Efficient, Ultra-facile Breast Cancer Histopathological Images Classification Approach Utilizing Deep Learning Optimizers

Sarpong Kwadwo Asare
School of Electronic Science and Engineering
University of Electronic Science and Technology of China
611731, West Hi-Tech Zone
Chengdu, Sichuan, P.R. China

Fei You
School of Electronic Science and Engineering
University of Electronic Science and Technology of China
611731, West Hi-Tech Zone
Chengdu, Sichuan, P.R. China

Obed Tettey Nartey
School of Computer Science and Engineering
University of Electronic Science and Technology of China
611731, West Hi-Tech Zone
Chengdu, Sichuan, P.R. China

ABSTRACT

Conventional approaches to breast cancer diagnosis are associated with drawbacks that ultimately affect the quality of diagnosis and subsequent treatment, pushing for the need for automatic and precise classification of breast cancer tumors. The advent of deep learning methods has witnessed an increasing interest in their applications in many tasks. The specific case of using convolutional neural networks with transfer learning has witnessed tremendous successes in many classification tasks. Nonetheless, with transfer learning, the sheer number of parameters associated with deep networks coupled with the distance disparity between source data and target data leave networks prone to overfitting, particularly in the case of limited data. Also, negative transfer may occur in the situation where the source and target domains are not related. This work proposes a simple convolutional neural network model trained from scratch for discriminating benign and malignant breast cancer tumors in histopathological images. Four deep learning optimization algorithms are leveraged and explored to ascertain how optimizers aid in finding good sets of parameters that help minimize loss and increase overall classification accuracy. By adopting a polynomial learning rate decay scheduling and implementing several data augmentation techniques that regulate overfitting and improve the generalization ability of the proposed model, accuracy, sensitivity, specificity, and Area Under the Curve values of 89.92%, 94.02%, 86.42%, and 0.884 (88.4%), respectively are reported.

Keywords

Breast Cancer, Convolutional Neural Networks, Deep Learning, Classification, Optimization methods

1. INTRODUCTION

Breast cancer is one of major contributors of death caused by cancer in women within the ages of 29 to 59 years [1]. A projection by the World Health Organization (WHO) estimates that by 2025, the number of breast cancer cases will shoot up to 19.3 million. [2]. Thus, making early diagnosis and treatment a vital step in preventing its spread and, subsequently, a reducing morbidity rates [3]. Traditional manual diagnosis approaches include mammography, ultrasound imaging, and biopsy. Though these methods are helpful in diagnosing and examining suspicious cancerous tissues, they require pathologists with a high level of expertise, the absence of which makes diagnosis error-prone, not to mention the associated intense workload on pathologists. Moreover, the level of agreement between specialists on diagnosis results is approximately 75% [4]. These factors necessitate the need for computer-aided diagnosis (CAD) systems.

Recent advancements in deep learning in medical diagnosis have demonstrated remarkable successes in many classification tasks, and it is drawing growing interest among researchers [5, 6]. Convolutional Neural Networks (CNN), a deep learning method, has achieved state-of-the-art performances in recognition tasks. The inherent nature of these deep models implies that they are capable of learning rich and useful features or patterns on their own, without requiring human intervention or the use of hand-engineered features. Different layers in deep models learn different features, which culminate in a specific classification or recognition task. Nonetheless, deep models thrive on the availability of large, and well annotated datasets, often due to the large number of parameters associated with these models. Such vast datasets are virtually nonexistent in the medical domain. Even in the case of datasets with appreciable size, class imbalance poses another challenge in classification. Class imbalance refers to the situation where there are more images for a particular class (or classes) compared to the other class(es). Consequently, a deep model gets exposed to more images of a particular class during training, and as such, skews its final classification output towards that particular class. In effect, though the classification accuracy might be high, the output is still highly weighted by the class with the most number of images. Another challenge is with the nature of the medical images. For a deep model, distinguishing between natural images (say a cat and an airplane), is pretty simple owing to the availability of visual clues. Breast cancer histopathological images do not possess image properties that present a lot of visual clues for a deep model during training. The images demonstrate a lot of inter and intra-class similarities that tend to hinder the generalization ability of a deep model on medical imaging classification tasks.

Several works have reported on transfer learning for breast cancer classification. Transfer learning is a technique that adapts a pre-trained deep model to a secondary tasks. A pre-trained model has already been trained on a huge dataset (usually ImageNet) and as such, possesses a rich features that can be transferred to a another tasks in a similar domain, achieving excellent performance. ImageNet possesses a collection of natural images and as such, a deep model trained on ImageNet performs extremely well on a secondary classification task with dataset which comprises natural images. The same is usually not the case for breast cancer histopathological images. Adapting a pre-trained model to a breast cancer classification task is not devoid of issues. The distance disparity between the source data (ImageNet) and the target data (breast cancer histopathological images) is a huge one, which without further image processing and augmentation techniques, impacts the ability of deep models to generalize on breast cancer data. This is because, the features a deep model learns when presented with a car as an input image is totally different from features it learns when the input image is a breast cancer image. The resulting problem is overfitting and poor model classification performance.

