maximum recall value, but the LR gave the best accuracy, precision, and f1score (Table 4.12, Fig. 4.17).

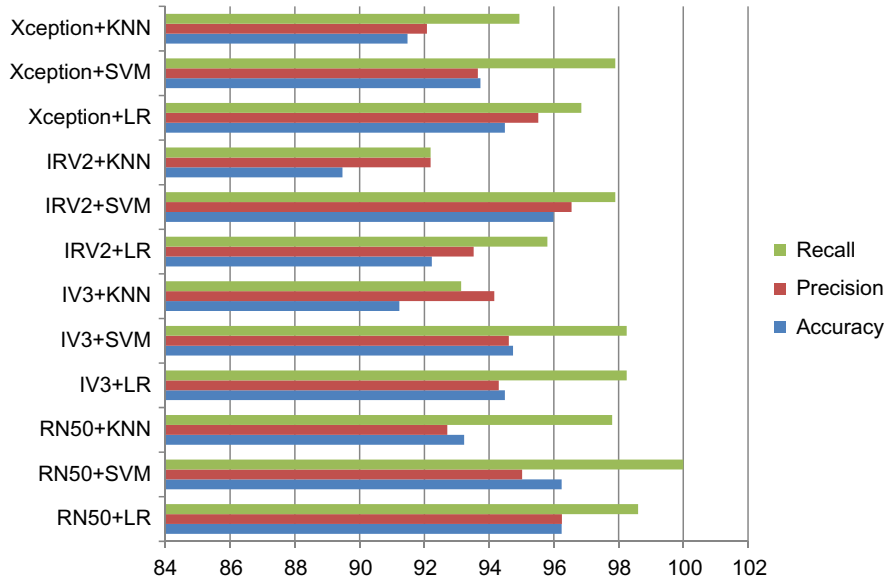

FIG. 4.15

Test performance graph for 40×.

Table 4.11 Result of ResNet-50 with LR, SVM, and K-NN on 100×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ResNet-50	LR	92.81	93.17	96.47	94.79
	SVM	90.41	89.32	97.53	93.24
	K-NN	91.37	94.24	92.91	93.57

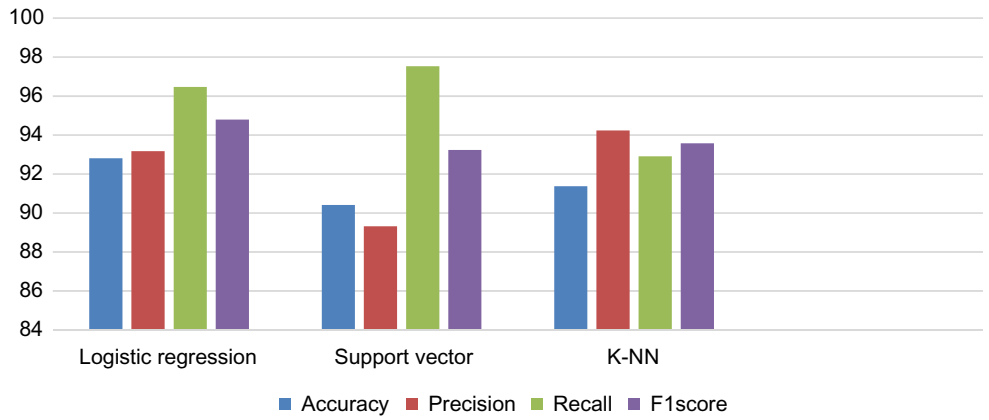

FIG. 4.16

Performance of ResNet50 with three different classifiers for 100×.

Table 4.12 Result of Inception V3 with LR, SVM, and K-NN on 100×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Inception V3	LR	90.89	92.10	94.70	93.38
	SVM	90.65	90.67	96.11	93.31
	K-NN	88.25	90.69	92.28	91.48

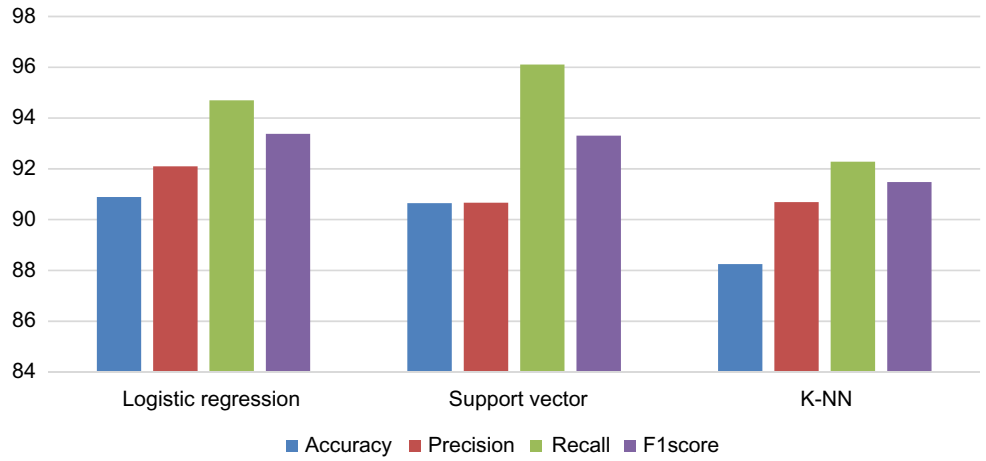

FIG. 4.17

Performance of InceptionV3 with three different classifiers for 100×.

