Breast Cancer Diagnosis in Histopathological Images Using ResNet-50 Convolutional Neural Network

Qasem Abu Al-Haija, Adeola Adebanjo
Department of Electrical and Computer Engineering, Tennessee State University,
3500 John A. Merritt Blvd, Nashville, TN 37209
Qabualha@Tnstate.edu, Aadebanj@Tnstate.edu,

***Abstract*— Breast cancer disease is the second most common world cause of cancer death in women. However, the early diagnostics and detection can provide a significant chance for correct treatment and survival. In this work, we propose an accurate and inclusive computational breast cancer diagnosis framework using ResNet-50 convolutional neural network to classify histopathological microscopy images. The proposed model employs transfer learning technique of the powerful ResNet-50 CNN pretrained on ImageNet to train and classify BreakHis dataset into benign or malignant. The simulation results showed that our proposed model achieves exceptional classification accuracy of 99% outperforming other compared models trained on the same dataset.**

***Keywords* — Breast Cancer, Histopathological Images, Medical image processing, ResNet-50, Convolutional Neural Network (CNN), Deep learning, Transfer Learning.**

I. Introduction

Cancer disease causes cells to divide uncontrollably in which abnormal cells grow and invade healthy cells in the body [1]. This can result in tumors, damage to the immune system, and other impairment that can be fatal. Indeed, the majority of cancer cases in the U.S. is breast cancer as reported by to the [2]. Breast Cancer Disease (BCD) is basically described as excessive or uncontrolled growth of breast tissues occurs. According to World Health Organization (WHO) [3], BCD is the second most common cause of death from cancer in women. However, BCD can be treated if it is detected at the earlier stages as reported by National Breast Cancer Foundation (NBCF) [4]. Therefore, BCD detection system is significantly on-demand to be addressed.

To cope with such medical image detection/classification task, deep learning (DL) has evolved as a subset of artificial intelligence that does its inferencing using deep neural networks by employing the artificial neural networks with several layers among the input layer and output layer. Indeed, image classification task using deep learning techniques has generated a lot of interest in recent years. ImageNet Large Scale Visual Recognition Challenge (ILSVRC) has been a genuine platform for researchers to showcase new ideas for classification. For example, in 2010, the use of GPUs, and a highly optimized implementation of 2D convolution was demonstrated as powerful enough to facilitate the training of large Convolutional Neural Networks (CNNs) [5].

CNNs have at least one Convolution layer, wherein instead of matrix multiplication, a convolution operation is performed on the input matrix in order to learn distinct low-level and high-level features of the image [6]. Deep CNNs are able to learn more features by increasing the depth of the network. However, increasing the depth of the network results in problems of vanishing gradients and degradation [7]. Thereafter, the continuous development in deep neural networks has enrich the AI field with the residual learning framework was presented in 2015 to ease the training of deep CNN networks [8]. This framework resulted in easier optimization of the network, and a higher accuracy. The network, later known as ResNet, was the basis of submissions to ILSVRC competition, where it won the first place on the task of ImageNet detection and ImageNet localization [9].

Residual neural networks (ResNet) address these challenges by introducing a “Residual block”, which features a “skip connection”, that adds the output from the previous layer to the layer ahead as illustrated in Fig.1. If x and $F(x)$ below do not have the same dimension, x is multiplied by a linear projection W to equalise the dimensions of the short-cut connection and the output layer. Empirical results from [9] demonstrate that the network is able to maintain stability even with far more layers than typical Convolutional Neural Networks.

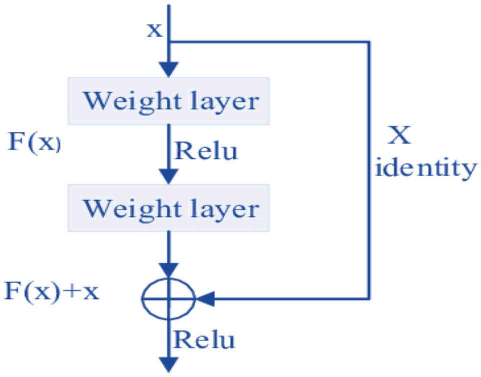

Fig. 1. Residual Network Building Block

In this paper, we propose to use a ResNet-50 (residual CNN with 50 layers deep) to produce classifications of histopathologic images to help providing an early detecting of BCD. The BCD dataset stores different kinds of breast cancer tissues and classifies them as either benign or malignant. While