In this work, a simple deep CNN model for classifying breast cancer histopathological images (Invasive Ductal Carcinoma) as either benign or malignant is proposed. The proposed model is trained from scratch on a breast cancer histopathological dataset. Training the model from scratch on a relatively small dataset allows the model to learn features by itself. Coupled with the drastic reduction in the number of parameters associated with the proposed model, overfitting is effectively minimized without trading off the model's performance. Also, the impact of learning rate when training a deep model with random initialization is accessed, as the choice of a learning rate determines the convergence of a model with minimal loss. To this end, four deep learning optimization algorithms are leveraged with two learning rate values (with a polynomial rate decay scheduling) and the performance of the proposed model is accessed, based on its ability to classify an image as either benign or malignant. A combination of data augmentation techniques results in a great classification performance, with less parameters and an effective check on overfitting.

2. RELATED WORK

CNNs have been consistently achieving impressive results in image classification tasks. Over the years, trend is evident in the development of several deep learning algorithms that seek to improve accuracies and minimize loss [7–10]. These models have achieved excellent accuracy performances in classification tasks on natural images datasets such as the CIFAR, MNIST and ImageNet. The application of deep models to the task of classifying breast cancer histopathological images has also witnessed a surge in recent years. Commendable outcomes have been reported in the literature, showing promising prospects in the application of deep models in medical imaging classification tasks. That, notwithstanding, data unavailability and imbalance in image classes still pose a challenge to the task of accurately classifying breast cancer images. Nonetheless, the application of deep models for breast cancer classification has superior performance compared to traditional classification methods. The work done in [11] adopts an end to end approach in training a convolutional neural network for accurately detecting breast cancer on screening mammograms. Their approach makes use of lesion annotations only during the initial training stage while preceding stages make use of only image-level labels. This method eliminates the need to rely on lesion annotations, which are hardly available. On the Digital Database for Screening Mammography (CBIS-DDSM), their approach obtains a per-image AUC of 0.88 for a single model and 0.91 AUC for averaging four models. On the full-field digital mammography (FFDM) images from the INbreast database, they obtain a per-image AUC of 0.95 and 0.98 for averaging four models.

In [12], authors employ a hybrid convolutional and recurrent deep neural network for classifying breast cancer pathological images, obtaining an accuracy of 91.3%. In [13], the authors introduced a hybrid CNN capable of utilizing the local and global features of an image for accurate prediction. In this work, the authors also introduced hierarchical voting and bagging techniques that help improve the classifier?s performance. Their approach achieved a classification accuracy of 87.5%. The authors in [14] proposed a multiple instance learning framework for a CNN by introducing a new pooling layer. The pooling layer aided in accumulating features with the most information from patches that make up a whole slide, and reported an accuracy of 88%. The work done in [2] fine-tunes the AlexNet model for binary classification of breast cancer tumors on two datasets. Their approach involves replacing the classification layer of the pre-trained AlexNet model with a fully connected layer connected to a support vector machine classifier. They report an accuracy of 87.2% with an AUC of 94%.

The authors in [15] introduced a transition module that can capture filters at different scales, collapsing the filters via global average pooling. This ultimately reduces the size of the network from convolutional to fully connected layers. Training on small dataset yielded an accuracy of 91.9%. Spanhol *et al.* [16] extracted features from a CNN and used these features as input to a traditional classifier. Bayramoglu *et al.* [17] distinguished between benign and malignant classes of breast cancer histology images.

3. BACKGROUND

3.1 Deep Learning and Breast Cancer Classification

Deep learning methods are making remarkable progress and performance in many computer vision tasks. Convolutional neural networks have become the popular deep learning method for wide range of task like classification, recognition and detection. This trend has seen deep models being implemented in histopathological image classification [18]. Nonetheless, the classification of histopathological images is more challenging compared to the case of natural image classification. This is because; 1) histopathological images have high resolutions, 2) the feature space representation of a pathological image patch is not adequately rich, and 3) number of images of in a histopathological dataset is smaller compared to natural image datasets. These factors make the task of classifying histopathological images a challenging one. Even with these chal-

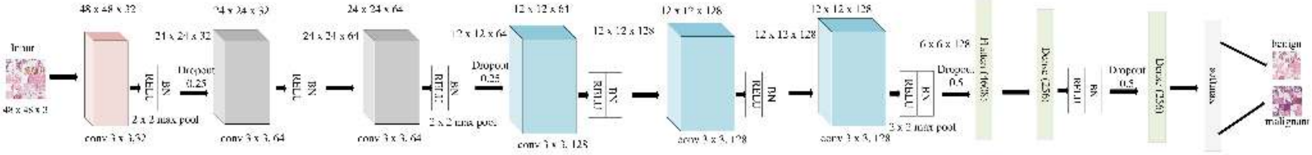

Fig. 1. Proposed CNN architecture

lenges, CNN models do a rather good job in classifying histopathological images. A popular application of CNN models for classification is implementing transfer learning, as this technique helps curb the issue of lack of well labeled and annotated images for histopathological classification.

Transfer learning is a technique that seeks to extract knowledge from one or source task and applies the knowledge to a target task [19]. In such a scenario, the target task has less high quality training data. A domain is represented as, $D = \{\chi, P(X)\}$, (where χ is a feature space, and $P(X)$ is a marginal probability distribution) and a task, $T = \{y, f(.)\}$ (where y is a label space, and $f(.)$ is an objective prediction function). Hence, for a source domain D_S and a task T_S, a target domain D_T and a task T_T, the objective of transfer learning to aid in improving the learning of the predictive function $f_T(.)$ in D_T through the knowledge in D_S and T_S, given that $D_S \neq\neq \mathrm{D}_T$ and $T_S \neq \mathrm{T}_T$. Based on this definition, assumption for transfer learning is that, the source and target domains are related to each other. However, in some cases when the source and target domain are not related, brute-force transfer learning may be unsuccessful and even in the worst case, degrade the performance of learning in the target domain. Considering the disparity between natural images and histopathological images,training a pre-trained CNN model (trained on ImageNet) on histopathological images hurts accuracy performance, even with state-of-the art CNN models.