Table 4.13 Result of Inception ResNet V2 with LR, SVM, and K-NN on 100×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Inception ResNet V2	LR	90.65	91.78	94.70	93.22
	SVM	91.37	91.58	96.11	93.79
	K-NN	91.85	94.31	93.64	93.97

Interpretation: With Inception ResNet V2, the Support Vector classifier gave the best recall value, but K-NN gave the highest accuracy, precision, and f1score (Table 4.13, Fig. 4.18).

Interpretation: With Xception, the Support Vector classifier gave the highest recall but LR had the highest accuracy, precision, and f1score (Table 4.14, Fig. 4.19).

4.6.3.4 Overall performance on 100×

Interpretation: The ResNet50 with LR gave the highest accuracy but Xception with the Support Vector classifier had a higher recall value than others. On the other hand, with Inception ResNet V2, K-NN had the highest precision (Fig. 4.20).

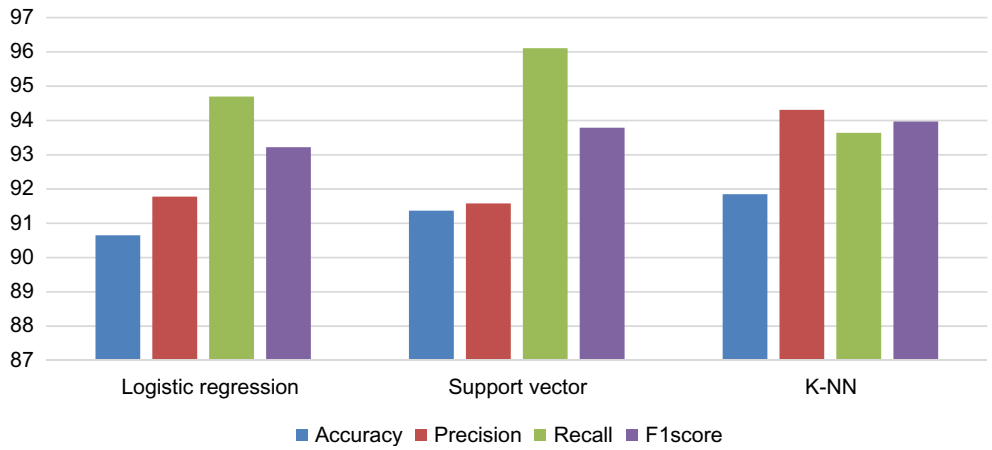

FIG. 4.18

Performance of Inception ResNet V2 with three different classifiers for 100×.

Table 4.14 Result of Xception with LR, SVM, and K-NN on 100×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Xception	LR	91.85	92.78	95.41	94.08
	SVM	91.37	90.23	97.88	93.90
	K-NN	89.21	89.80	95.12	92.39

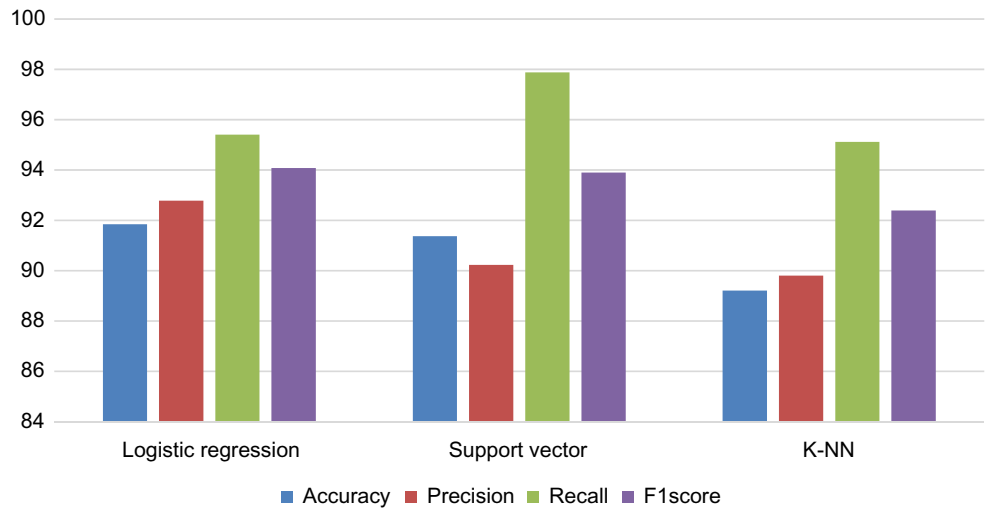

FIG. 4.19

Performance of Xception with three different classifiers for 100×.