978-1-7281-9615-2/20/$31.00 ©2020 IEEE

the benign tissues are non-cancerous tumors, malignant tumors are cancerous, and can invade nearby tissue or spread to other parts of the body. Based on the collected images, a ResNet-50 network is trained using transfer learning. We show that the testing accuracy of our model is superior. In particular, the core contributions of the proposed work can be listed as follows:

- We provide a comprehensive efficient classification model that can classify the breast cancer imaging of BreakHis dataset into benign and malignant. Besides, the we present detailed preprocessing operations for the collected medical images prior to the use with deep learning algorithms.
- We employ the transfer learning technique for ResNet50 CNN that is pre-trained with ImageNet dataset to learn the new features for BreakHis dataset leveraging the power of free access GPU runtime provided by Google Co-laboratory.
- Extensive experimental findings are given to provide more insight into the proposed architecture and methodology. This includes simulation results related to the classification error and accuracy for training and testing as well as benchmarking of our results with existing related work.

The remainder of this paper is structured as follows: the next section, section II describes and discusses the system design modeling and architecture. Section III provides details about experimental environment, evaluation, and discussion. Finally, Section IV concludes the paper.

II. SYSTEM MODELING ARCHITECTURE

In this section, we describe our proposed system model which comprises four modules including: data collection module, data preprocessing module, feature learning module and data classification module. The complete system architecture showing all components is illustrated in Fig. 2. All subsystems (modules) are explained in the upcoming subsections.

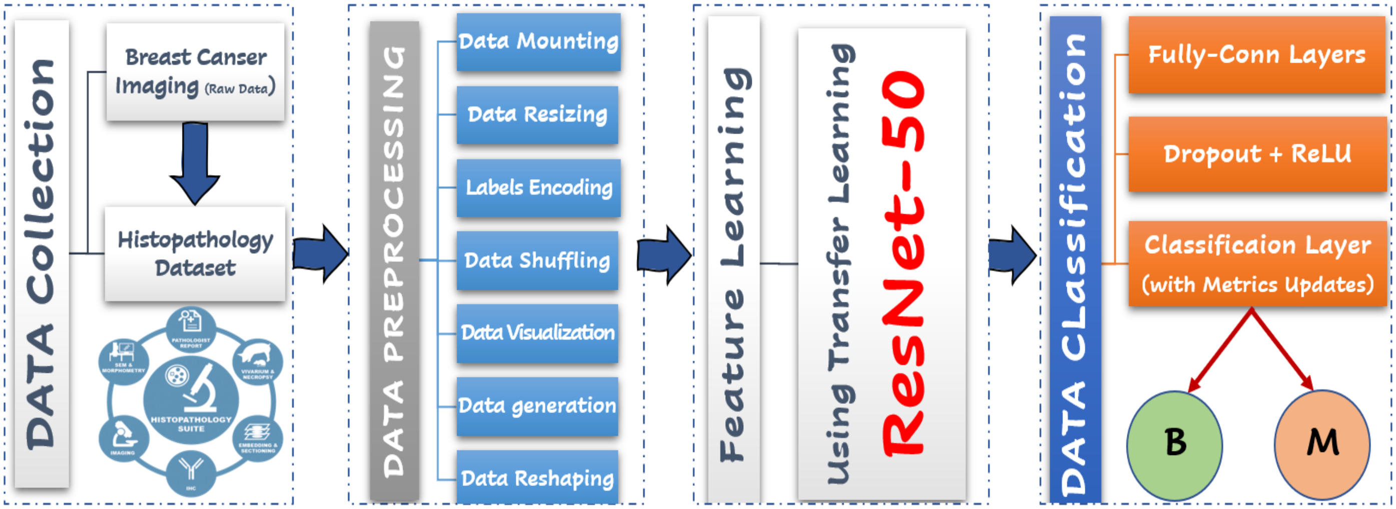

Fig. 2. Proposed System Architecture (Comprehensive data flow and system modules)

2.1 Data Collection Module

BCD data are available in the form of histopathological stained images [10] datasets such as WDBC breast cancer dataset [11], MITOS & ATYPIA-14 breast cancer dataset [12] and BreakHis-16 breast cancer dataset [13]. In this research, we are employing the BreakHis histopathology images breast cancer dataset to implement a deep detection system that can be used to provide binary classification of the BC image-data into *benign* (B) or *malignant* (M). Illustration samples for the benign and malignant images are provided in Fig. 3. BreakHis composed of RGB images (700 x 460 resolution) are taken by an accurate system composed of high-resolution camera (Samsung SCC-131AN) coupled with a 3.3x microscopic unit (Olympus BX-50). Images are captured in four different magnification levels that are equivalently distributed (~25% for each level) as provided in Table 2.