For this reason, works on histopathological image classification implemented CNN models trained from scratch with Gaussian distribution [18], [20]. Training a CNN model from scratch on a relatively smaller dataset allows the model to learn data patterns by itself with a lesser number of parameters to learn from, which increases the ability of the model to generalize well on target data. When training CNN models, the learning rate is an important hyper-parameter to consider and adjust in a bid to achieve a higher accuracy with minimal loss. Choice of a learning rate determines the rate at which a model adapts to a problem. It controls how the weights of the networks are adjusted with respect to the loss gradient. Setting this hyper-parameter too small may cause the network not to learn anything at all. Too high a value may also cause the network overshoot in areas of low loss. Setting a scheduler that adjusts the learning rate appropriately is a vital step in training deep models. In this work, polynomial learning rate scheduler is adopted and four deep learning optimizers are leveraged in order to access performance of the proposed model in obtaining the best accuracy performance with minimal loss.The next section provides a brief insight to deep learning optimization methods.

3.2 CNN Optimization Methods

Optimization is a vital aspect of machine learning and deep learning. Optimization methods enable neural networks to learn useful patterns from data. These useful patterns enable CNN models to accurately predict and assign labels to input data. Generally, for classification tasks, we define a scoring function that maps input data to output data class labels. A scoring function is defined in terms of two parameters, a *weight matrix* **W** and a *bias vector* **b**, as given in Equation 1.

$$f(x_i, W, b) = Wx_i + b \tag{1}$$

A good shot at improving classification accuracy is to tune parameters of a *weight matrix* **W** or a *bias vector* **b**. Nonetheless, the process of adjusting these parameters in improving classification accuracy is not straightforward, and it is often classified as an optimization problem.When training deep learning models, a common assumption is to consider the objective function as a sum of a finite number of functions, as shown in Equation 2.

$$f(x) = \frac{1}{n}\sum_{i=1}^{n} f_i(x) \tag{2}$$

where $f_i(x)$ is a loss function on the training data instance indexed by i. A loss function measures the agreement between predicted class labels and ground truth labels, for a given data. The goal of training deep learning models is to minimize the loss function, thereby increasing accuracy. Optimization methods aid in finding a set of parameters **W** and **b** that help to minimize the loss function *w.r.t* the scoring function [21]. We briefly touch on four optimization methods explored in our work. Interested readers are referred to the great work by [21] for more information about CNN optimizers.

3.2.1 Stochastic Gradient Descent (SGD). SGD differs from the standard vanilla gradient descent algorithm in that, it computes the gradient of the objective function and performs parameter updates on small batches of training data, rather than the entire training data. The vanilla gradient descent algorithm computes the gradient of the objective function and performs parameter updates for the entire training data, mathematically expressed in Equation 3.

$$\theta = \theta - \eta.\nabla_\theta J(\theta) \tag{3}$$

where J_θ is the objective function for parameters ?, and ? is the learning rate. This kind of implementation is very slow and intractable for large datasets. SGD rather performs parameter update for every epoch, given some training example, x^i and label y^i. Mathematically, this is expressed in Equation 4 as [23];

$$\theta = \theta - \eta.\nabla_\theta J(\theta; x^i; y^i) \tag{4}$$

This kind of parameter update is faster, resulting in faster convergence. Nonetheless, the main idea behind the different gradient descent algorithm remains the same. We first evaluate parameters in an iterative manner, compute the loss, and then take a small step in the direction of minimal loss. The learning rate controls the size of the step, making it a significant parameter during the optimization process. Small step size results in the network learning virtually nothing. Large step size may render the network overshooting areas of lower loss, which even lead to overfitting.

3.2.2 Adagrad. Adagrad is an adaptive learning rate algorithm in that, it adapts the learning rate to the network parameters [22]. It performs smaller updates on parameters that change frequently, while larger updates are performed on parameters that have infrequent features. This implies that for every parameter θ_i at every time *t* uses a different learning rate. Denoting the gradient at time step *t* as g_t, the objective function is defined in Equation 5 as [20]

$$g_{t,i} = \nabla_\theta J(\theta_{t,i}) \tag{5}$$

The learning rate is modified for every parameter θi at time step *t* based on gradients already computed for θ_i. The update rule is given in Equation 6.

$$\theta_{t+1,1} = \theta_{t,1} - \frac{\eta}{\sqrt{R_{t,ii} + \in}} . g_{t,i} \tag{6}$$

$R_t \in R^{dxd}$ is a diagonal matrix, and $\in$ is a smoothing term. Using Adagrad eradicates the need for manually updating the learning rate. However, the squared gradients in the denominator in Equation 6 keep accumulating. This causes the learning to shrink, and with time, the learning rate becomes too small for the network to learn anything.