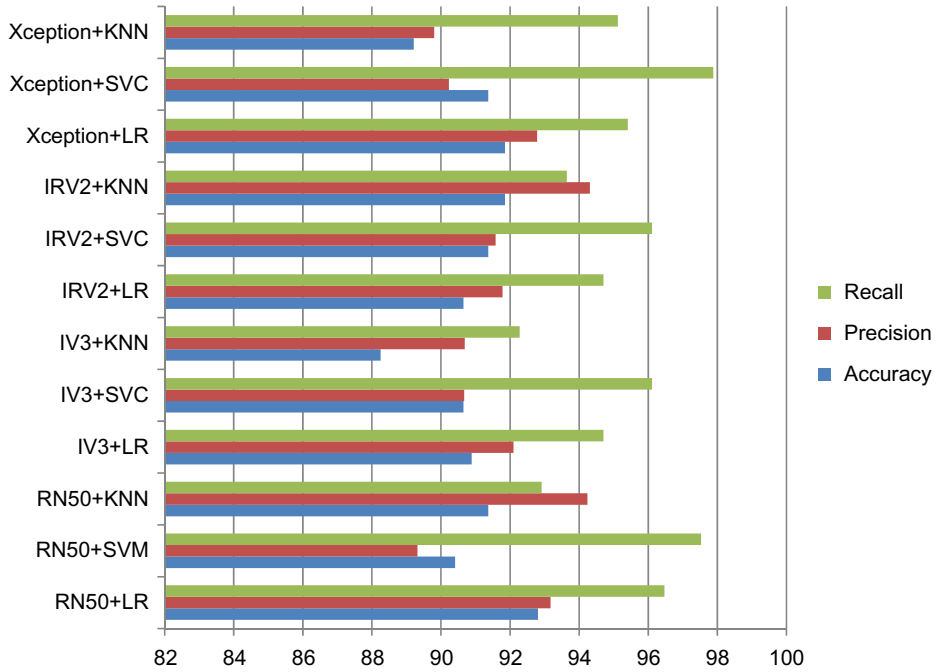

FIG. 4.20

Test performance graph for 100×.

Table 4.15 Result of ResNet-50 with LR, SVM, and K-NN on 200×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ResNet-50	LR	94.29	94.89	96.65	95.76
	SVM	91.81	90.41	98.14	94.12
	K-NN	92.06	91.01	97.31	94.05

4.6.3.5 Test performance on 200×

Interpretation: With ResNet50, SVM had the highest recall value but LR had the highest accuracy, precision, and f1score (Table 4.15, Fig. 4.21).

Interpretation: With InceptionV3, Support Vector had the highest accuracy, recall, and f1score but K-NN had the highest precision (Table 4.16, Fig. 4.22).

Interpretation: With Inception ResNet V2, the Support Vector classifier had the highest accuracy, precision, recall, and f1score (Table 4.17, Fig. 4.23).

Interpretation: With Xception, the Support Vector classifier had the best recall value, K-NN had the best precision, and LR had the highest accuracy and f1score (Table 4.18, Fig. 4.24).

Overall performance on 200×

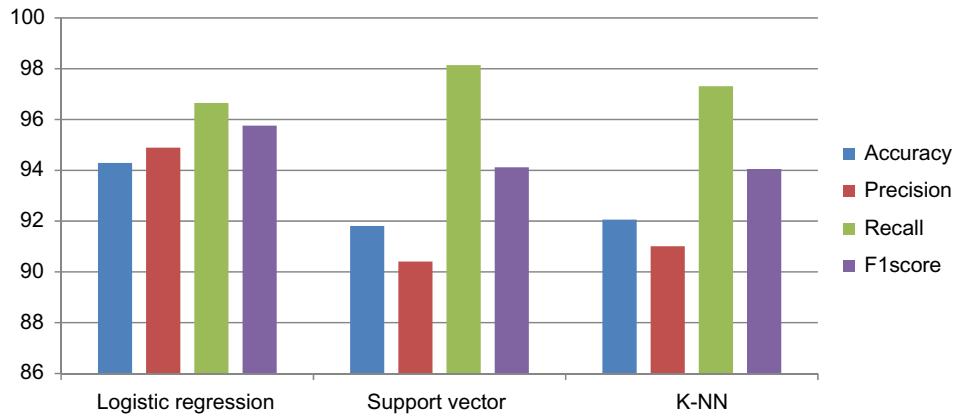

FIG. 4.21

Performance of ResNet50 with three different classifiers for 200×.