TABLE I. IMGAES' DISTRIBUTION IN THE BREAKHIS DATASET

Category	Magnification level				Total
	40X	100X	200X	400X	
Benign	652	644	623	588	**2480**
Malignant	1370	1437	1390	1232	**5429**
Total	**1995**	**2081**	**2013**	**1820**	**7909**

Besides, there are many reasons to for this selection including:

- A comprehensive dataset with 7909 biopsy histopathology images breast cancer including both benign and malignant images acquired on 82 patients, publicly available.
- A Recent dataset associated with automated classification tasks, published by IEEE Transactions on Biomedical Engineering, 2016.
- A common histopathology dataset, in which its automated classification system is would be very valuable computer-aided diagnosis tool for clinician, if developed with high accuracy classification by employing CNN techniques.

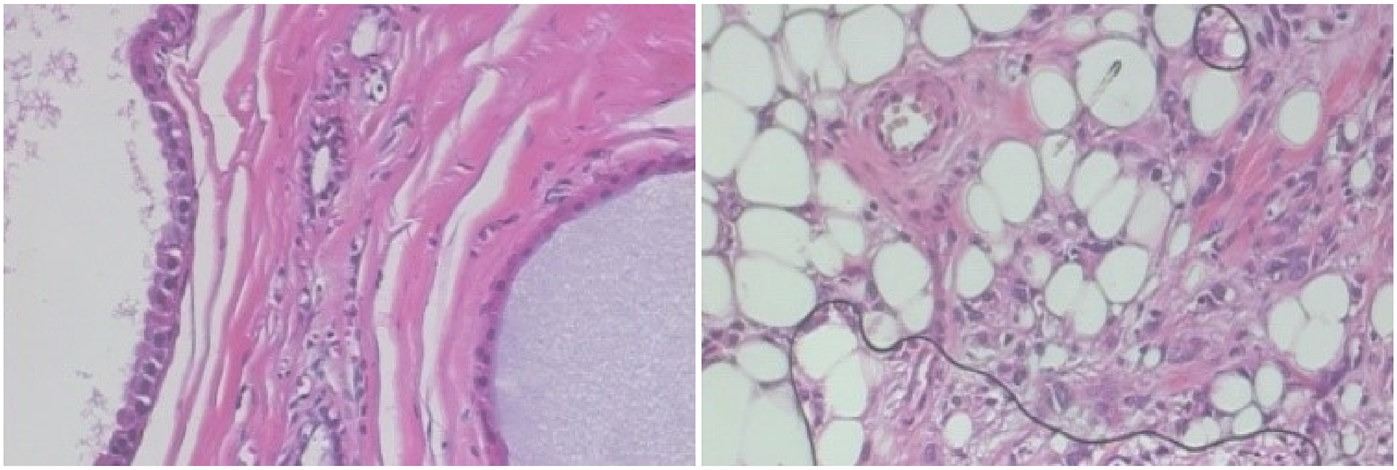

Fig. 3. Sample Images (a) Benign (b) Malignant.

2.2 Data Preprocessing Module

Generally, data preprocessing term belongs to all the conversions performed on the raw data prior to be processed by the deep learning module. For instance, training a convolutional neural network on raw images will probably lead to bad classification performances [14]. In this work, the collected dataset has passed through seven preprocessing operations (as shown in the second stage of Fig. 2) before it is fed to the next ResNet module, these operations are:

Data Mounting: This stage is used to mount Google Drive Account (DGA) as a virtual drive, just like a USB Drive on windows OS so you can browse and access your Drive from Google Co-laboratory. Therefore, we have uploaded our BreakHis dataset into a folder (named as BCDataset) on Google drive and then accessed the data into Co-Lab using $Python/glop$ library that enable the reading of dataset from external folders and $Python/pandas$ library that provides data manipulation including data framing, data reading and writing between in-memory data structures and different formats [15].

Data Resizing: This stage is necessary to remove redundancy from the input data which only contributes to the computational complexity of the network without providing any significant improvements in the result. This is achieved with the help of $Python/keras.preprocessing$ library. As a result of traying several image dimensions, we end up with image dimension of $144\,x\,96$ which minimizes the size of image dimensions by a factor of 5 while maintaining image readability with efficient computational complexity.