3.2.3 Adaptive Moment Estimation (Adam). The Adaptive Moment Estimation (Adam) algorithm also computes adaptive learning rates for each parameter [23]. It stores an exponentially decaying average of past squared gradients (just as RMSprop [24]) as well as an exponentially decaying average of previous gradients. These two averages are expressed in Equations 7 and 8, respectively.

$$n_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t \tag{7}$$

$$u_t = \beta_2 u_{t-1} + (1 - \beta_2) g_t^2 \tag{8}$$

where n_t and u_t are the mean (first momentum) and the variance (second momentum), respectively. g_t is the gradient at time step *t*. The update step is expressed in Equation 9.

$$\theta_{t+1} = \theta - \frac{\eta}{\sqrt{\hat{u}_t} + \in} . \hat{n} \tag{9}$$

where $u_t^\wedge$ and $n_t^\wedge$ are the estimates of the first and second moments, depicted in Equations 10 and 11, respectively.

$$\hat{n} = \frac{n_t}{1 - \beta_1^t} \tag{10}$$

$$\hat{u} = \frac{u_t}{1 - \beta_2^t} \tag{11}$$

3.2.4 Rectified Adam (RAdam). Introduced by Liu et al. [25], the Rectified Adam (or RAdam), as a variant of the Adam optimizer that seeks to resolve the issue of large variance in the early stage of training which results in poor generalization when implementing adaptive learning rates. The authors attribute the large variance to the lack of training samples during the early stage. They introduce a learning rate warm-up heuristic, as a variance reduction technique, that helps stabilize training and improve generalization. The authors argue that, rather than setting the learning rate as a constant or decreasing it over time, the learning rate warm sets the learning rate as some small value in the first few steps. They introduced a rectification term and applied it to the Adam optimizer, yielding the RAdam optimizer. The rectification term is expressed in Equation 12.

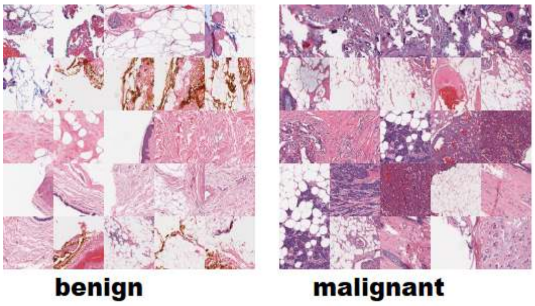

Fig. 2. Sample images from benign and malignant classes

$$r_t = \sqrt{\frac{(\rho_t - 4)(\rho_t - 4)\rho_\infty}{(\rho_\infty - 4)(\rho_\infty - 2)\rho_t}} \tag{12}$$

where ρ is the degree of freedom. The authors realized that, for an approximated simple moving average (SMA), when its length is less than or equal to four, the variance of the adaptive learning rate is intractable, and the learning rate is inactivated [24].

3.3 Materials

The breast cancer histopathology image dataset was developed by [26, 27] for classifying Invasive Ductal Carcinoma (IDC). IDC is the most prevalent sub type of all breast cancers. The original dataset consists of 162 whole mount slide images of breast cancer specimens scanned at 40x. The spatial dimensions of slide images mean these images are naturally huge. Therefore, in making the dataset somehow less cumbersome to work with, a total of 277,524 patches, each 50x50 pixels were extracted. Out of this, 198,738 are negative samples (samples without breast cancer), and 78,786 are positive samples (samples with breast cancer). This class distribution clearly indicates a huge class imbalance. Coupled with the high resolution characteristics of histopathological images, the task of accurately classifying histopathological images becomes challenging. Every image in the dataset has a specific file format. For instance, an image *10253-idx5-x1351-y1101-class0.png* can be interpreted as follows ? *10253-idx5* is the patient ID, *x1351* is the x-coordinate of the crop, *y1101* is the y-coordinate of the crop and *class0* is the class label (0 means benign, 1 means malignant). Figure 2 shows sample benign and malignant images from the dataset.

4. METHODOLOGY

4.1 CNN Architecture

Convolutional neural networks are feed-forward neural networks. However, unlike traditional neural networks, where each neuron in the input layer is connected to all output neurons in the subsequent layer, in CNNs, neurons in the next layer are connected to only a small region of the preceding layer. This concept is known as local connectivity, and it drastically reduces the number of parameters in a network. The basic layers of a CNN are; convolutional layer, activation layer, pooling layer, dropout layer, and fully connected layer. Each layer applies different sets of learnable filters, and by stacking these layers together, a CNN model can learn filters capable of detecting different features useful for a specific task.

The CNN architecture proposed in this work consists of six convolutional layers, detailed as;

- First convolutional layer learns 32 filters, each of size 3 x 3
- Second convolutional layer learns 64 filters, each of size 3 x 3
- Third convolutional layer learns 64 filters, each of size 3 x 3
- Fourth convolutional layer learns 128 filters, each of size 3 x 3
- Fifth convolutional layer learns 128 filters, each of size 3 x 3
- Sixth convolutional layer learns 128 filters, each of size 3 x 3

The proposed network is detailed in Figure 1. RELU activation is applied to every convolutional and fully connected layer. The RELU layer enhances faster convergence and also ensures that all negative activations are converted to zero. A batch normalization layer is then applied [28] after each RELU activation layer. Batch normalization layers help normalize the activations of an input volume before passing activations to the next layer. Batch normalization layers are effective in reducing the number of epochs required to train a network, stabilizing the network, and also allow for a number of learning rate and regularization strengths. A pooling layer is applied after the batch normalization layer for the first, third and sixth convolutional layers. Pooling layers reduce the spatial size of the input volume, allowing for a reduction in the number of parameters. In the proposed architecture, max-pooling layers have a size of 2 x 2. Dropout with keep probability of 0.25 is applied after the third and sixth convolutional layers. A flatten layer, two fully connected layers and a dropout layer with a probability of 0.5 are added after the second fully connected layer.