Table 4.16 Result of Inception V3 With LR, SVM, and K-NN on 200×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Inception V3	LR	89.83	90.71	94.42	92.53
	SVM	91.56	90.66	97.40	93.91
	K-NN	90.07	92.36	93.66	93.01

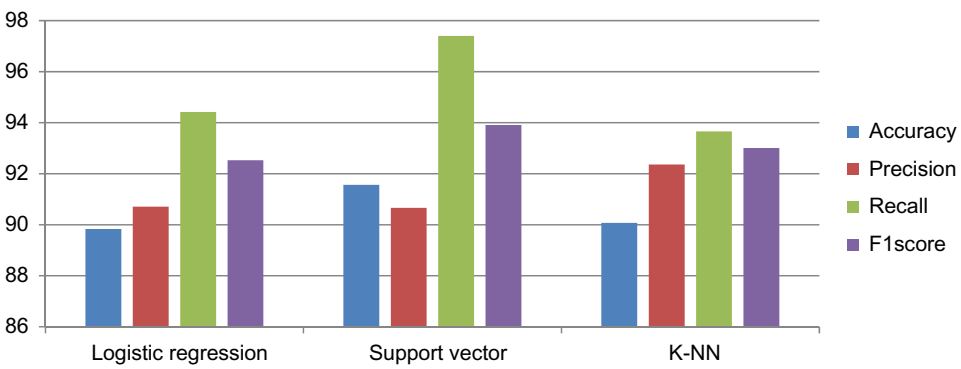

FIG. 4.22

Performance of InceptionV3 with three different classifiers for 200×.

Table 4.17 Result of Inception ResNet V2 With LR, SVM, and K-NN on 200×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Inception ResNet V2	LR	91.32	90.63	97.03	93.72
	SVM	94.29	94.24	97.40	95.80
	K-NN	89.83	92.25	93.24	92.74

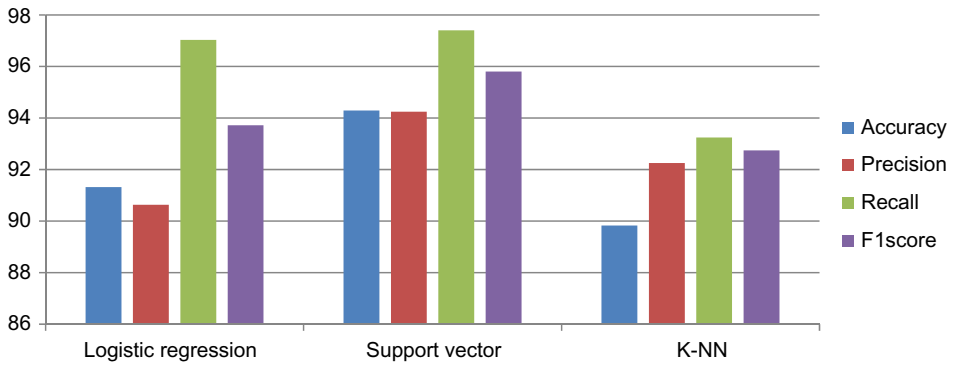

FIG. 4.23

Performance of Inception ResNet V2 with three different classifiers for 200×.

Table 4.18 Result of Xception with LR, SVM, and K-NN on 200×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Xception	LR	90.82	89.73	97.40	93.40
	SVM	90.32	88.33	98.51	93.15
	K-NN	88.34	90.25	92.59	91.41

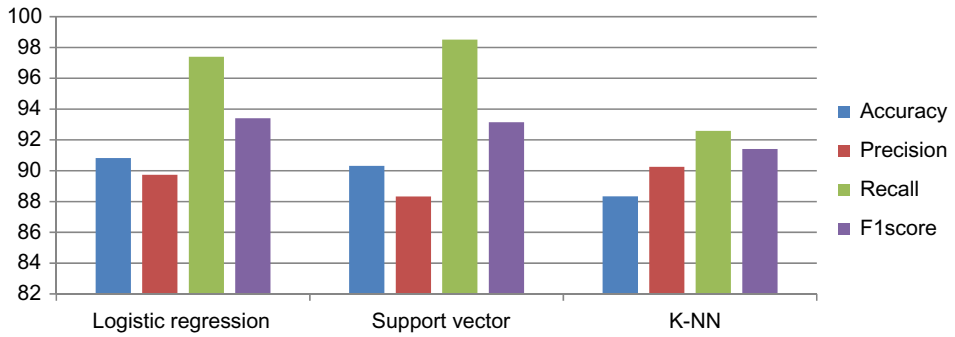

FIG. 4.24

Performance of Xception with three different classifiers for 200×.