Data Encoding: This stage is used to convert categorical data (textual data) into numerical values in which our deep learning predictive models can understand. In this step, we have employed the label encoding technique to converting each value in a the category column into a number, that is it, the category 'Benign' has given the value '0' while the category 'Malignant' has given the value '1'. This is achieved with the help of $Python/sklearn.preprocessing$ library.

Data Shuffling: This stage is used to redistribute the data samples of the training dataset to ensure that each data sample creates an "independent" change on the model, without being biased by the same points before them. This is achieved with the help of $Python/keras.preprocessing$ library.

Data Visualization: This stage is necessary to sample and examine the input data to ensure the readability of the input images by plotting few random samples of the training dataset via 2D representation with the new image dimensions. This is achieved with the help of both $Python/tensorflow$ and $Python/numpy$ libraries.

Data Generation: This stage is used to generate batches of tensor image data with real-time data augmentation. The data will be looped over (in batches) for both training and testing. Also, batch normalization is performed at this stage with image plotting for sample normalized images along with encoded labels. This is achieved with the help of both $Python/tensorflow$, $Python/keras.preprocessing$ and $Python/matplotlib$ libraries.

Data Reshaping: This stage is used to customize the input layer of ResNet-50 to accommodate the input shape for our preprocessed dataset ($Img_{Width} = 144$, $Img_{Height} = 96, No_{Channels} = 3$). This is achieved with the help of $Python/keras.preprocessing$ library.

2.3 Feature Learning Module

In machine learning, feature learning (FL) is a set of techniques that allows a system to automatically discover the representations needed for feature detection, prediction, or classification from the preprocessed dataset [16]. This allows a machine to learn the features and use them to perform a specific task such as classification or prediction. In deep learning, the feature learning can be accomplished by developing a complete convolutional neural network (CNN) to train and test the set of images or by customizing a pretrained CNN in the classification/ prediction for the new images-set. The later technique is called Transfer learning. The idea of both techniques is illustrated in Fig.4. According to the figure, with transfer learning, you use the convolutional base (green module in the figure) and only re-train the classifier to your dataset (pink module).

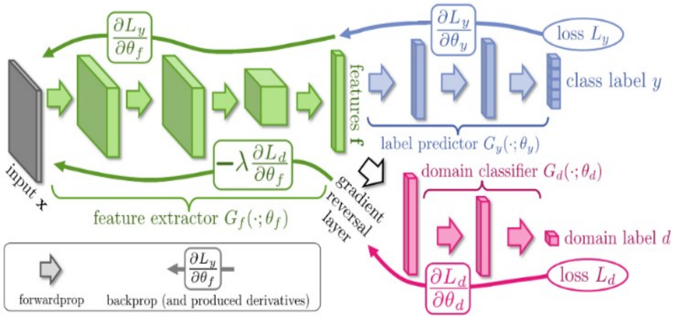
$\frac{\partial L_y}{\partial \theta_f}$
$\frac{\partial L_y}{\partial \theta_y}$
loss L_y
class label y
features f
label predictor $G_y(\cdot;\theta_y)$
$-\lambda\frac{\partial L_d}{\partial \theta_f}$
domain classifier $G_d(\cdot;\theta_d)$
input x
gradient reversal layer
feature extractor $G_f(\cdot;\theta_f)$
domain label d
loss L_d
$\frac{\partial L_d}{\partial \theta_d}$
forwardprop
$\frac{\partial L_y}{\partial \theta_f}$
backprop (and produced derivatives)

Fig. 4. Illustration of no-transfer learning vs transfer learning CNN [17].

$Transfer\,learning\,(TL)$ is usually employed in the applications of DL which enable you to utilize a pretrained network to perform new prediction/classification tasks. This, indeed, require fine-tuning the learning parameters of the utilized pretrained network with randomly initialized weights to accommodate the new learning tasks. TL usually provides much faster and easier learning/training than training the network from scratch. As reported in [18], transfer learning is an optimization, a shortcut to saving time or getting better performance. This is illustrated in Fig. 5 that analyze the training performance of CNN employing transfer learning vs CNN with no transfer learning.

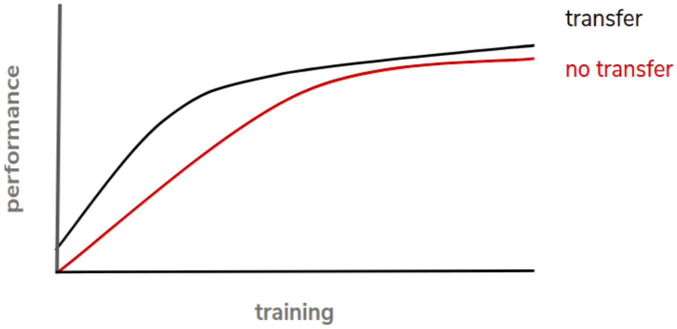

Fig. 5. Training performance of no-transfer learning CNN vs transfer learning CNN **[18]**.