The network's weight are initialized using Gaussian distribution. For each optimizer used, experiments are performed with two learning rate values, 0.01 and 0.001, respectively with a polynomial decay scheduling, expressed in Equation 13. The polynomial learning rate scheduling allows the learning rate to decay over a fixed number of epochs. The two learning rate values are a popular choice in many works. Input image patches to the CNN model are resized to 48 x 48.

$$\alpha = \textit{initLR} * (1 - \frac{epoch}{T_{epochs}})^p \tag{13}$$

initLR is the base learning rate, T_{epochs} is the total number of epochs, p is the exponential power, which is set to 1. For the SGD optimizer, the momentum is set to 0.9. The momentum factor aids the loss function in arriving at a global minimum. For the Adam optimizer, β_1 value is set to 0.9 and β_2 is set to 0.99. The model is trained with a batch size of 64, for a total of 40 epochs. All experiments are carried out using Keras (version 2.2.4) [29] with Tensorflow backend (version 1.12) [30] and CUDA 9.0. The hardware platform is an RTX 2080 graphic card with 8GB memory and a 32GB RAM. 80% of the dataset is used as training data and 20% as testing data. 10% of the training data is reserved as validation data.

In a bid to increase the ability of the proposed model to generalize well on training data and minimize overfitting, data augmentation techniques are implemented. Data augmentation involves techniques aimed at purposely perturbing data before feeding them into a network for training. As a result, the network sees new data that are slightly modified versions of the input data, allowing the network to learn robust features. Table 1 shows the data augmentation techniques employed in this work. To account for the skew in data, the class weight for the training data are also computed. Computing the class weight tells the model to pay attention to the class with lesser data samples.

Table 1. **Data augmentation techniques and values**

Parameter	Value
Rescale	1/255
Rotation range	0.2
Zoom range	0.05
Shear range	0.05
Width shift range	0.1
Height shift range	0.1
Horizontal flip	True
Vertical flip	True

4.2 Performance Metrics

The model's performance is accessed in terms of accuracy, sensitivity, specificity, precision, recall, and F1-score. These parameters are related to the true positive (TP), true negative (TN), false positive (FP), and false false-negative (FN) rates, respectively. True positive measures how correctly a classifier predicts the positive class.

True negative measures how correctly a classifier predicts the negative class. False positive measures how, incorrectly, a classifier predicts the positive class. False negative measures how, incorrectly, a classifier predicts the negative class. These metrics are expressed mathematically in Equations 14 to 19.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \tag{14}$$

$$Sensitivity = \frac{TP}{TP + FN} \tag{15}$$

$$Specificity = \frac{TN}{TN + FP} \tag{16}$$

$$Precision = \frac{TP}{TP + FP} \tag{17}$$

$$Recall = \frac{TP}{TP + FN} \tag{18}$$

$$F1 - score = 2(\frac{precision * recall}{precision + recall}) \tag{19}$$

5. RESULTS

This section presents experimental outcomes. Table 2 and Table 3 show accuracy, sensitivity, specificity and AUC values for the four optimizers with learning rates 1e-3 and and1e-2. In terms of accuracy, the RAdam optimizer records the best value of 89.92%, with a learning rate of 1e-3. RAdam also yields the best sensitivity value 94.02% with the same learning rate. In terms of individual optimizers, the highest accuracy obtained by the SGD optimizer is 88.94%, with a learning rate of 1e-3; Adagrad yields its highest accuracy of 87.15% with a learning rate of 1e-2, with Adam optimizer yielding its highest accuracy of 89.71% with a learning rate of 1e-3 .

With the exception of the Adagrad optimizer, the remaining optimizers show an accuracy edge when the learning is set to 1e-3 compared to 1e-2. Accuracy and loss plots are shown in Figure 3 and

Table 2. **Accuracy, Sensitivity, Specificity and AUC values outcomes when learning rate = 1e-3. Best results are indicated in bold**

Optimizer	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC
SGD	88.94	91.97	81.21	0.859
Adagrad	85.53	88.62	77.64	0.831
Adam	89.80	93.34	80.76	0.878
RAdam	**89.92**	**94.02**	79.47	0.853

Table 3. **Accuracy, Sensitivity, Specificity and AUC values outcomes when learning rate = 1e-2. Best results are indicated in bold**

Optimizer	Accuracy (%)	Sensitivity (%)	Specificity (%)	AUC
SGD	88.35	89.11	**86.42**	0.865
Adagrad	87.15	92.09	74.54	0.860
Adam	89.71	91.44	85.27	**0.884**
RAdam	88.84	91.13	83.00	0.871

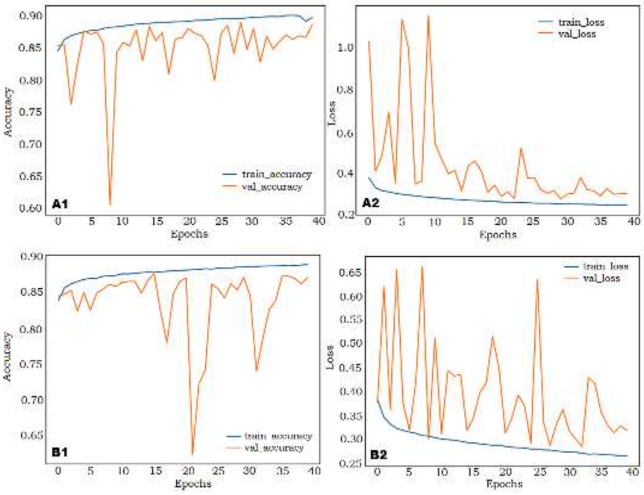

Fig. 3. Accuracy and loss plots for Adagrad and SGD optimizers. A1 is accuracy plot for Adagrad optimizer, and A2 is its corresponding loss plot. B1 is an accuracy plot for SGD and B2 is its corresponding loss plot. The respective plots indicate that overfitting is effectively minimized.