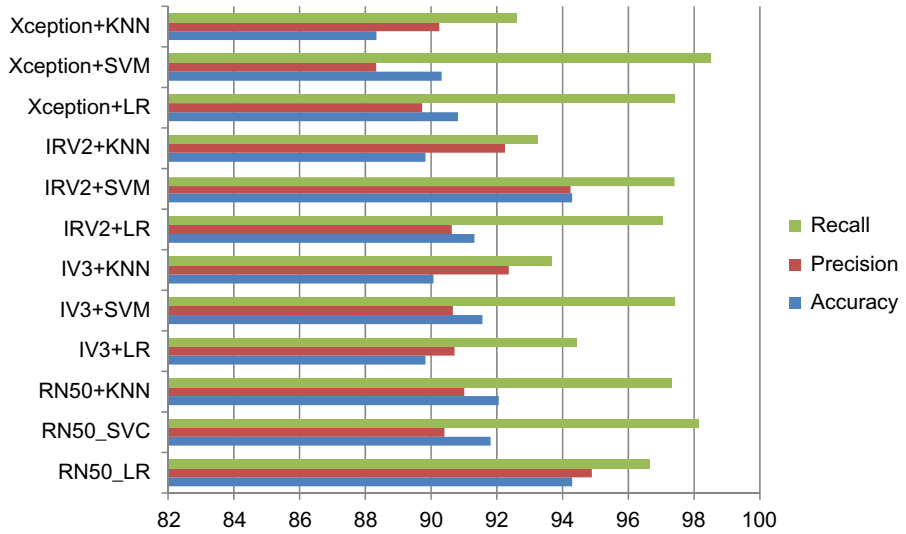

FIG. 4.25

Test performance graph for 200×.

Interpretation: The ResNet50 with LR and Inception ResNet V2 with Support Vector classifier had the highest accuracy but the Xception with Support Vector classifier had the highest recall value while ResNet50 with LR had the highest precision (Fig. 4.25).

4.6.3.6 Test performance on 400×

Interpretation: With ResNet50, the Support Vector classifier had the maximum value for accuracy, precision, recall, and f1score (Table 4.19, Fig. 4.26).

Interpretation: With InceptionV3, the Support Vector classifier had the maximum value for accuracy, precision, recall, and f1score (Table 4.20, Fig. 4.27).

Interpretation: With Inception ResNet V2, the Support Vector classifier had the highest accuracy, precision, recall, and f1score (Table 4.21, Fig. 4.28).

Interpretation: With Xception, the LR had the maximum accuracy, precision, recall, and f1score (Table 4.22, Fig. 4.29).

Table 4.19 Result of ResNet-50 with LR, SVM, and K-NN on 400×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
ResNet-50	LR	91.48	92.94	94.80	93.86
	SVM	92.86	93.75	96.00	94.86
	K-NN	89.65	91.63	93.12	92.37

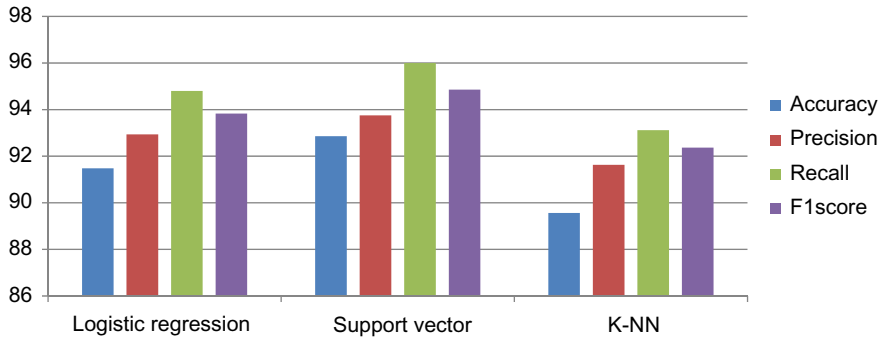

FIG. 4.26

Performance of ResNet50 with three different classifiers for 400×.

Table 4.20 Result of Inception V3 with LR, SVM, and K-NN on 400×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Inception V3	LR	91.21	91.60	96.00	93.75
	SVM	92.86	91.79	98.40	94.98
	K-NN	89.01	89.02	95.53	92.16

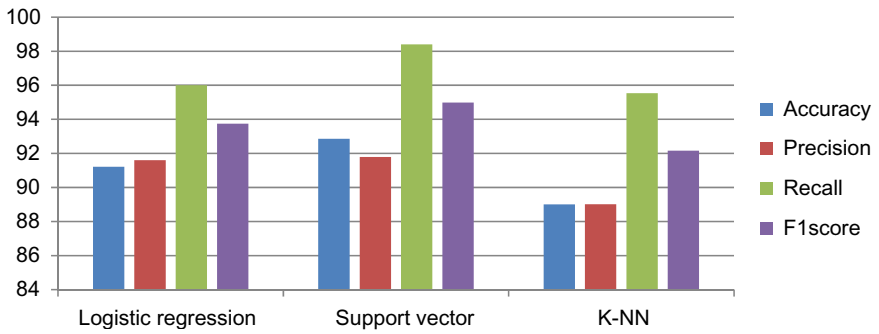

FIG. 4.27

Performance of InceptionV3 with three different classifiers for 400×.

4.6.3.7 Overall performance on 400×

Interpretation: Resnet50 with the Support Vector classifier and InceptionV3 with the Support Vector classifier had the highest accuracy but InceptionV3 with the Support Vector classifier gave the highest recall value and ResNet-50 with the Support Vector gave the highest precision (Fig. 4.30).