In this module, we are utilizing the transfer learning technique to retrain the powerful ResNet-50 CNN with Fine-tuning for the network parameters and hyperparameters. This is accomplished by creating a model of ResNet-50 with pretrained parameters (weights) from $imagenet\ dataset$ [19] after the preprocessing of the collected dataset (histopathologic images). This is achieved with the help of both $Python/keras.applications.resnet$50 and $Python/keras.models$. Illustration of this module can be depicted from Fig.6.

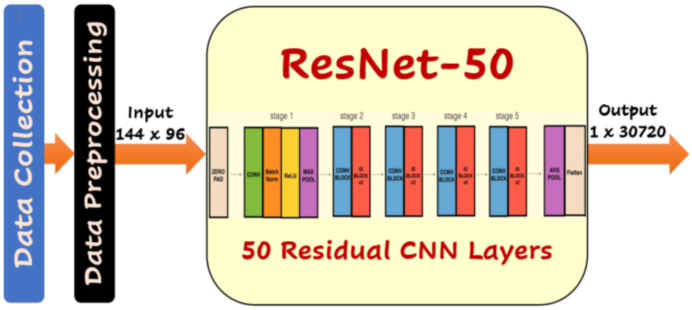

Fig. 6. Feature Learning Module Using ResNet-50 Transfer Learning

2.4 Data Classification Module

Data classification is an essential feature to separate large datasets into classes for the purpose of decision making, pattern recognition and others [20]. A classification layer makes use of fully connected layer and computes the cross-entropy loss for multi-class classification problems with mutually exclusive classes. This is achieved using $Paython/keras.layers$, $Paython/keras.models$ and $Paython/keras/optimizers$. Illustration of this module can be depicted from Fig.7.

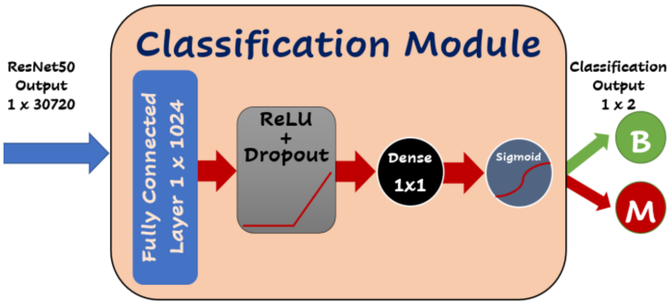
Classification Module
ResNet50 Output 1 x 30720
Fully Connected Layer 1 x 1024
ReLU + Dropout
Dense 1x1
Sigmoid
Classification Output 1 x 2
B
M

Fig. 7. Data Classification Module

According to the figure, the module receives the features from ResNet-50 and pass through a fully connected (FC) layer composed of 1024 neurons configured a 40% of dropout to prevent over-fitting [21]. After that, the units have been activated with rectification function namely known as $ReLU$. $ReLU$ function is $MAX\,(X,0)$ that sets all negative values in the matrix X to zero while all other values are kept constant. The reason of using $ReLU$ is that training a deep network with $ReLU$ tended to converge much more quickly and reliably than training a deep network with sigmoid activation [22]. Finally, to provide the probabilities for the classes, the output layer was composed of one neuron unit configured with Sigmoid function (Binary classifier). $Sigmoid$ is mathematical function that takes as input a vector of K real numbers and normalizes it into a probability distribution consisting of two probabilities (e.g. Benign vs Malignant) [23]. S$igmoid$ function is defined as follows:

$$S(x)_i = \frac{1}{1+e^{-x}} = \frac{e^x}{e^x+1}\,, \quad S\colon \mathbb{R}^k \longmapsto \mathbb{R}^k$$

$$for\ i\ =\ 1,2\ldots,K\ and\ x\ =\ (x_1,x_1,\ldots,x_K) \in \mathbb{R}^k$$

Besides, to calculate the loss for training and testing we have employed *Mean Squared Error (MSE) loss*. MSE Loss [24] is calculated as the average of the squared differences between the predicted and actual values. The result is always positive regardless of the sign of the predicted and actual values and a perfect value is 0.05 MSE function is defined as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2$$

Where: n is the number of samples, Y_i is the original data sample, and $\hat{Y}_i$is the predicted data sample