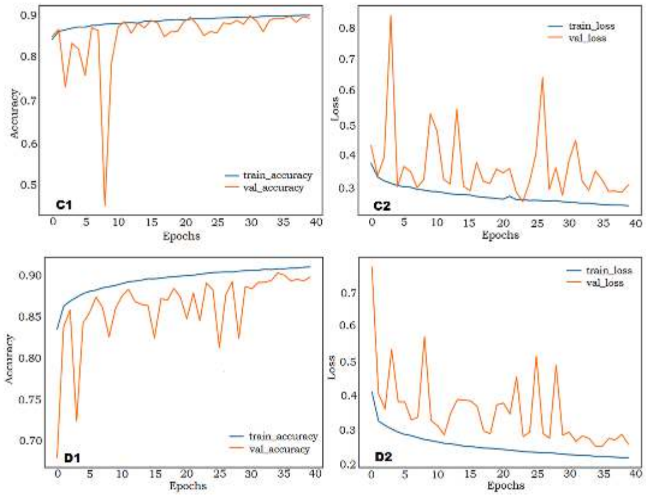

Fig. 4. Accuracy and loss plots for Adam and RAdam optimizers. C1 is the accuracy plot for Adam optimizer, and C2 is its corresponding loss plot. D1 is the accuracy plot for RAdam and D2 is its corresponding loss plot. The respective plots indicate that overfitting is effectively minimized

Figure 4, respectively. A similar pattern is observed with the sensitivity values, with SGD, Adam, and RAdam, all yielding higher sensitivity values with a learning rate of 1e-3 compared to 1e-2. Adagrad still throws an exception, with higher accuracy and sensitivity values obtained when the learning is set to 1e-2. This outcome conforms with the assertion in [20] that, for the Adagrad optimizer, a good rule of thumb is to set the learning rate to 1e-2, as this is mostly used in many applications. However, for specificity values, a reverse trend is observed.

SGD yields the best value of 86.42% when the learning rate is set to 1e-2. Adam and RAdam also demonstrate a competitive edge *w.r.t* sensitivity when the learning rate is set to 1e-2 compared to 1e-3. For Adagrad, however, setting the learning rate to 1e-3 yields 77.64% specificity compared to 74.54% with a learning rate of 1e-2. This work adopts a polynomial learning rate decay scheduling, rather than a fixed learning rate decay. Hence, for purposes of comparative analysis, the performance of the proposed model is accessed in terms of accuracy when a fixed learning rate is used. For a learning of 1e-3, SGD records 85.54%, Adagrad records 86.39%, Adam records 87.77%, and RAdam records 85.99%, respectively. For a learning rate of 1e-2, SGD records 88.59%, Adagrad records 86.94%, Adam records 87.55%, and RAdam records 87.43%. Higher accuracies are obtained when the polynomial decay is used compared to those obtained when a fixed learning rate is used for both learning rate values.

For a binary classification, the Area Under The Curve (AUC) Receiving Operating Characteristic curve (ROC curve) shows the capability of a model in distinguishing between classes by providing a summary of the trade-off between the true positive rate and the false-positive rate for a predictive model. AUC ROC curves are shown in Figure 5. For the AUC values, Adam records the best AUC value of 0.884 with a learning rate of 1e-2. It is observed that individual optimizers record higher AUC values when the learning rate is set to 1e-2 compared to when the learning rate is 1e-3.

However, for a dataset that exhibits such a huge class imbalance, the AUC ROC does not always give an accurate measure of the performance of model in distinguishing classes. The precision and recall values give a better indication of a model?s performance. A summary of precision, recall and F1-score values for the individual classes is provided in Table 4 and Table 5 respectively. The numerical advantage of the benign class (class 0) in terms of the number

Table 4. **Precision, Recall and F1-score values when learning rate = 1e-3. The per-class values obtained by the benign class outweigh that of the malignant class because the benign class has more data samples**

Optimizer	IDC Class	Precision (%)	Recall (%)	F1-Score (%)
SGD	benign	93	92	92
	malignant	80	81	81
Adagrad	benign	93	84	88
	malignant	68	83	75
Adam	benign	93	92	93
	malignant	81	84	82
RAdam	benign	93	92	93
	malignant	80	82	81

Table 5. **Precision, Recall and F1-score values when learning rate = 1e-2. The per-class values obtained by the benign class outweigh that of the malignant class because the benign class has more data samples**

Optimizer	IDC Class	Precision (%)	Recall (%)	F1-Score (%)
SGD	benign	94	89	92
	malignant	76	86	81
Adagrad	benign	90	92	91
	malignant	79	75	77
Adam	benign	94	91	93
	malignant	80	85	82
RAdam	benign	93	91	93
	malignant	84	83	82