Table 4.21 Result of Inception ResNet V2 With LR, SVM, and K-NN on 400×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Inception ResNet V2	LR	90.93	92.55	94.40	93.47
	SVM	92.03	92.66	96.00	94.30
	K-NN	87.09	89.59	92.69	91.12

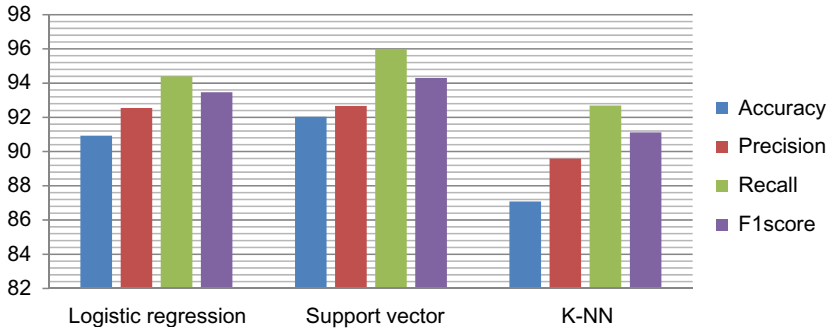

FIG. 4.28

Performance of Inception ResNet V2 with three different classifiers for 400×.

Table 4.22 Result of Xception with LR, SVM, and K-NN on 100×

Feature Extractor	Classifier	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)
Xception	LR	91.21	91.60	96.00	93.75
	SVM	89.84	90.80	94.80	92.76
	K-NN	84.34	84.23	93.19	88.48

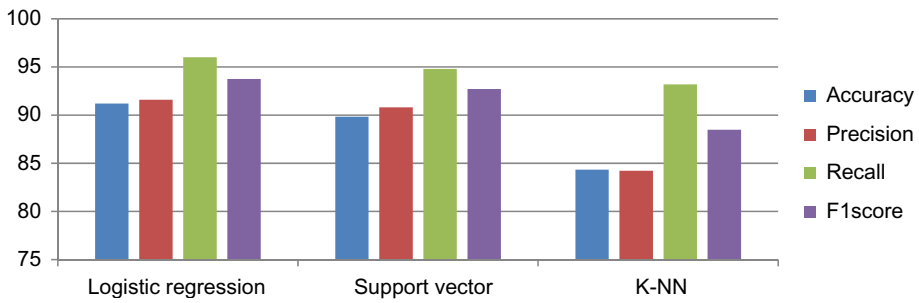

FIG. 4.29

Performance of Xception with three different classifiers for 400×.

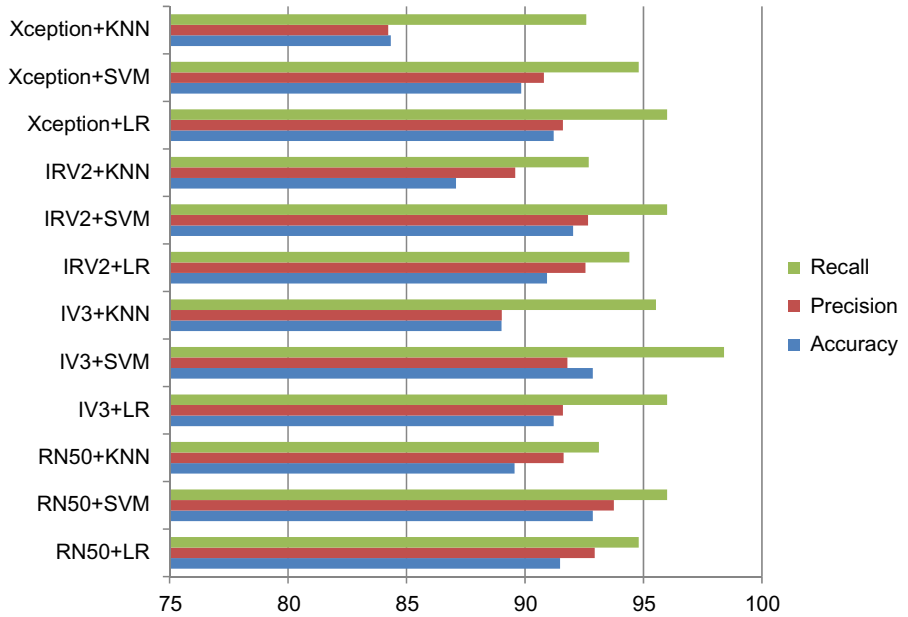

FIG. 4.30

Test performance graph for 400×.

4.7 DISCUSSION

In this work, several pretrained deep learning architectures were applied for feature extraction instead of using them as a classifier, which enabled us to save on the time for training. The dimensions of features were reduced so that the classifiers could fit them properly within a shorter period of time. The results reveal that the best validation and test accuracy for each of the magnification factors was quite impressive. Beside accuracy, we also analyzed the performance in terms of precision and recall since precision and recall are important for medical image classification as we always want to classify the tumorous image correctly rather than classifying the nontumorous image correctly and this can be considered as a tradeoff between precision and recall. The result for each of the combinations of feature extractor and classifier were shown and interpreted graphically. As it is not wise to make a decision in medical diagnosis based on a machine learning model, this model can assist the pathologists for diagnosis of breast tumor detection.