Finally, to finalize the compilation of *Keras model*, we have utilized Root Mean Square Propagation optimizer (RMSprop) [25]. *RMSprop* utilizes the magnitude of recent gradients to normalize the gradients. In *RMSprop*, we divide the learning rate for a weight by a running average of the magnitudes of recent gradients for that weight. That is, it, keep a moving average of the squared gradient for each weight. The update is done separately for each parameter as follows:

$$For\ each\ Parameter\ w^j$$

$$(j\ subscript\ dropped\ for\ clarity)$$

$$\nu_t = \rho\nu_{t-1} + (1-\rho) * g_t^2$$
$$\Delta\omega_t = -\frac{\eta}{\sqrt{\nu_t + \epsilon}} * g_t$$
$$\omega_{t+1} = \omega_t + \Delta\omega_t$$

η : *Initial Learning rate*
ν_t : *Exponential Average of squares of gradients*
g_t : *Gradient at time t along* ω^j

III. Experimental Environment and Evaluation

To accomplish this proposed classification task, we have utilized different development tools and packages including *Python* 3.7 language along with *TensorFlow* 2.0 package and other aforementioned libraries, *Google Colaboratory* development environment leveraging the power of free access *GPU runtime* and *Google Drive* for storing and accessing the dataset. Besides, the experimental setup for training/testing model has been configured as follows: 75% of the dataset used for training (i.e., ~6000 images, here we used all images with 100X, 200X, 400X magnifications from both classes for training), 25% of the dataset used for testing (i.e., ~2000 images, here we used all images with 40X magnification from both classes for testing), the number of epochs = 200, the number of steps per epoch (iterations)=200, the number of verbose=1 (i.e., one progress bar line per epoch), and the *batch_size* = 32. This configurations have been achieved using *Python/model.fit* & *Python/matplotlib* libraries.

Also, the plot for loss function comparing the behavior of training loss and testing loss obtained during the training process is presented in Fig. 8. It can be clearly seen, both losses are systematically decreasing while training proceeds with faster threshold obtained for the training loss after only 25 epochs. However, the testing loss has saturated after almost 125 epochs of training process with less than 0.05 of MSE. This difference in saturation levels and threshold of training loss and testing loss is permitted to avoid underfitting or overfitting.

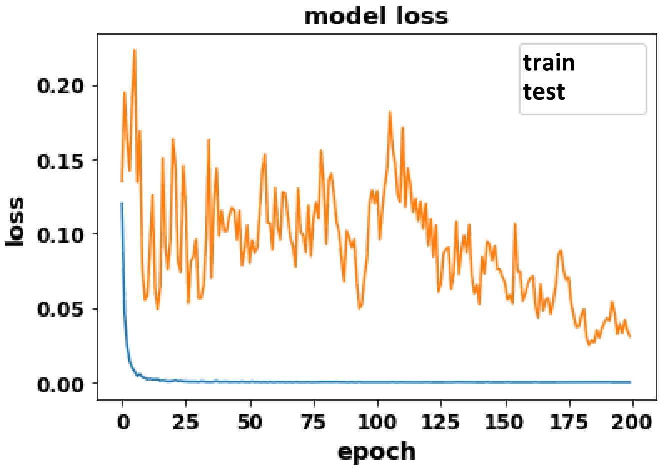

Fig. 8. Training/Testing Losses vs. number of epochs

Moreover, the plot for accuracy metric comparing the performance of training accuracy and testing accuracy obtained during the training process is given in Fig. 9. According to the figure, both accuracy curves are steadily increasing while training proceeds with faster ceiling level obtained for training accuracy after which recorded almost 100% only after 25 epochs. While the testing accuracy level was fluctuating between 98% and 99.8% after 100 epochs recoding an average testing accuracy of 99% of overall testing accuracy.