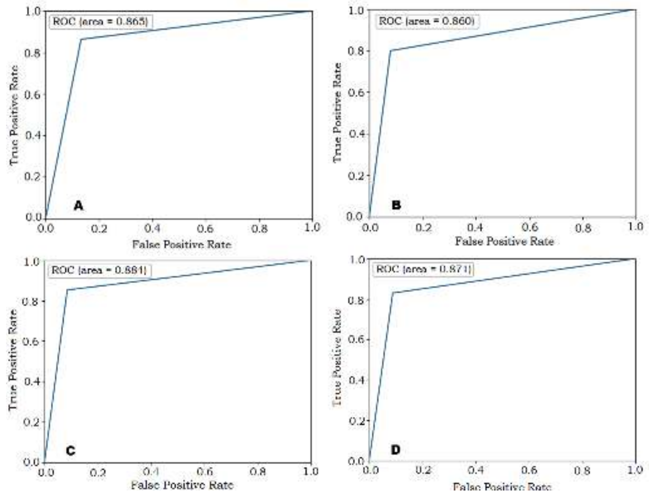

Fig. 5. AUC ROC curves for all four optimizers. A is SGD, B ? Adagrad, C ? Adam, D - RAdam. The best AUC value (0.884) is recorded with the Adam optimizer when the learning rate is set to 1e-2

of images is evident in the precision, recall and F1-score values. The highest precision value for benign class is 94%, compared to 84% for the malignant class. Recall values show the highest value of 92% for benign and 85% for malignant, with the highest F1-score values of 93% for benign class and 82% for malignant class. Nonetheless, the precision, recall and F1-score values indicate a good performance by the model.

6. DISCUSSION

In this work, a CNN model is trained from scratch, for discriminating breast cancer histopathological images as either benign or malignant. The proposed model consists of six convolutional layers, with an RELU layer and a batch normalization layer, after each convolutional layer. A filter size of 3 x 3 is used in order to capture relevant features in the images. The images in the dataset used in this work were scanned at 40x magnification factor, at a size of 50 x 50 pixels. The dataset exhibits a huge class variation with images belonging to the benign class being almost twice the number of images belonging to the malignant class. Of particular concern when training CNN models from scratch is the issue of overfitting, since the weights of the model are initialized randomly. The accuracy plots in Figure 3 and Figure 4 indicate that overfitting is effectively checked. This is attributed to the addition of batch normalization layers in the model's architecture as well as implementing data augmentation.

For training the model, four CNN optimization algorithms are explored with a polynomial learning rate decay scheduling, rather than a fixed learning rate, to evaluate the impact of such scheduling on the overall performance of our model. Overall best results obtained are: an accuracy of 89.92%, sensitivity of 94.02%, specificity of 86.42% and AUC value of 0.884.

This work also explored four deep learning optimizers with two learning rate values. Results indicate that, training the model with a learning of rate of 1e-3 yielded better results compared to training with a value of 1e-2, except for the Adagrad optimizer. A learning rate value of 1e-2 is a little high for training the model, especially when the model's weight are randomly initialized and it ultimately impacts the final accuracy of the model. Training with a slightly lower value of 1e-3 allows the network to learn useful patterns relevant to the overall classification task. The exception with the Adagrad optimizer is in line with the fact that, for most applications,

Table 6. **Accuracy performance with other works using the same IDC dataset. Best results are indicated in bold**

Work	Accuracy (%)
[26]	84.23
[27]	84.68
This work	**89.92**

the Adagrad optimizer performs well when the learning rate is set to 1e-2. Again, it is observed that, for both learning rate values, training with a fixed learning rate led to accuracy reduction for all optimizers, and this validates the choice of the adopted polynomial decay scheduling over a fixed decay scheduling.

That said, the inter-class precision, recall and F1-score values indicate that, results are heavily weighted by the benign class, even after applying data augmentation and computing class weights. The reason for this could be that, the augmented data still had a considerable number of samples for the benign class, hence affecting the overall output performance. That notwithstanding, the overall accuracy, sensitivity, specificity and AUC values obtained by the proposed model are impressive for a such dataset with a huge class imbalance.

6.1 Performance Comparison With Related Works

This work focuses on discriminating breast cancer histopathological images in to benign and malignant classes. As mentioned, images in the dataset used in this work were scanned at a magnification factor of 40X. For this reason, the comparison is based on works that used the same IDC dataset used in our work, as shown in Table 6. In [26], the authors used a 3 layer CNN architecture, consisting of two convolutional and pooling layers, and one fully connected layer. The convolutional layers learn 16 and 32 filters, each of size 8 x 8, and the fully connected layer learns 128 filters, of size, 8 x 8. The network accepts input patches of size 50 x 50. An accuracy of 84.23% was reported. In [27], authors adopt the AlexNet model for classifying benign and malignant classes. They adopt three patch selection techniques. First, they first resize 50 x 50 patches to 32 x 32, then cropped each 50 x 50 image to 32 x 32 and finally applied cropping and additional rotations, achieving an accuracy of 84.68%.

7. CONCLUSION

The focus of this work is to classify breast cancer images into benign and malignant by training a CNN model from scratch, rather than relying on transfer learning. A polynomial learning rate decay scheduling was adopted, that allows the learning rate to decay over a fixed number of epochs, and four deep learning optimization algorithms were explored. By applying data augmentation techniques on the highly unbalanced and complex dataset of histopathological images, the proposed CNN model achieves an overall accuracy of 89.92% and effectively minimizes overfitting. This outcome is an indication that, convolutional neural networks have a tremendous capability of solving tasks, even when models are trained from scratch. Future work will aim at discriminating multiple breast cancer classes.

8. CONFLICT OF INTEREST

The authors declare no conflict of interest.

9. ACKNOWLEDGEMENT

Sarpong Kwadwo Asare and Obed Tettey Nartey contributed equally to this work.