4.8 CONCLUSION

In this work, we have classified breast cancer histopathological images into two major classes—benign and malignant by our proposed model using some deep feature extractors and supervised classifiers. The field of machine learning is huge and there are lots of feature extractors and classifiers that can be used to automate this task. Since the overall performance of this model is not 100%, there is room for improvement.

REFERENCES

[1] Who.int, Breast Cancer, Available from: http://www.who.int/cancer/prevention/diagnosis-screening/breast-cancer/en/, 2015. (Accessed 12 February 2018).

[2] Mayoclinic.org, Breast Cancer–Diagnosis and Treatment–Mayo Clinic, Available from: https://www.mayoclinic.org/diseases-conditions/breast-cancer/diagnosis-treatment/drc-20352475, 2018. (Accessed 10 June 2018).

[3] F.A. Spanhol, L.S. Oliveira, C. Petitjean, L. Heutte, A dataset for breast cancer histopathological image classification. IEEE Trans. Biomed. Eng. 63 (7) (July 2016) 1455–1462, https://doi.org/10.1109/TBME.2015.2496264.

[4] F.A. Spanhol, L.S. Oliveira, C. Petitjean, L. Heutte, Breast cancer histopathological image classification using convolutional neural networks. in: 2016 International Joint Conference on Neural Networks (IJCNN), Vancouver, BC, 2016, pp. 2560–2567, https://doi.org/10.1109/IJCNN.2016.7727519.

[5] F.A. Spanhol, L.S. Oliveira, P.R. Cavalin, C. Petitjean, L. Heutte, Deep features for breast cancer histopathological image classification. in: 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, 2017, pp. 1868–1873, https://doi.org/10.1109/SMC.2017.8122889.

[6] H. Youh, G. Rumbe, Comparative study of classification techniques on breast cancer FNA biopsy data, Int. J. Int. Multimed. Artif. Intell. 1 (2010) 6–12.

[7] M. Meraliyev, M. Zhaparov, K. Artykbayev, Choosing best machine learning algorithm for breast cancer prediction, Int. J. Adv. Sci. Eng. Technol. 5 (3) (2017) 50–54.

[8] L. Shen, End-to-End Training for Whole Image Breast Cancer Diagnosis using An All Convolutional Design, eprint arXiv:1708.09427, 2017.

[9] A. Osareh, B. Shadgar, Machine learning techniques to diagnose breast cancer. in: 2010 5th International Symposium on Health Informatics and Bioinformatics, Antalya, 2010, pp. 114–120, https://doi.org/10.1109/HIBIT.2010.5478895.

[10] Agarap, Abien Fred, On Breast Cancer Detection: An Application of Machine Learning Algorithms on the Wisconsin Diagnostic Dataset, https://doi.org/10.1145/3184066.3184080, arXiv:1711.07831, 2017.

[11] J. Kriti Virmani, N. Dey, V. Kumar, PCA-PNN and PCA-SVM based CAD systems for breast density classification, in: Applications of Intelligent Optimization in Biology and Medicine, 2016. https://dblp.uni-trier.de/db/series/isrl/isrl96.html.

[12] L. Saba, N. Dey, A.S. Ashour, et al., Automated stratification of liver disease in ultrasound: an online accurate feature classification paradigm, Comput. Methods Prog. Biomed. 130 (2016) 118–134.

[13] N. Dey, A. Ashour, Classification and Clustering in Biomedical Signal Processing, first ed., IGI Global, Hershey, PA, USA, 2016.

[14] S. Cheriguene, N. Azizi, N. Zemmal, N. Dey, H. Djellali, N. Farah, Optimized tumor breast cancer classification using combining random subspace and static classifiers selection paradigms, in: A.E. Hassanien, C. Grosan, T.M. Fahmy (Eds.), Applications of Intelligent Optimization in Biology and Medicine, Intelligent Systems Reference Library, 96, Springer, Cham, 2016, pp. 289–307.

[15] N. Zemmal, N. Azizi, N. Dey, M. Sellami, Adaptive semi supervised support vector machine semi supervised learning with features cooperation for breast cancer classification, J. Med. Imaging Health Inf. 6 (1) (2016) 53–62.

[16] S. Kamal, N. Dey, S.F. Nimmy, S.H. Ripon, N.Y. Ali, A.S. Ashour, F. Shi, Evolutionary framework for coding area selection from cancer data, Neural Comput. & Applic. 29 (4) (2018) 1015–1037.

[17] A. Bhattacherjee, S. Roy, S. Paul, P. Roy, N. Kausar, N. Dey, Classification approach for breast cancer detection using back propagation neural network: A study, in: Biomedical Image Analysis and Mining Techniques for Improved Health Outcomes, IGI Global, 2016, pp. 210–221.