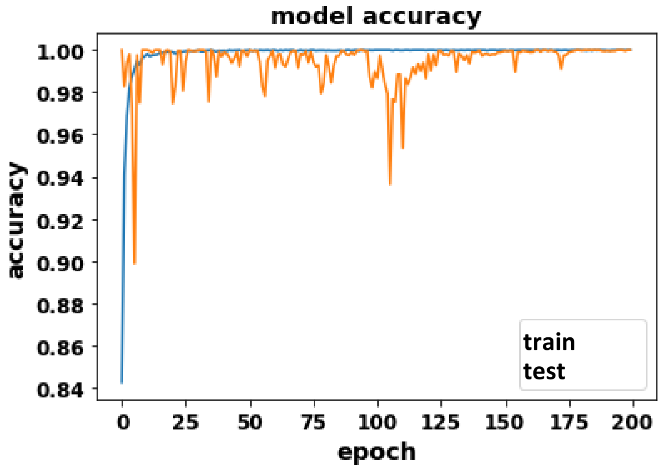

Fig. 9. Training/Testing Accuracies vs. number of epochs

Finally, even though the exiting state-of-art researches for classifying the histopathological images dataset use different network configurations, learning policies, programming techniques, and computing platforms, however, we still can compare the classification system performance in terms of training and testing accuracy metrics. Therefore, for better readability, we summarize the time accuracy metrics for related state-of-art research's in the following table, Table I, in chronological order. According to the comparison of the table, it can be seen that our proposed model has recorded an attractive result in terms of both training and testing accuracy showing superiority over all other compared methods.

TABLE. II. COMPARISON WITH STATE OF ART METHODS: ACCURACY METRIC

Research	**Method**	**Accuracy**
Gour et. al., 2020 [26]	Customized ResHist [152- Residual Learning-CNN]	84.34%
Gupta et. al., 2020 [27]	Hybrid CNN [Employed Several Pre-Trained CNNs)	93.27%
Dabeer et. al., 2019 [28]	Customized LeNet-5 CNN	93.45%
Sagar et. al., 2019 [29]	Pre-Trained DenseNet201CNN	98.30%
Kassani et. al., 2019 [30]	Pre-Trained Combined CNNs [DenseNet201+ VGG19 + MobileNetV2]	98.13%
Gandomkar et al., 2018 [31]	Pre-Trained ResNet-152 CNN	98.77%
Adeshina et. al., 2018, [32]	New Deep CNN/14 Layers (DCNN-14)	91.5%
Han et al., 2017 [33]	New Class Structure-Based Deep CNN (CSDCNN)	93.20%
Sun, et. al, 2017 [34]	Pre-Trained GoogLeNet CNN	95.00%
Spanhol et al. 2016 [35]	Pre-Trained AlexNet CNN	84.60%
Proposed Method	Pre-Trained ResNet-50 CNN	**99.10 %**

IV. CONCLUSIONS

An efficient model for classifying the stained histological breast cancer images with high level of classification accuracy. To increase the robustness of the classifier, we employed the transfer learning of the powerful ResNet-50 CNN pretrained on ImageNet. The developed model makes use of *BreakHis* dataset with 75% of the images used for training and 25% used for testing. Indeed, the proposed work provides a comprehensive model for medical image processing/classification from input layer to the output layer. Eventually, to our knowledge, the reported results are superior to the automated analysis of breast cancer images reported in literature [26-35].

REFERENCES

[1] R. Nall, Y. Ranchod, What to know about cancer, Medically reviewed by Yamini Ranchod, Medical News Today (MNT), Jan, 6, 2020.

[2] NCI, Breast Cancer—Patient Version, National Cancer Institute (NCI), 2020.

[3] WHO. (2018). Data of breast cancer sufferers in the world. World Health Organization (WHO), Jan. 3. 2019.

[4] NDCF, Learn about breast cancer, 2019 National Breast Cancer Foundation.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. Commun. ACM 60 (6), pp. 84–90, 2017.

[6] S. Saha. A Comprehensive Guide to Convolutional Neural Networks-ELI5 way. Towards Data Science, A Medium publication sharing concepts, ideas, and codes, 2018.

[7] S. Jia. Vanishing Gradient vs Degradation. Towards Data Science, A Medium publication sharing concepts, ideas, and codes, Sep, 2018.

[8] C. Shorten. Introduction to ResNets. Towards Data Science, A Medium publication sharing concepts, ideas, and codes, Jan, 2019.

[9] K. He, X. Zhang, S. Ren, J. Sun. Deep Residual Learning for Image Recognition. Microsoft Research. 2015.

[10] J. D. Toro, et. al., Analysis of Histopathology Images. Chapter 10, Biomedical Texture Analysis Fundamentals, Tools and Challenges, Elsevier, 2017, Pages 281-314, 10.1016/B978-0-12-812133-7.00010-7.