10. REFERENCES

[1] K. D. Miller et al., ?Cancer treatment and survivorship statistics, 2016,? CA. Cancer J. Clin., 2016.

[2] D. A. Ragab, M. Sharkas, S. Marshall, and J. Ren, ?Breast cancer detection using deep convolutional neural networks and support vector machines,? PeerJ, 2019.

[3] R. A. Smith, V. Cokkinides, and H. J. Eyre, ?American Cancer Society Guidelines for the Early Detection of Cancer, 2006,? CA. Cancer J. Clin., 2006.

[4] J. G. Elmore et al., ?Diagnostic concordance among pathologists interpreting breast biopsy specimens,? JAMA - J. Am. Med. Assoc., 2015.

[5] Y. Lecun, Y. Bengio, and G. Hinton, ?Deep learning,? Nature. 2015.

[6] K. Sirinukunwattana, S. E. A. Raza, Y. W. Tsang, D. R. J. Snead, I. A. Cree, and N. M. Rajpoot, ?Locality Sensitive Deep Learning for Detection and Classification of Nuclei in Routine Colon Cancer Histology Images,? IEEE Trans. Med. Imaging, 2016.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich et al., ?Going deeper with convolutions,? in Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on. Cvpr, 2015.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton, ?Imagenet classification with deep convolutional neural networks,? in Advances in neural information processing systems, 2012, pp. 1097?1105.

[9] K. Simonyan and A. Zisserman, ?Very deep convolutional networks for large-scale image recognition,? International Conference on Learning Representations (ICLR) (oral), 2015.

[10] G. Huang, Z. Liu, K. Q.Weinberger, and L. van der Maaten, ?Densely connected convolutional networks,? in Proceedings of the IEEE conference on computer vision and pattern recognition, vol. 1, no. 2, 2017, p. 3.

[11] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride, and W. Sieh, ?Deep Learning to Improve Breast Cancer Detection on Screening Mammography,? Sci. Rep., 2019.

[12] R. Yan et al., "Breast Cancer histopathological image classification using a hybrid deep neural network," Methods, 2019.

[13] B. Zhang, ?Breast cancer diagnosis from biopsy images by serial fusion of Random Subspace ensembles,? in Proceedings - 2011 4th International Conference on Biomedical Engineering and Informatics, BMEI 2011, 2011.

[14] K. Das, S. Conjeti, A. G. Roy, J. Chatterjee, and D. Sheet, ?Multiple instance learning of deep convolutional neural networks for breast histopathology whole slide classification,? in Proceedings - International Symposium on Biomedical Imaging, 2018.

[15] S. Akbar, M. Peikari, S. Salama, S. Nofech-Mozes, and A. L. Martel, ?The transition module: a method for preventing overfitting in convolutional neural networks,? Comput. Methods Biomech. Biomed. Eng. Imaging Vis., 2019.

[16] F. A. Spanhol, L. S. Oliveira, C. Petitjean, and L. Heutte, 'Breast cancer histopathological image classification using convolutional neural networks,' in Proc. Int. Joint Conf. Neural Netw. (IJCNN), Jul. 2016, pp. 2560-2567.

[17] N. Bayramoglu, J. Kannala, and J. Heikkil, "Deep learning for magnification independent breast cancer histopathology image classification," in Proc. 23rd Int. Conf. Pattern Recognit. (ICPR), Dec. 2016, pp. 2440-2445.

[18] G. Litjens, T. Kooi, B.E. Bejnordi, S. Aaa, F. Ciompi, M. Ghafoorian, V.D.L. Jawm, G.B. Van, C.I. S nchez, A survey on deep learning in medical image analysis, Med. Image Anal. 42 (2017) 60?88.

[19] Pan S.J, Yang Q. "A survey on transfer learning." IEEE Transaction on Knowledge Data Engineering 2010;22(10):1345?59.

[20] D. Bardou, K. Zhang, and S. M. Ahmad, ?Classification of Breast Cancer Based on Histology Images Using Convolutional Neural Networks,? IEEE Access, 2018.

[21] Sebastian Ruder. ?An overview of gradient descent optimization algorithms?. In: CoRRabs/1609.04747 (2016).URL:http://arxiv.org/abs/1609.04747.

[22] J. Duchi, E. Hazan, and Y. Singer, ?Adaptive subgradient methods for online learning and stochastic optimization,? in COLT 2010 - The 23rd Conference on Learning Theory, 2010.

[23] D. P. Kingma and J. L. Ba, ?Adam: a Method for Stochastic Optimization,? Int. Conf. Learn. Represent. 2015, 2015.

[24] G. E. Hinton, N. Srivastava, and K. Swersky, ?Lecture 6a-overview of mini-batch gradient descent,? COURSERA Neural Networks Mach. Learn., 2012.

[25] L. Liu et al., ?On the Variance of the Adaptive Learning Rate and Beyond.? In: CoRRabs/1908.03265 (2019).URL:https://arxiv.org/abs/1908.03265

[26] A. Janowczyk and A. Madabhushi, ?Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases,? J. Pathol. Inform., 2016.

[27] A. Cruz-Roa et al., ?Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks,? in Medical Imaging 2014: Digital Pathology, 2014.

[28] S. Ioffe and C. Szegedy, ?Batch normalization: Accelerating deep network training by reducing internal covariate shift,? in 32nd International Conference on Machine Learning, ICML 2015, 2015.

[29] F. Chollet, ?Keras: Deep Learning for humans,? Github, 2015.

[30] M. Abadi et al., ?TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems,? 2016.