[18] H. Das, B. Naik, H.S. Behera, Classification of Diabetes Mellitus Disease (DMD): A Data Mining (DM) Approach, Progress in Computing, Analytics and Networking, Springer, Singapore, 2018, pp. 539–549.

[19] R. Sahani, C. Rout, J.C. Badajena, A.K. Jena, H. Das, Classification of intrusion detection using data mining techniques, in: Progress in Computing, Analytics and Networking, Springer, Singapore, 2018, pp. 753–764.
[20] H. Das, A.K. Jena, J. Nayak, B. Naik, H.S. Behera, A Novel PSO Based Back Propagation Learning-MLP (PSO-BP-MLP) for Classification, Computational Intelligence in Data Mining, Vol. 2, Springer, New Delhi, 2015, pp. 461–471.
[21] C. Pradhan, H. Das, B. Naik, N. Dey, Handbook of Research on Information Security in Biomedical Signal Processing. IGI Global, Hershey, PA, 2018, pp. 1–414, https://doi.org/10.4018/978-1-5225-5152-2.
[22] K.H.K. Reddy, H. Das, D.S. Roy, A Data Aware Scheme for Scheduling Big-Data Applications with SAVANNA Hadoop. Futures of Network, CRC Press, 2017.
[23] B.S.P. Mishra, H. Das, S. Dehuri, A.K. Jagadev, Cloud Computing for Optimization: Foundations, Applications, and Challenges, 39 Springer, 2018.
[24] P.K. Pattnaik, S.S. Rautaray, H. Das, J. Nayak (Eds.), Progress in Computing, Analytics and Networking: Proceedings of ICCAN 2017, Vol. 710, Springer, 2018.
[25] C.R. Panigrahi, M. Tiwary, B. Pati, H. Das, Big data and cyber foraging: Future scope and challenges, in: Techniques and Environments for Big Data Analysis, Springer, Cham, 2016, pp. 75–100.
[26] Cs231n.github.io, CS231n Convolutional Neural Networks for Visual Recognition, Available from: http://cs231n.github.io/convolutional-networks/, 2018. (Accessed 25 September 2018).
[27] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, Arxiv.org. Available from: https://arxiv.org/abs/1512.03385, 2018. (Accessed 25 September 2018).
[28] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, Z. Wojna, Rethinking the Inception Architecture for Computer Vision, Arxiv.org. Available from: https://arxiv.org/abs/1512.00567, 2018. (Accessed 25 September 2018).
[29] C. Szegedy, S. Ioffe, V. Vanhoucke, A. Alemi, Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, Arxiv.org. Available from: https://arxiv.org/abs/1602.07261, 2018. (Accessed 25 September 2018).
[30] F. Chollet, Xception: Deep Learning with Depthwise Separable Convolutions, Arxiv.org. Available from: https://arxiv.org/abs/1610.02357, 2018. (Accessed 25 September 2018).
[31] Keras.io, Keras Documentation, Available from: https://keras.io/, 2018. (Accessed 10 June 2018).
[32] Cs231n.github.io, CS231n Convolutional Neural Networks for Visual Recognition, Available from: http://cs231n.github.io/transfer-learning/, 2018. (Accessed 10 June 2018).
[33] GeeksforGeeks, Introduction to Dimensionality Reduction-GeeksforGeeks, Available from: https://www.geeksforgeeks.org/dimensionality-reduction/, 2018. (Accessed 11 June 2018).
[34] Plot.ly, Principal Component Analysis, Available from: https://plot.ly/ipython-notebooks/principal-component-analysis/, 2018. (Accessed 11 June 2018).
[35] J. Brownlee, Supervised and Unsupervised Machine Learning Algorithms. Machine Learning Mastery, Available from: https://machinelearningmastery.com/supervised-and-unsupervised-machine-learning-algorithms/, 2018. (Accessed 10 June 2018).
[36] J. Brownlee, Logistic Regression for Machine Learning. Machine Learning Mastery, Available from: https://machinelearningmastery.com/logistic-regression-for-machine-learning/, 2018. (Accessed 10 June 2018).
[37] M. Learning, U. Code, Understanding Support Vector Machine Algorithm From Examples (Along With Code). Analytics Vidhya, Available from: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/, 2018. (Accessed 10 June 2018).
[38] J. Brownlee, K-Nearest Neighbors for Machine Learning. Machine Learning Mastery, Available from: https://machinelearningmastery.com/k-nearest-neighbors-for-machine-learning/, 2018. (Accessed 10 June 2018).
[39] Scikit-image.org, Scikit-Image: Image Processing in Python—Scikit-Image, Available from: http://scikit-image.org/, 2018. (Accessed 10 June 2018).
[40] Scikit-learn.org, scikit-Learn: Machine Learning in Python—Scikit-Learn 0.19.1 Documentation, Available from: http://scikit-learn.org/stable/, 2018. (Accessed 10 June 2018).