[11] W. H. Wolberg, W. N. Street, O. L. Mangasarian, Breast cancer Wisconsin (diagnostic) data set, UCI Machine Learning Repository

[12] Lowe A KMe a, Grunkin M. Mitos atypia grand challenge 2014. 2014. https: //mitos-_atypia-_14.grand-_challenge.org/Dataset/

[13] Spanhol, F., Oliveira, L. S., Petitjean, C., Heutte, L. A Dataset for Breast Cancer Histopathological Image Classification, IEEE Transactions on Biomedical Engineering (TBME), 63(7):1455-1462, 2016. DOI: 10.1109/TBME.2015.2496264

[14] K. K. Pal and K. S. Sudeep. Preprocessing for image classification by convolutional neural networks. IEEE International Conference on Recent Trends in Electronics, Information & Communication Technology (RTEICT), Bangalore, 2016, pp. 1778-1781.

[15] W. McKinney. pandas: powerful Python data analysis toolkit Release 1.0.3. By Pandas Development Team.

[16] Y. Bengio; A. Courville; P. Vincent. Representation Learning: A Review and New Perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence. 35 (8): 1798–1828, 2013. arXiv:1206.5538. doi:10.1109/tpami.2013.50. PMID 23787338.

[17] Ganin, Y., & Lempitsky, V. Unsupervised Domain Adaptation by Backpropagation. ACM 32nd International Conference on Machine Learning, vol. 37, 2015.

[18] R.F. Lopes. Wild Data Part 3: Transfer Learning. Stratio Big Data Inc, 2018.

[19] K. He, X. Zhang, S. Ren and J. Sun . Deep Residual Learning for Image Recognition. arXiv:1512.03385v1, 2015.

[20] O. Knocklein. Classification Using Neural Networks. Medium: Toward Data Science. Jun. 2019.

[21] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res., 15(1):1929–1958, Jan. 2014.

[22] J. Brownlee. A Gentle Introduction to the Rectified Linear Unit (ReLU). Machine Learning Mastery Inc., Deep Learning Performance, Jan, 2019.

[23] Fei-Fei Li, "CS231n: Convolutional Neural Networks for Visual Recognition", Computer Science, Stanford University, 2019.

[24] M. Binieli .Machine learning: an introduction to mean squared error and regression lines. #MATHEMATICS, Free Code Camp, 16 OCT 2018

[25] A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, B. Recht . The Marginal Value of Adaptive Gradient Methods in Machine Learning. Advances in Neural Information Processing Systems 30 (NIPS 2017), Pp. 4148—4158, 2017.

[26] M. Gour, S. Jain, T. S. Kumar, Residual learning based CNN for breast cancer histopathological image classification. Int J Imaging Syst Technol. 2020; 30: 621– 635. https://doi.org/10.1002/ima.22403

[27] K. Gupta, N. Chawla, Analysis of Histopathological Images for Prediction of Breast Cancer Using Traditional Classifiers with Pre-Trained CNN, Procedia Computer Science,Volume 167, 2020, Pages 878-889, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2020.03.427.

[28] S. Dabeer, M. M. Khan, S. Islam. Cancer diagnosis in histopathological image: CNN based approach. Informatics in Medicine Unlocked, Elsevier, Vol.16, 2019, 100231.

[29] A. Sagar. Convolutional Neural Network for Breast Cancer Classification: Deep Learning for solving the most commonly diagnosed cancer in women. Towards Data Science, A Medium publication sharing concepts, ideas, and codes, 2019.

[30] S. H. Kassani, P. H. Kassani, M. J. Wesolowski, K. A. Schneider, and R.,Deters. Classification of histopathological biopsy images using ensemble of deep learning networks. ACM 29th Annual International Conference on Computer Science and Software Engineering (CASCON '19). IBM Corp., USA, 92–99, 2019.

[31] Z. Gandomkar, P. C. Brennan, and C. Mello-Thoms. MuDeRN: Multi-category classification of breast histopathological image using deep residual networks. Artificial Intelligence in Medicine 88, pp.14 – 24, 2018.

[32] Adeshina SA, Adedigba AP, Adeniyi AA, Aibinu AM. Breast cancer histopathology image classification with deep convolutional neural networks. In: 2018 14th international conference on electronics computer and computation (ICECCO). IEEE; 2018. p. 206–12.

[33] Z. Han, B, Wei, Y. Zheng, Y. Yin, K. Li, and S. Li. Breast cancer multi-classification from histopathological images with structured deep learning model. Scientific reports 7, 1 (4172), 2017.

[34] J. Sun and A. Binder, "Comparison of deep learning architectures for H&E histopathology images," 2017 IEEE Conference on Big Data and Analytics (ICBDA), Kuching, 2017, pp. 43-48.

[35] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte. Breast cancer histopathological image classification using convolutional neural networks. International joint conference on neural networks (IJCNN). IEEE, 2560—2567, 2016.